DEVELOPMENT OF A LOWER-EXTREMITIES EXOSKELETON FOR POSTURAL-SWAY IN SIT-TO-STAND MOVEMENT

LEE JUN HAN

UNIVERSITI TUNKU ABDUL RAHMAN

DEVELOPMENT OF A LOWER-EXTREMITIES EXOSKELETON FOR POSTURAL-SWAY IN SIT-TO-STAND MOVEMENT

LEE JUN HAN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Biomedical Engineering with Honours

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

May 2025

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Name : Lee Jun Han

ID No. : 2101827

Date : 12th May 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Mr. Chong Yu Zheng, for placing his trust in me and granting full ownership of this project. His support, encouragement, and the opportunity to work with the SONY MOCOPI system were invaluable. I am also deeply thankful to my co-supervisor, Dr. Chan Siow Cheng, for her patient mentorship and generosity in answering my many questions, academic or otherwise, and for providing essential tools such as the EMG system and Wii Balance Board.

My appreciation extends to Ms. Tan Yin Qing, for her guidance and insights in biomechanics, as well as for allowing me to use the Wobble Board in the early stages. I am also grateful to the lecturers of the Department of Mechatronics and Biomedical Engineering (DMBE) for cultivating a supportive and intellectually engaging environment. Special thanks to Head of Department, Dr. Goh Choon Hian, for entertaining my algorithm-related questions, and to, my moderator, Dr. Tan Lee Fan, for her thoughtful evaluation and kind feedback.

To my family, thank you for being my foundation. My mother and sister (psychology major), both thoughtful and empathetic, supported me emotionally and helped me stay grounded. My twin brother (M.Sc.. Physics), though often the comic relief, readily stepped in when things got tough, assisting with planning, GUI development, and even hands-on tasks like sewing. My father reminded me of the importance of discipline and persistence, especially when motivation ran low.

To my friends, Ying Hang, Yan Xing, and all who stayed despite my flaws, offering support in every sense, intellectual, emotional, and physical, thanks. To my UTAR peers from CHKL, Ten Yong, Jing Hao, Yao Zong, Ming Jun, and Ken Sean, thank you for years of friendship and encouragement. I’m especially grateful to Jia Hao for sharing his mechanical design insights and helping with 3D printing tasks when I needed it most.

To the juniors I’ve worked with or mentored, I hope our interactions added value to your journey, just as they have to mine. And finally, to everyone who took part or offered support in this project, THANK YOU.

ABSTRACT

Postural sway, a critical indicator of balance control, is often heightened in individuals with chronic low back pain (CLBP), age-related decline, and neurodegenerative disorders such as Parkinson's disease. While passive exosuits offer ergonomic support, they lack the capacity for real-time corrective intervention. This project introduces a lightweight, fabric-based active back-support exoskeleton powered by pneumatic actuation, designed to detect and correct static postural sway. The system integrates an ESP32 microcontroller, IMU-based sway detection, and a threshold-based control algorithm for real-time actuation. The fully assembled prototype was evaluated in 15 subjects across varying stance (normal, tandem) and visual (eyes open/closed) conditions using surface electromyography (sEMG) and centre of pressure (CoP) metrics. Under the most challenging balance condition, Tandem Stance Eyes Closed (TSEC), results showed a 51.5% reduction in CoP pathlength, 27.4% and 41.8% decreases in Vapmean and Vmlmean, and 34.7% and 20.7% reductions in DMLSD and DAPSD, respectively. sEMG analysis indicated a significant drop in trunk muscle activation, with External Obliques (38.3%), Rectus Abdominis (51.2%), and Erector Spinae (L3) Right (41.8%) showing the largest reductions in RMS amplitude during TSEC trials. The sway detection algorithm achieved 70% classification accuracy, supporting low-power, real-time execution on embedded hardware. These findings validate the exosuit's ability to enhance postural stability and reduce neuromuscular strain during quiet stance. The system demonstrates potential as a practical rehabilitation aid for individuals with CLBP, older adults, or those with early-stage Parkinson's disease. Future work may involve adaptive control integration and broader clinical validation.

Keywords: Active Exoskeleton, Postural Sway, Centre of Pressure (CoP), Kinematics, Biomechanics

Subject Area: R856-857 Biomedical engineering. Electronics. Instrumentation

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

ACL	Anterior Cruciate Ligament
APA	Anticipatory Posture Adjustments
AUC	Area Under Curve
Ax	Anterior-Posterior Axis
Az	Medial-Lateral Axis
BMI	Body Mass Index
BSE	Back Support Exoskeleton
CAD	Computed Aided Design
CB	Chest Breadth
CL	Clavicular Length
CLBP	Chronic Lower Back Pain
CNN	Convolutional Neural Networks
CNS	Central Nervous System
CoM	Centre of Mass
CoP	Centre of Pressure
D_RMS	Root Mean Square Sway
DAP	Displacement in Anterio-Posterior
DAPSD	AP Sway Dispersion
DDVD	Dynamic Difference of Vector Distances
DI	Directional Indices
DIAP	Directional Index in Anterior-Posterior
DIML	Directional Index in Medial-Lateral
DML	Displacement in Medial-Lateral
DMLSD	ML Sway Dispersion
DNS	Dynamic Neuromuscular Stabilisations
DoF/DOF	Degree of Freedom
DR	Displacement Range
EC	Eyes Closed
EEG	Electroencephalogram
EMG	Electromyography
EO	External Obliques
ES	Erector Spinae

ES-L	Left Erecter Spinae
ES-R	Right Erecter Spinae
FFT	Fast Fourier Transform
FPR	False Positive Rate
GRU	Gates Recurrent Units
GUI	Graphical User Interface
HG	Hip Girth
IMU	Inertial Measurement Unit
INST	Unstable Condition
IoT	Internet of Things
IPD	Intermittent Proportional Derivative
KNN	K-Nearest Neighbours
LQR	Linear Quadratic Regulator
LSTM	Long Short-Term Memory
MAD	Median Absolute Deviation
MEMS	Micro-Electromechanical System
MIMU	Magneto-Inertial Measurement Unit
MiPump2	Mi Portable Electric Air Compressor 2
ML	Machine Learning
MS	Multiple Sclerosis
MVC	Maximum Voluntary Contraction
NB	Naïve Bayes
NSEC	Normal Stance Eyes Closed
NSEO	Normal Stance Eyes Open
NTP	Network Time Protocol
OMCS	Optical Motion Capture Systems
PAM	Pneumatic Artificial Muscle
PD	Parkinson’s Disease
PEA	Prediction Ellipse Area
PLA	Polylactic Acis
PNF	Proprioceptive Neuromuscular Facilitation
Q%	Quality Index
RA	Rectus Abdominis
RF	Random Forest

RI	Relative Importance
RMS	Root Mean Square
ROC	Receiver Operating Characteristic
SampEn	Sample Entropy
SD	Standard Deviation
SEA	Series Elastic Actuator
sEMG	Surface Electromyography
SPI	Serial Peripheral Interface
STAB	Stable Condition
SV	Sway Vector
SVamp	Stability Vector Amplitude
SVaz	Stability Vector Azimuth
SVM	Support Vector Machine
SW	Shoulder Width
TL	Torso Length
TPR	True Positive Rate
TSEC	Tandem Stance Eyes Closed
TSEO	Tandem Stance Eyes Open
TTS	Time to Stabilisation
Vapmean	Mean Anterio-Posterior Velocity
Vmlmean	Mean Medial-Lateral Velocity

a_x, a_y, a_z	Acceleration in X, Y, Z Axes Respectively
H_1	Chest-to-ankle Height
H_2	Chest-to-Waist Height
τ	Torque
m	Mass
g	Gravitational Acceleration
σ	Standard Deviation
Σ	Sum
P	Pressure

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Postural sway refers to the natural shifting of the body's centre of pressure (CoP) while maintaining balance, that occur even during quiet standing, which is crucial for postural control and involves sensory and motor coordination. These micro-adjustments are essential for maintaining upright balance and are mediated by complex neuromuscular coordination involving visual, vestibular, and somatosensory systems. However, excessive, or poorly controlled sway is commonly associated with chronic low back pain (CLBP), aging, and neurological disorders such as Parkinson’s disease (PD), increasing the risk of falls and functional decline (Brumagne et al., 2008). In CLBP, exaggerated sway often results from altered proprioceptive inputs and compensatory trunk stiffness, contributing to instability and recurrent pain episodes (Sung & Lee, 2024b; Alshahrani et al., 2025). These individuals may adopt rigid motor strategies and exhibit slower or asymmetric sway, especially under conditions where visual cues are limited. Targeted interventions, such as proprioceptive neuromuscular facilitation (PNF) exercises and inspiratory muscle training, have shown to improve postural control and reduce pain (Sipko et al., 2021; Borujeni, & Yalfani, 2019). Age-related degeneration in muscle strength and sensory integration also leads to compromised sway regulation. Elderly individuals with CLBP show increased sway in both anterior-posterior and mediolateral directions and significantly impaired dynamic balance during functional tests, highlighting the interaction between musculoskeletal and sensory deficits in older populations (Mesci et al., 2016). Parkinson’s patients display abnormal sway due to impaired basal ganglia-mediated control, further increasing fall risk and physical disability.

Therapeutic exercises aimed at reducing postural sway can also enhance functional capacity and balance performance, especially in those with severe low back pain (Kuukkanen, & Mälkiä, 2000). Gender differences in response to interventions like back support exoskeletons (BSEs) must be considered, as studies have shown varying effects on postural stability between males and females (Park et al., 2019 & 2021). While BSEs can reduce physical demands, they may also challenge postural balance due to their design. Addressing rigid postural control strategies in individuals with recurrent low back pain, which often rely heavily on ankle proprioception, is important for preventing pain recurrences (Brumagne et al., 2008). Overall, correcting postural sway through therapeutic exercises, proprioceptive training, and well-designed exoskeletons can improve stability and reduce pain, with future research needed to explore gender-specific effects and task conditions. While, passive back support exoskeletons have emerged as a supplementary tool to reduce muscular load and provide ergonomic support, they lack the capacity to dynamically assist postural correction or adapt to real-time changes in balance demands.

This project aims to address the issue of postural sway using an active exoskeleton, with a focus on upper body support. Unlike current solutions that primarily use passive exoskeletons or lower body active exoskeletons, which tend to be bulky and aesthetically unappealing for everyday use, our approach involves developing a lightweight, active back support exoskeleton. The goal is to correct postural sway by integrating an active component that is both functional and discreet. Current literature supports the effectiveness of lower body exoskeletons in mitigating postural sway, but there is a notable gap in research regarding upper body exoskeletons. This project will investigate whether an active upper body exoskeleton can also effectively address postural sway. The report will review relevant literature, evaluate the proposed model's feasibility, and discuss the conceptual designs of the prototype. Additionally, we will explore the challenges and problems associated with the initial solutions and propose strategies to address these issues.

1.2 Importance of the Study

Postural sway is a common condition characterised by instability in the body's stance, it is a critical determinant of physical function and fall risk, especially in vulnerable groups like older adults, individuals with neuromuscular disorders, and those recovering from injury. The inability to regulate sway effectively leads to muscular overcompensation, fatigue, and in many cases, long-term disability. This condition affects a broad population and can significantly impact daily functioning and quality of life. Current solutions include soft passive exoskeletons, often made of elastics, which are discreet and user-friendly but may not provide sufficient support for rehabilitation purposes. While passive exosuits made of elastic materials are lightweight and user-friendly, they fall short in therapeutic and rehabilitative contexts due to their lack of adaptive control.

On the other hand, many existing lower body active exoskeletons are bulky and complex, incorporating large power systems, control units, battery packs, and actuators, making them less portable and practical for daily use. Our proposed solution aims to develop an upper body active exoskeleton that distributes components more uniformly and discreetly, reducing overall bulkiness. By investigating this approach, we aim to contribute new knowledge and potential solutions to the field of postural sway, offering a more practical and aesthetically acceptable alternative. The intended application ranges from fall prevention in elderly individuals to balance support in people with CLBP and early-stage Parkinson's disease. Our design philosophy emphasizes portability, aesthetic integration, and functional efficacy, offering a compelling alternative to current tools in the rehabilitation and occupational support space (Alshahrani et al., 2025; Ruhe, 2011).

1.3 Problem Statement

The primary focus of this study is to address postural sway, specifically in static or quiet stance situations, through the development and implementation of an active exoskeleton system. The challenge is to create a solution that effectively corrects postural sway while being practical, lightweight, and suitable for everyday use.

Postural sway, characterised by the natural but sometimes excessive oscillation of the body while standing still, presents a significant challenge for maintaining balance, especially in individuals with muscular or neurological impairments. While existing solutions primarily focus on passive exoskeletons or bulky lower body active exoskeletons, there remains a gap in effective, practical solutions for correcting postural sway in the upper body. The traditional passive exoskeletons, often made from elastic materials, provide limited support for rehabilitation and are not suited for more intensive balance correction. Conversely, lower body active exoskeletons, though effective, are typically cumbersome and impractical for everyday use. Our project seeks to address these issues by developing an active back support exoskeleton designed to correct postural sway specifically during static stances.

1.4 Aim and Objectives

The primary aim of this project is to design and develop a lightweight, active back support exoskeleton to effectively address postural sway in the trunk area, thereby improving balance and stability during quiet stance. To achieve this aim, the project has the following objectives:

1. Development of back support fabric-based exoskeleton for standing posture sway detection and correction.
2. Develop standing sway classification and detection algorithm.
3. Test and validate functionality of prototype.

1.5 Scope and Limitation of the Study

This project will focus on developing an upper body active exoskeleton specifically aimed at improving postural sway during static stances. The scope includes the design and integration of the exoskeleton, the development of a postural sway detection system, and preliminary performance evaluations. The project will not address lower body balance issues directly, nor will it explore dynamic or high-motion scenarios beyond static stance corrections. Limitations of the project include potential challenges in achieving optimal actuator performance while maintaining a lightweight and comfortable design. Additionally, the effectiveness of the exoskeleton in real-world conditions may be constrained by the accuracy and responsiveness of the detection system. The project will also need to navigate the trade-offs between the complexity of the design and the practical usability of the exoskeleton, ensuring that it remains portable and user-friendly.

1.6 Contribution of the Study

This study presents a novel contribution to the expanding field of postural control and wearable assistive technologies by introducing an active upper-body exoskeleton specifically designed to mitigate postural sway during static standing. While previous research has extensively explored the relationship between chronic low back pain (CLBP) and postural instability, most interventions have concentrated on lower-body support systems or passive textile-based aids. Clinical evidence demonstrates that individuals with CLBP exhibit significantly increased postural sway, characterised by larger sway areas and higher sway velocities, under both visual (eyes open) and non-visual (eyes closed) conditions. This instability is further exacerbated by age, body mass index (BMI), and pain severity (Alshahrani et al., 2025). Despite these findings, current interventions either lack the capacity for active correction or are too cumbersome for everyday use, particularly among elderly or functionally impaired individuals.

Additionally, research on older adults with CLBP has identified pronounced impairments in both dynamic balance and static postural control, particularly along the anterior-posterior axis. These impairments are evident in functional assessments such as the timed-up-and-go and chair stand tests, where individuals with CLBP perform significantly worse than healthy controls (Mesci et al., 2016). These findings underscore the critical need for user-friendly, targeted interventions that can enhance upper-body stability without compromising mobility or comfort. However, most commercially available exoskeletons for back support remain passive and offer limited functionality for balance correction or therapeutic engagement.

Neurophysiological studies further reveal that individuals with chronic musculoskeletal pain often adopt maladaptive postural strategies, such as trunk stiffening and reduced sway variability. These compensatory mechanisms may undermine long-term motor adaptability and postural reflexes (Sung & Lee, 2024b). The proposed system addresses these issues by employing active actuators in conjunction with sensor-driven feedback loops, enabling real-time, adaptive sway modulation tailored to the user’s physiological state. This feature holds promise for early-stage neurodegenerative populations, including those with Parkinson’s disease, where deficits in anticipatory postural adjustments are a primary contributor to fall risk.

Methodologically, this study also advances the field by introducing a novel framework for postural sway detection, classification, and correction through an integrated wearable platform. The hybridization of exoskeletal support with intelligent control algorithms marks a significant shift from traditional mechanical interventions toward more interactive, therapeutic systems. By targeting the upper trunk—an often overlooked but critical component of postural regulation—this research opens new avenues in exoskeleton ergonomics and rehabilitation science.

In conclusion, the study makes a multidimensional contribution by addressing a well-defined clinical gap, advancing wearable robotics toward active upper-body stabilization, integrating intelligent real-time sway correction mechanisms, and establishing foundational principles for applications in aging, musculoskeletal, and neurodegenerative populations. This approach aligns with emerging evidence suggesting that postural sway abnormalities in CLBP are not uniform across individuals (Mikkonen et al., 2022), thus reinforcing the need for personalized, dynamic interventions over static, one-size-fits-all solutions.

1.7 Outline of the Report

This project report begins with Chapter 1: Introduction, which presents the background and clinical relevance of postural sway in populations such as elderly individuals, patients with chronic low back pain (CLBP), and those with early-stage Parkinson’s disease. It defines the problem of insufficient support in current passive and lower-body exoskeleton systems and states the aim of developing a lightweight, upper-body exosuit for static sway correction. The objectives, scope, limitations, and novelty of the proposed approach are also discussed, outlining the need for a discreet, functional system that integrates wearable sensing and real-time actuation.

Chapter 2: Literature Review covers a wide range of foundational knowledge relevant to postural sway, including its physiological mechanisms, types (e.g., quiet stance, anticipatory adjustments), and clinical implications. The chapter reviews current treatment modalities, existing exoskeleton technologies, and limitations of passive supports. It highlights the lack of adaptive, upper-body active systems specifically designed for static sway correction. The review also includes detailed comparisons of actuators (e.g., pneumatic, servo, soft robotics), wearable sensor technologies (e.g., IMUs, sEMG), and algorithmic methods for sway detection, classification, and correction.

Chapter 3: Methodology and Work Plan provides a step-by-step breakdown of the exosuit's design and development process. It begins with system requirements and component selection, including material choice, actuator force calculations, and biomechanical design constraints. The hardware development section details CAD iterations, fabrication strategies, circuit design, and integration of pneumatic and electronic subsystems. This chapter also introduces the system architecture, graphical user interface (GUI), and the full algorithm pipeline for sway classification. Biomechanical assessment methods are described, including CoP metric extraction and EMG muscle group monitoring. The chapter concludes with testing protocols involving 36 initial data collection subjects and 10 for validation, detailing their stances, visual conditions, and inclusion criteria.

Chapter 4: Results and Discussion presents and interprets experimental findings. It includes prototype performance and usability observations (e.g., don/doff time, comfort), sway detection algorithm results, and statistical analysis of EMG and CoP data across all test conditions. Detailed figures show reductions in sway pathlength, directional deviations, and trunk muscle activity. Subject-level heatmaps and stance-specific boxplots offer insights into how different individuals and stances responded to the intervention. The discussion connects these outcomes to clinical goals, supporting the use of such exosuits for balance rehabilitation and trunk stabilization.

Chapter 5: Conclusion and Recommendations summarizes the project's achievements in designing a functional, lightweight pneumatic exosuit capable of reducing sway and muscular load. It reflects on design trade-offs and technical challenges encountered. The recommendations section proposes future work directions, such as adaptive thresholding, machine learning-based control, improved comfort padding, longer battery life, and clinical trials for elderly and Parkinson's populations. Appendices include extended graphs, raw data tables, hardware schematics, and codes.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explores key aspects of postural sway and corrections. It begins with an overview of postural sway, highlighting its role in stability and movement, followed by types and causes. It then breaks down the stages of clinical implications as well as current treatment. The chapter also covers postural sway detection, examines actuators used in exoskeletons, and reviews sensors for collecting biomechanical data. Additionally, it reviews previous back support exoskeletons designed for rehabilitation, offering insights for safe and effective exoskeleton design. Overall, it provides a comprehensive view of technological advancements to help determine the feasibility of powered back support exoskeletons for postural sway.

2.2 Postural Sway

Postural sway refers to the involuntary, continuous movement of the body's centre of mass as it strives to maintain balance during quiet standing. It reflects the body's complex interaction with its environment, influenced by sensory input, neurological control, and physical condition. A variety of factors, including sensory deficits, injuries, and age, can significantly alter postural sway, making it a valuable measure in assessing balance and stability across different populations.

2.2.1 Postural Sway Dynamics

Postural sway consists of two primary components: a slow non-oscillatory movement and a faster damped-oscillatory motion. These components are managed through feedback mechanisms within the sensory-motor system (Kiemel et al., 2006). The slow non-oscillatory movement reflects the body's large-scale adjustments to maintain balance, while the faster, oscillatory movement involves finer adjustments. Both mechanisms work in tandem to ensure postural stability, highlighting the critical role of the nervous system in managing sway under different conditions.

2.3 Types and Causes of Postural Sway

2.3.1 Age-Related Variations in Postural Sway

The development of postural stability follows a clear trajectory across the lifespan. In childhood, postural sway decreases as sensory and motor systems mature, with significant improvements noted around age seven. Boys generally demonstrate faster stabilisation compared to girls, possibly due to differences in motor development (Riach & Hayes, 1987). However, in older adults, the natural increase in sway is more closely associated with the deterioration of sensory function than age itself (Anson et al., 2017). This suggests that interventions targeting sensory enhancement could mitigate the effects of aging on balance and postural control.

2.3.2 Sensory Impairments and Their Impact on Postural Sway

Sensory inputs, particularly visual, vestibular, and proprioceptive information, play a crucial role in maintaining postural stability. Any impairment in these systems can lead to significant changes in postural sway. For instance, individuals with vision loss, vestibular dysfunction, or proprioceptive deficits often exhibit exaggerated sway patterns as the body struggles to compensate for missing or altered sensory feedback (Carroll & Freedman, 1993). This highlights the interdependence of sensory systems in maintaining balance and the need for multisensory rehabilitation in populations with sensory impairments.

2.3.3 Postural Sway Following Injuries

Injuries, particularly those affecting the musculoskeletal system, can have long-lasting effects on postural control. For example, anterior cruciate ligament (ACL) reconstruction, a common surgical procedure following knee injuries, has been shown to result in persistent postural sway deficits even after patients have returned to normal activities. These findings underscore the importance of incorporating balance-focused rehabilitation programs to address lingering instability and prevent further injury (Paterno et al., 2013).

2.3.4 Postural Sway in Neurological Conditions

In neurological conditions like Parkinson’s disease, postural sway becomes a significant risk factor for falls. As the disease progresses, motor control deteriorates, leading to an increase in sway amplitude and frequency. Excessive postural sway in Parkinson’s patients is closely correlated with an elevated risk of falls, making postural assessment a critical tool in evaluating fall risk and designing intervention strategies (Frenklach et al., 2009).

2.3.5 Summary

Postural sway serves as a critical measure of balance and stability across various populations. Its dynamics are influenced by age, sensory function, injury, and neurological conditions. Targeted interventions, such as rehabilitation programs and balance training exercises, can help improve postural control and reduce the risk of falls in populations with increased sway. Understanding the factors that influence postural sway is essential for developing effective strategies to enhance balance and prevent injuries. Table 2.1 provides a summary of the distinct types of postural sway and their underlying causes, emphasising the wide range of factors that can influence balance and stability.

Table 2.1: Types of Postural Sway and Their Causes

Type of Postural Sway	Causes	Study
Normal postural sway	Natural movement to maintain balance, controlled by sensory feedback loops	(Kiemel et al., 2006)
Increased sway in childhood	Immature visual and motor systems in young children	(Riach & Hayes, 1987)
Sway with sensory impairment	Loss of proprioception, vision, or vestibular function	(Anson et al., 2017)
Increased sway post-injury	Long-term deficits following ACL or other musculoskeletal injuries	(Paterno et al., 2013)
Parkinson’s disease	Progressive motor deficits leading to excessive sway and increased fall risk	(Frenklach et al., 2009)

2.4 Forms of Postural Sway

Postural sway refers to the involuntary movements made by the body to maintain balance in various static and dynamic contexts. The degree and nature of sway depend on the task being performed, the physical and cognitive demands placed on the body, and individual factors such as age, sensory input, and muscle function. This literature review explores key forms of postural sway, including quiet stance, lifting tasks, anticipatory postural adjustments (APA), and cognitive load conditions, Table 2.2 provides an overview.

Table 2.2: Forms of Postural Sway

Type	Characteristics	Study
Quiet Stance	Involves minor involuntary sway to maintain balance; influenced by vision, age, and muscle activation.	(Kiemel et al., 2006)
Lifting Tasks	Postural sway increases during lifting due to added load, with adjustments to maintain COM and COP.	(Hill et al., 2018)
Anticipatory Postural Adjustments	APAs are pre-movement adjustments to avoid balance loss during voluntary motion, such as swaying or weight shifts.	(Krishnamoorthy & Latash, 2005)
Cognitive Load Conditions	Cognitive tasks during standing increase sway as attention is divided between cognitive and postural control.	(Mitra et al., 2013)

2.4.1 Quiet Stance

Quiet stance is the simplest and most studied form of postural sway, involving minor involuntary movements while a person stands still. The centre of pressure (CoP) displacement during quiet stance is relatively low compared to more dynamic tasks, making it an ideal measure of baseline balance and stability. Factors such as vision, muscle stiffness, proprioception, and age can influence postural sway during quiet stance. For instance, older adults typically exhibit increased sway due to sensory decline and reduced muscle mass (Kiemel et al., 2006; Kouzaki & Masani, 2012). Quiet stance serves as a reference point for understanding more dynamic postural tasks and is characterised by slow feedback loops that help stabilise the centre of mass (COM) (Kiemel et al., 2006).

Quiet stance refers to the act of maintaining balance with minimal voluntary movement. During quiet stance, postural sway is controlled primarily by slow, non-oscillatory feedback mechanisms that help stabilise the COM. Several key factors influence quiet stance such as feedback loops. The body's ability to stabilise itself during quiet stance is governed by slow feedback loops, which prevent excessive sway by adjusting muscle activity and sensory input (Kiemel et al., 2006). Besides, the degree of muscle activation during quiet stance can significantly influence sway patterns. Increased activation of the ankle muscles, for example, can lead to greater sway, while external supports, such as orthotics, can reduce sway by adding passive stiffness (Warnica et al., 2014).

The spectral content of sway us also key. Studies examining the spectral analysis of postural sway during quiet stance have revealed that different sensory inputs, such as vision and surface compliance, affect sway differently in specific directions (anteroposterior versus medio-lateral) (Singh et al., 2012). Lastly, older adults tend to exhibit greater postural sway during quiet stance due to age-related declines in muscle strength, proprioception, and physiological factors like increased tremors in the plantar flexor muscles (Kouzaki & Masani, 2012). Table 2.3 summarises the factors and findings of quiet stance postural sway.

Table 2.3: Factors of Quiet Stance Postural Sway

Key Factor	Findings	Study
Feedback Loops in Quiet Stance	Slow feedback mechanisms stabilise COM during quiet stance, minimising postural sway.	(Kiemel et al., 2006)
Muscle Activation	Increased muscle activation increases sway; passive stiffness (e.g., orthotics) reduces sway.	(Warnica et al., 2014)
Spectral Content of Sway	Vision and surface compliance influence sway in specific directions (anteroposterior vs. medio-lateral).	(Singh et al., 2012)

Age and Muscle Volume	Older adults show greater sway due to reduced muscle volume and increased physiological tremor.	(Kouzaki & Masani, 2012)

2.4.2 Lifting Tasks

Postural sway increases during lifting tasks due to the additional physical load, which requires the postural system to adjust dynamically to maintain balance. When lifting, individuals must control both the COM and COP, making sway more pronounced. External loads, changes in posture, and environmental factors can significantly affect sway patterns, often leading to larger oscillations as the body compensates for the shifting weight (Hill et al., 2018). This increase in sway highlights the complex interaction between mechanical loading and postural control systems during physical tasks.

2.4.3 Anticipatory Postural Adjustments (APA)

Anticipatory postural adjustments (APAs) are pre-movement shifts made by the body to prepare for voluntary actions, such as weight shifting, reaching, or body swaying. APAs help minimise the risk of imbalance by adapting the body's posture to account for upcoming movements. These adjustments are particularly important in tasks requiring dynamic stability, such as walking, turning, or sports activities (Krishnamoorthy & Latash, 2005). Effective APAs reduce the likelihood of falls or instability by proactively aligning the body’s COM before motion.

2.4.4 Cognitive Load Conditions

Engaging in cognitive tasks while standing has been shown to increase postural sway. When attention is divided between cognitive demands and postural control, the brain allocates fewer resources to balance, leading to a greater degree of sway. This phenomenon is most apparent in quiet stance conditions, where individuals show increased postural instability when simultaneously performing cognitive tasks such as counting or problem-solving (Mitra et al., 2013). This interaction underscores the importance of attentional resources in maintaining postural stability, especially in populations where multitasking may pose additional risks, such as older adults.

2.5 Clinical Implications

Postural sway is a critical measure of balance control, and excessive sway is often associated with a range of clinical conditions, including Parkinson's disease, age-related decline, and musculoskeletal disorders. Increased postural sway can significantly elevate the risk of falls, particularly in populations such as the elderly and individuals with neurological impairments. Understanding the clinical implications of postural sway is essential for developing effective interventions to reduce fall risk and improve balance.

Key clinical implications include Parkinson’s disease, aging, and occupational hazards. For instance, patients with Parkinson’s disease exhibit increased postural sway, particularly in the mediolateral (side-to-side) direction. This abnormal sway pattern compromises stability and significantly increases the risk of falls. Addressing postural instability is thus a major focus in managing Parkinson’s symptoms. Besides, age-related deterioration in proprioception and muscle mass leads to greater postural sway, particularly during quiet standing. This increased sway contributes to a higher risk of falls and decreased mobility in older adults (Kouzaki & Masani, 2012). Balance training and strength maintenance are crucial for mitigating these effects. Occupational hazards are also implications, thus the vast development of exoskeletons in military (Mendoza et al., 2023). Workers involved in physically demanding jobs, such as repetitive tasks or heavy lifting, often experience elevated postural sway due to muscle fatigue and strain. This increased sway, particularly in the lumbar region, heightens the risk of musculoskeletal injuries over time (Koopman et al., 2019). Ergonomic interventions, including the use of assistive devices, are commonly recommended to reduce injury risk.

2.6 Current Treatments

Several interventions are used to improve postural control and minimize excessive sway, which is detailed in Table 2.4. These approaches include physiotherapy, balance training, and the application of assistive devices like passive exoskeletons. While passive exoskeletons have demonstrated effectiveness in reducing physical strain during activities such as lifting, no active exoskeleton technology has been specifically developed to aid in quiet stance. This gap exists due to the inherent challenges of real-time feedback and movement adaptation required for static tasks.

Table 2.4: Types of Postural Sway and Their Current Treatment

Type of Postural Sway	Current Treatment	Study
Quiet Stance	Balance training, passive back-support exoskeletons for reducing muscle strain.	(Park et al., 2021)
Dynamic Movements	Physiotherapy and active exoskeletons designed to support dynamic movement tasks.	(Layne et al., 2022)

2.6.1 Challenges in Developing Active Exoskeletons for Quiet Stance

Active exoskeletons are designed to provide real-time adjustments to support movement and balance. However, in the context of quiet stance, where postural adjustments are minimal and constant, designing an exoskeleton capable of continuous, precise feedback is highly complex. The continuous monitoring required for small, involuntary movements makes passive systems more practical for quiet stance applications. For instance, passive exoskeletons, such as the Laevo model, effectively reduce muscle strain without needing to engage in dynamic posture correction (Park et al., 2021). These devices are tailored for load reduction, not for real-time balance control.

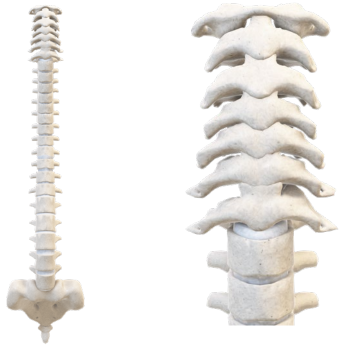
Active exoskeletons are highly effective for dynamic movement tasks, where movement support and real-time feedback are essential. However, current technology lacks the ability to provide the continuous, subtle feedback necessary for static balance tasks like quiet stance. As a result, passive systems remain the preferred option for tasks involving postural control during periods of relative stillness, while active exoskeletons are used to assist with more dynamic movements.

2.6.2 Summary

Postural sway plays a critical role in the diagnosis and management of various clinical conditions, particularly in populations at risk of falls and musculoskeletal injuries. While interventions like passive exoskeletons are useful for reducing strain during physical tasks such as lifting, the development of active systems for static postural control, such as quiet stance, remains limited. Future research should focus on bridging this gap to create more comprehensive treatment options for balance-related disorders.

2.7 Anatomical and Biomechanical Considerations of Spine in Exoskeleton Design

The cervical spine, comprising seven vertebrae (C1-C7), depicted in Figure 2.1, represents a structurally delicate yet highly mobile region of the body. Its role in supporting and protecting critical neurological structures, such as the spinal cord and brainstem, makes it particularly vulnerable to injury. Consequently, the design of assistive devices like exoskeletons must consider the anatomical and biomechanical characteristics of this region to avoid exacerbating injury risks. This literature review discusses the constraints and risks associated with cervical spine support, highlighting why exoskeleton designs should avoid direct contact or support at this level.



(a) (b)

Figure 2.1: (a) Spinal Cord, (b) Cervical Spine (C1-C7)

2.7.1 Anatomical and Biomechanical Vulnerability of the Cervical Spine

The cervical spine is uniquely characterised by its wide range of motion and relatively low structural support compared to other spinal regions. This flexibility, crucial for head rotation, flexion, and extension, also makes the cervical vertebrae more prone to injury under mechanical stress. According to Shea et al. (1991), undue pressure or improper mechanical load in this area significantly raises the risk of spinal cord damage, especially in patients with pre-existing conditions like cervical spinal cord injuries (SCIs). Given this vulnerability, exoskeleton designs generally avoid targeting the cervical spine to prevent additional injury risk and maintain the integrity of the spinal cord.

The cervical spine plays a critical role in balancing and aligning the head over the torso, a function that requires high flexibility. Research suggests that exoskeletons applied directly to the cervical region could interfere with natural posture and proprioception, leading to reduced mobility. Goldschmidt et al. (2019) note that biomechanical support systems, such as exoskeletons, can compromise the natural alignment and movement of the cervical spine, restricting head and neck motion. This interference can result in discomfort and increase the risk of biomechanical strain, emphasising the importance of preserving the cervical spine's natural range of motion.

2.7.2 High Risk of Injury in the Cervical Region

While cervical exoskeletons have been investigated for their potential to alleviate neck strain during repetitive tasks, these systems can inadvertently increase the risk of muscle fatigue or injury. Misalignment of the exoskeleton with the natural biomechanics of the cervical spine can impair neck motion, leading to discomfort and strain over time. Giovanelli et al. (2022) suggest that exoskeleton support is more effective when applied to the thoracic and lumbar regions, as these areas bear more load and are less prone to biomechanical disruption than the cervical spine. Supporting the cervical region directly with an exoskeleton can lead to muscle strain, emphasising the importance of designing devices that align with natural spinal movements.

2.7.3 Implications for Exoskeleton Design

The anatomical and biomechanical constraints of the cervical spine present significant challenges for exoskeleton design. To mitigate injury risks and maintain cervical mobility, it is recommended that exoskeletons be designed to support the spine at T1 and below. By doing so, exoskeletons can provide the necessary support for posture and movement without compromising the flexibility and function of the upper spinal region.

Key implications for exoskeleton design include:

1. Injury Prevention: Positioning exoskeletons at T1 and below minimises the risk of spinal cord injuries in the fragile cervical region.

2. Maintaining Natural Range of Motion: Avoiding the cervical spine in exoskeleton design preserves natural head and neck mobility, reducing discomfort and biomechanical strain.
3. Ergonomic Considerations: Supporting the thoracic and lumbar regions ensures load bearing without hindering upper spinal flexibility, enhancing long-term user comfort and device usability.

2.7.4 Summary

The cervical spine's anatomical and biomechanical characteristics make it especially prone to injury and strain, underscoring the need for careful consideration in exoskeleton design. Supporting the spine at T1 and below minimises the risk of injury while maintaining cervical mobility and alignment, ensuring that assistive devices can enhance, rather than hinder, natural movements. Further research and development in exoskeleton design should continue to prioritise these considerations to optimise safety and usability. Table 2.5 summarises the key reasons why exoskeleton support should avoid the cervical spine and instead focus on regions below T1, highlighting the implications for preventing injury and maintaining natural biomechanical function.

Table 2.5: Reasons Why Exoskeleton Support Should Avoid the Cervical Spine and Be Positioned from T1 and Below

Reason	Implication	Study
Fragility and Vulnerability	Supporting the cervical spine increases the risk of spinal cord injury due to its delicate structure and mobility.	(Shea et al., 1991)
Alignment and Mobility Impact	Exoskeleton support at the cervical spine can interfere with natural head and neck movements, leading to discomfort.	(Goldschmidt et al., 2019)
Muscle Fatigue and Strain	Cervical exoskeletons can cause neck strain or fatigue due to misalignment with natural cervical biomechanics.	(Giovanelli et al., 2022)

2.8 Actuators for Exoskeletons

Pneumatic actuators, electrical motors, series elastic actuators, hydraulic actuators, and cable-driven systems are commonly employed in exoskeletons, as outlined in Table 2.6. Pneumatic actuators, such as McKibben air muscles (constructed from latex or silicone rubber) and PneuNets bending actuators (made from elastomeric materials), are particularly lightweight and offer smooth operation. However, a limitation of conventional pneumatic systems is their dependence on an external air compressor. Pneumatic actuators are ideal for lightweight applications, while electrical brushless DC motors excel in energy efficiency and control precision. Series elastic actuators (SEAs) are favoured for their ability to control force precisely and absorb shocks, especially in rehabilitation settings. Hydraulic actuators, though powerful and compact, are less frequently used due to the complexity of their systems.

Table 2.6: General Types of Actuators for Exoskeletons

Studies	Actuator	Description	Advantages
Park et al., 2014; Hu et al., 2019; Pardoel & Doumit, 2019	Pneumatic (McKibben/ PneuNets)	- Inflatable inner bladder within a braided mesh. -	Lightweight, adjustable, smooth operation
Renesas2024; Hybart & Ferris, 2022; Hsu et al., 2023	Electrical (Brushless DC)	Converts DC electrical energy to mechanical energy	High efficiency, precise torque and speed control
Junior et al., 2016; Meijneke et al., 2021;	Series Elastic Actuator (SEA)	Motor connected to load via an elastic element	
Tang et al., 2013; Lu et al., 2017	Hydraulic Actuator	Converts hydraulic energy to mechanical energy	Compact, powerful

2.8.1 Pneumatic Actuators

Pneumatic artificial muscles (PAMs), also known as McKibben air muscles, are widely used as pneumatic actuators in exoskeletons. For example, Park et al. (2014) utilised PAMs in an active soft orthotic device to enhance mobility. PAMs consist of an inflatable inner bladder, typically made of latex or silicone, encased in a braided mesh that converts radial expansion into linear contraction.

End clamps secure the muscle, preventing slippage during the contraction and relaxation phases. These actuators can contract up to 25% of their original length, and their force output is influenced by variations in size and stiffness (Soft Robotics Toolkit, n.d.a). Another common pneumatic actuator is the PneuNets bending actuator, which consists of a series of chambers embedded in an elastomer. When inflated, these chambers generate movement, and their behaviour is determined by the geometry of the chambers and the elastomer's material properties (Soft Robotics Toolkit, n.d.b).

Compared to traditional pneumatic cylinders, both PAMs and PneuNets offer the advantages of being significantly lighter and smoother in operation, as they lack sliding mechanical parts. For instance, PAMs weigh approximately 27 g/m (Baiju, 2022), whereas pneumatic cylinders can weigh up to 647 g/m (RS Malaysia, 2020). However, PAMs and PneuNets are also more susceptible to damage due to the thinner materials used in their construction, making them less durable than pneumatic cylinders (Baiju, 2022; Soft Robotics Toolkit, n.d.a). PAMs and PneuNets offer efficient, lightweight solutions for exoskeleton applications, but their durability can be a limitation when compared to more robust pneumatic cylinders, which are also not considered due to high force and linear only actuation. Overall, PAM and PneuNets offer the advantage of being softer and flexible when unactuated, with PneuNets, other pneumatic based soft robotics and several types of PAMs having bending motion capabilities (Guan et al.,2020a).

To further explore the realm of pneumatic based actuators, there are many types of soft pneumatic actuators that go beyond just the normal uniaxial actuation of conventional pneumatic actuators (Chen et al., 2022). A literature review table is constructed in Table 2.7 below discussing the array of non-linear pneumatic-based actuators.

Table 2.7: Literature Review Table: Pneumatic-Based Actuators

Source	Journal	Key Findings
Guan et al., 2020a	Soft Robotics	Proposed bending and helical PAMs inspired by elephant trunks, with a model of generalised bending behaviour. Demonstrated the use of bending in soft robotic manipulators.
Xiao et al., 2021	Smart Materials and Structures	Developed a BPAM with multi-degree freedom, which can bend in 3D space. Demonstrated its applicability in flexible, soft-bodied robots.
Takashima et al., 2011	SICE Annual Conference	A new curved PAM actuator using shape-memory polymer was created, capable of bending upon air inflation with heat-controlling actuation directions.
Guan et al., 2020b	Smart Materials and Structures	Presented nonlinear models of bending extensile and contractile PAMs, used in humanoid hands with improved bending performance.
Saga et al., 2022	Sensors	Developed a smart pneumatic muscle actuator with integrated bend sensors, mimicking the human muscle spindle for accurate bending feedback.
Geng et al., 2011	Applied Mechanics and Materials	Focused on elongation-type PAMs and their bending stiffness, exploring nonlinear relationships between air pressure and bending deformation.

2.8.2 Electrical Motors (Stepper and Servo)

In exoskeleton system design, servo and stepper motors play key roles, each suited to different tasks based on their distinct advantages and limitations. Servo motors are preferred for high-precision, dynamic control, while stepper motors excel in cost-effective, incremental movement applications. This section highlights the key features, pros, cons, and applications of each motor type, supported by literature (Fattah, 2010; Hong-bin et al., 2017).

2.8.2.1 Servo Motors: Precision and Dynamic Control

Servo motors offer continuous, precise control of speed, position, and torque through closed-loop systems with feedback mechanisms such as encoders, making them ideal for complex rehabilitation tasks (Flieh et al., 2017 & 2019). Their high torque and fast response make them well-suited for tasks like walking assistance and dynamic posture correction (Anderson et al., 2019).

1. Energy Efficiency: Some designs use less magnet material, improving energy efficiency and reducing costs (Flieh et al., 2017).
2. Versatility: Servo motors come in several types (AC, DC brushless, synchronous), allowing for a wide range of applications (Krishnan, 1987).
3. Safety: Wireless drives reduce the risk of electrocution and offer better environmental sealing (Jiang et al., 2019).

However, servo motors are also complex and costly due to their need for continuous feedback systems, and can suffer from backlash and thermal sensitivity, especially in miniaturised designs (Sun et al., 2023; Barth, 2000; Krishnan, 1987).

2.8.2.2 Stepper Motors: Simplicity and Cost Efficiency

Stepper motors, operating through discrete steps without the need for feedback, are ideal for simple, incremental movements in more affordable applications (Fattah, 2010). They excel at holding positions without consuming power, making them useful in static load-bearing exoskeleton components (Fu & Ran, 2022).

1. Easy Control: Stepper motors require no complex control systems, reducing costs and simplifying implementation (Harshvardhan et al., 2015).
2. High Torque at Low Speeds: Ideal for automation equipment where low to medium speed is required (Fu & Ran, 2022).

However, stepper motors are limited by mechanical oscillation, step loss, and poor adaptability to varying loads due to their open-loop control, which can cause issues in precision-critical tasks (Wang et al., 2017; Bianculli, 1970). They are best suited for upper-limb exoskeletons and robotic prosthetics that prioritise positional accuracy over dynamic movement (Faghihi et al., 2015).

2.8.2.3 Summary

As a summary, Table 2.8 compares the basic differences of the stepper and servo motor, while Table 2.9 discusses the pros and cons of each actuator.

Table 2.8: Comparative Table of Servo and Stepper Motors

Feature	Servo Motors	Stepper Motors
Control Mechanism	Closed loop with continuous feedback (Flieh et al., 2017)	Open loop, no feedback (Fattah, 2010)
Precision	High precision and dynamic control (Flieh et al., 2019)	Moderate precision for incremental steps (Fattah, 2010)
Torque	High peak torque, fast response (Anderson et al., 2019)	High torque at low speeds (Fu & Ran, 2022)
Cost	Higher cost due to complex control (Sun et al., 2023)	Lower cost, simple control (Harshvardhan et al., 2015)
Energy Efficiency	Energy-efficient designs available (Flieh et al., 2017)	Holds position without power consumption (Fu & Ran, 2022)
Applications	Dynamic rehabilitation exoskeletons (Flieh et al., 2017)	Upper-limb exoskeletons, prosthetics (Faghihi et al., 2015)
Drawbacks	Backlash, thermal sensitivity (Krishnan, 1987)	Mechanical oscillation, step loss (Wang et al., 2017)

Table 2.9: Pros and Cons Table of Servo and Stepper Motor

Motor Type	Pros	Cons
Servo Motors	High precision and torque (Flieh et al., 2019)	Expensive and complex control systems (Sun et al., 2023)
	Fast response for dynamic tasks (Anderson et al., 2019)	Thermal sensitivity, backlash in small designs (Barth, 2000)
	Energy-efficient options (Flieh et al., 2017)	Requires constant feedback, increasing costs (Flieh et al., 2018)

Stepper Motors	Easy control, cost-effective (Harshvardhan et al., 2015)	Prone to mechanical oscillation, step loss (Wang et al., 2017)
	High torque at low speeds (Fu & Ran, 2022)	Low precision, poor adaptability to varying loads (Bianculli, 1970)
	Holds position without continuous power (Fu & Ran, 2022)	Limited by low resolution, reduced torque in multi-rotor setups (Groenhuis et al., 2021)

2.8.3 Series Elastic Actuators and Cable-Driven Systems

Series Elastic Actuators (SEAs) and cable-driven systems are gaining prominence in the fields of physical human-robot interaction, rehabilitation robotics, and exoskeleton design due to their compliance, safety, and flexibility. This subsection synthesises the latest advancements in the design, modelling, and control of these systems, highlighting their growing role in enhancing human-robot interaction. Several studies have developed advanced models and control strategies for cable-driven SEAs. A notable approach involved using a velocity-controlled DC motor as the power source, combined with a two degrees of freedom (2-DOF) control scheme, to achieve robust torque control. This method demonstrated superior performance compared to traditional PD controllers, particularly in applications requiring precision and adaptability. Similarly, another study utilised the 2-DOF control method to effectively separate reference tracking from robustness goals, validating its effectiveness through simulations (Zou et al., 2016).

Besides, trajectory-tracking control in cable-driven upper-limb exoskeletons using SEAs has been a focus of research due to the low inertia and inherent compliance offered by elastic components. One approach combined iterative learning techniques with a model predictive controller, achieving high precision and sensorless force control (Shu et al., 2023). Additionally, impedance control has been improved by targeting specific frequency ranges relevant to human-robot interaction. For example, a $H\infty$ synthesis framework was introduced to optimise stiffness control across multiple frequency bands, ensuring precise and robust performance (Yu et al., 2019). Recent innovations in SEA design have improved their application in various robotic systems. A novel backdrivable cable-driven SEA (BCDSEA) was introduced, incorporating a cable-pulley system and a backspring to enhance backdrivability and dynamic performance (Wang et al., 2019). Another advancement presented a compact rotary SEA with Bowden cables for upper-limb rehabilitation robots. This design featured direct spring displacement measurement and mechanisms to absorb backlash, thus improving overall actuator performance (Zhang et al., 2020).

Control strategies for SEAs have been extensively researched and validated through experiments. For instance, a PD controller optimised using the linear quadratic regulator (LQR) method was applied to a cable-driven SEA, showing improved tracking during sinusoidal movement experiments (Ai et al., 2021). Additionally, a disturbance observer-based torque-mode control algorithm was introduced to address variable friction in Bowden cables, ensuring zero output torque control. This method was validated through human subject experiments, demonstrating its robustness and adaptability (Lu et al., 2015). The integration of SEAs in rehabilitation systems and exoskeletons has been extensively explored. A notable example is the development of a body weight support system using a linear SEA, which provides precise unloading force for gait training. This system demonstrated lower power consumption and more accurate cable force control compared to conventional systems (Mirzaee et al. 2019). In addition, SEAs with Bowden cables have been utilised in exoskeletons to achieve compliant actuation, mimicking the behaviour of biological muscles, thereby improving the flexibility and performance of exoskeleton joints (Zou et al., 2016).

Overall, advancements in SEAs and cable-driven systems have significantly enhanced the design and control of robotic systems, particularly for human-robot interaction and rehabilitation applications. Innovations in modelling, control strategies, and design have made these systems more compliant, efficient, and safe. Ongoing research promises further improvements, particularly in the areas of backdrivability, energy efficiency, and sensorless control, making these systems increasingly adaptable to diverse applications. Table 2.10 summarises both systems.

Table 2.10: Summary Table of SEA and Cable-Based Systems

Aspect	**Series Elastic Actuators (SEAs)**	**Cable-Driven Systems**
Modelling & Control	2-DOF control for robust torque control (Zou et al., 2016)	Trajectory tracking using predictive control (Shu et al., 2023)
Impedance Control	Optimised stiffness control across frequency bands (Yu, Zou, & Sun, 2019)	Sensorless control in cable-driven systems (Shu et al., 2023)
Design Innovations	Backdrivable cable-driven SEAs (Wang et al., 2019)	Bowden cables with backlash absorption (Zhang et al., 2020)
Control Strategies	LQR-optimized PD controller (Ai et al., 2021)	Torque-mode control with disturbance observers (Lu et al., 2015)
Applications	Gait training, lower power consumption (Mirzaee, Moghadam, & Saba, 2019)	Compliant actuation for exoskeletons, mimicking muscles (Zou et al., 2016)
Advantages	High compliance, precise control, low power consumption	Flexibility, cost-effectiveness, sensorless control
Limitations	Complex control strategies, potential for mechanical oscillation	Variable friction and reduced backdrivability in some designs

2.8.4 Soft Robotic Actuators

Soft robotic actuators come in various forms, including pneumatic-based systems, and are integral to the field of soft robotics. General soft robotic actuators include small-scale actuators that range from nanometres to centimetres in size and utilise smart materials responsive to stimuli like heat and light, enabling large deformations and complex motions (Chathuranga et al,, 2022; Sambyal et al., 2023). Fluidic Elastomer Actuators (FEAs) are another common type, utilising pressurised fluids within elastomeric structures to achieve flexible, adaptable movement, particularly useful in delicate environments such as surgery (Boyraz et al., 2018). Electromagnetic soft actuators, which use flexible materials like silicone combined with metals, generate motion through magnetic forces, offering advantages in speed, precision, and compactness (Do et al., 2018).

Specifically focusing on pneumatic-based soft actuators, Pneumatic Networks (Pneu-Nets) utilise pressurised air in elastomer channels to create movement, with recent innovations enhancing efficiency by reducing gas consumption and increasing actuation speed (Mosadegh et al., 2014). Precharged pneumatic actuators, which store pressurised air for untethered operation, modulate motion via tendons, making them compact and self-sufficient (Li et al., 2018). Origami-inspired pneumatic actuators, combining principles of origami with pneumatic designs, offer higher force output and compact storage, making them highly effective in space-constrained robotic applications (Kim et al., 2021). Soft pneumatic actuators have become particularly valuable in tasks requiring delicacy, such as in rehabilitation and object manipulation, where their flexibility and compliance enable safe interactions (Antonelli et al., 2018). In conclusion, the diversity of soft robotic actuators, especially pneumatic systems, underscores their adaptability and effectiveness in generating complex and controlled motions across various applications. Table 2.11 compiles the unique features and applications of different soft robotics actuators.

Table 2.11:Summary Table of Soft Robotics Actuators

Type of Actuator	Key Features	Applications	Source
Small-Scale Soft Actuators	Nanometre- to centimetre-scale soft actuators using smart materials like heat, light, and magnetic fields.	Nanorobotics, biomedical devices, small-scale manipulators.	Mushtaq et al., 2019
Fluidic Elastomer Actuators (FEAs)	Uses pressurised fluid inside elastomers to generate motion; high compliance and adaptability.	Medical devices, soft robotics in delicate environments, rehabilitation.	Boyraz et al., 2018
Electromagnetic Soft Actuators	Flexible actuators using silicone and metal; produces motion through magnetic fields; high speed and precision.	Tactile displays, biomedical devices, soft robotic grippers.	Do et al., 2018

Pneumatic Networks (Pneu-Nets)	Elastomeric actuators that use air pressure in channels to create bending and motion; fast actuation with reduced gas usage.	Soft robotics requiring large amplitude movements with simple controls.	Mosadegh et al., 2014
Precharged Pneumatic Actuators	Actuated by precharged air; controlled by tendons for complex movements; eliminates need for constant air supply.	Untethered autonomous robots, soft grippers.	Li et al., 2018
Origami-Inspired Pneumatic Actuators	Soft actuators with origami-patterned chambers; expandable design for higher force output and compactness when not in use.	Wearable robotics, robotic gloves, soft grippers.	Kim et al., 2021
Soft Pneumatic Hand Actuators	Hyper-elastic silicone actuators for human-robot interaction; mimics human hand movements with high compliance.	Robotic hands for collaborative robotics, rehabilitation devices.	Antonelli et al., 2018
Pneumatic Memory Actuators	Uses air (not electricity) to control the actuation state, enabling reduced hardware for complex soft robot control.	Soft robotic systems like robotic hands and musical instruments.	Hoang et al., 2021
Pneumatic Helical Soft Actuators	Helical chamber design to enable bending and twisting motions with higher force output.	Soft robotic grippers for complex shapes, manipulators.	Hu & Alici, 2020
Shape Memory Alloy (SMA) Actuators	High work density, compliant, and responsive to thermal stimuli; cooling limitations restrict actuation speed.	Soft bioinspired robots, prosthetics, artificial muscles.	Huang et al., 2019
Flexible SMA Actuators	Shape memory alloy wires with enhanced flexibility for wearables; low weight and high force/weight ratio.	Wearable robots, soft exoskeletons.	Copaci et al., 2020

2.9 Systems, Sensors, and Devices for Postural Sway Detection

Postural sway refers to the continuous, natural movement of the body's centre of mass (CoM) or CoG while maintaining a standing position. This subtle motion is a critical indicator of balance and stability and is widely used to assess fall risk, particularly in older adults and individuals with balance impairments. Several methods and parameters are utilised to measure postural sway, each offering unique insights into balance control mechanisms.

The primary parameters used to quantify postural sway include path length and sway range, which measure the total distance travelled by the centre of pressure (CoP) and the extent of sway in different directions (Malaya et al., 2020; Ge et al., 2019; Pollind, & Soangra, 2020a). Sway velocity is another key measure, representing the speed at which the CoP moves and is useful for distinguishing between different balance conditions and populations (Dieën, Koppes, & Twisk, 2010; Pollind, & Soangra, 2020a, Voss et al., 2021). The Root Mean Square (RMS) of sway is frequently used to quantify the magnitude of sway, offering an overall measure of stability (Pollind, & Soangra, 2020a, Voss et al., 2021). Another important parameter is sample entropy (SampEn), which analyses the temporal structure of sway, providing insights into the complexity and predictability of postural control (Malaya et al., 2020). Lastly, the total sway area measures the area covered by CoP movements, serving as an indicator of overall stability (Voss et al., 2021, Degani et al., 2017).

Besides, The Sway Vector (SV) and Directional Indices (DI) are widely recognised as reliable and robust measures for assessing postural stability. These parameters are particularly advantageous because they are independent of trial length and sampling frequency, making them less susceptible to noise and variations in experimental design (Janusz et al., 2016; Błaszczyk, 2016). The use of SV and DI allows for a more nuanced description of postural control, making them effective in distinguishing between different conditions, such as age-related decline and diseases like Parkinson's (Błaszczyk, 2016). Among these measures, the Stability Vector Amplitude (SVamp) and Stability Vector Azimuth (SVaz) offer novel insights into postural control by providing reference values for stable human posture. In healthy, young individuals, SVamp is typically around 9.2 ± 1.6 mm/s, while SVaz is approximately 0.9 ± 0.1 rad. These parameters are sensitive to visual input and have weak to moderate correlations with anthropometric characteristics, indicating their specificity in capturing fine aspects of postural control (Błaszczyk, & Beck, 2023). In clinical populations, such as multiple sclerosis (MS) patients, SVamp and SVaz show pathology-specific increases in sway velocity, particularly during eyes-closed (EC) tests. This increase highlights their potential diagnostic value in tracking disease progression (Błaszczyk, et al., 2021).

Visual conditions also have a significant impact on postural sway. The absence of visual input, as in eyes-closed conditions, tends to increase sway across measures such as SV, Directional Index in Anterior-Posterior (DIAP), and Directional Index in Medial-Lateral (DIML), emphasising the crucial role of vision in maintaining postural stability (Błaszczyk, & Beck, 2023; Janusz et al., 2016; Błaszczyk, et al., 2021). Thus, the Sway Vector (SV), along with Stability Vector Amplitude (SVamp) and Azimuth (SVaz), provides valuable and reliable descriptors for evaluating postural control. These measures are robust to experimental noise and sensitive to visual conditions, making them useful for differentiating between healthy individuals and those with conditions like age-related decline, Parkinson’s disease, and multiple sclerosis. By establishing reference values for stable posture and highlighting the role of visual input in balance, SV and its related measures are crucial tools in assessing postural stability and diagnosing balance impairments.

2.9.1 Technological Methods for Measuring Postural Sway

Several technologies are used to accurately assess postural sway. Force plates are highly sensitive devices that measure CoP movements and provide detailed sway parameters, making them the gold standard in postural sway analysis (Goble, & Baweja, 2018; Degani, et al., 2017; Sturnieks et al., 2011). These platforms are widely used in research to track sway in anterior-posterior and medial-lateral directions, offering high-accuracy measurements, but they are often impractical for non-laboratory settings due to their cost and lack of portability (Sturnieks et al., 2011). To overcome this, a similar method implementing pressure sensors in insoles are developed. Embedded in insoles or mats, pressure sensors detect the distribution of pressure under the feet, providing information on changes in CoG which are suitable for real-time postural analysis in clinical or rehabilitation settings enhancing portability (Walsh et al., 2020). Sway Path Length, Sway Area, and Sway Velocity are key indicators of instability, especially in elderly populations. CoP measures can differentiate balance performance across sensory conditions, such as eyes-open versus eyes-closed trials (Ramdani et al., 2009).

More recently, inertial measurement units (IMUs) have emerged as portable, cost-effective alternatives to force plates. IMUs provide a portable alternative by using accelerometers and gyroscopes to measure linear and angular movement. IMUs offer flexibility for non-laboratory applications, although noise and drift can affect their accuracy, requiring advanced filtering techniques like the Kalman Filter. Neville et al. (2015) validated the effectiveness of portable IMUs for postural sway detection, showing strong correlations with force platforms ($r = 0.79$) and motion capture systems ($r = 0.88$). These wearable sensors can measure key parameters such as sway velocity, RMS, and path length, enabling more accessible assessments (Pollind, & Soangra, 2020a; Voss et al., 2021). IMU systems also track angular displacement and acceleration, which are useful for assessing fall risks. Time to Stabilization (TTS) is another dynamic parameter used to assess how quickly a person regains stability after perturbations (Goel et al., 2022). Kinect motion tracking offers a non-invasive option for estimating CoM sway, using motion capture technology to evaluate postural stability without the need for physical contact (Mazumder et al., 2017).

Optical motion capture systems (OMCS) and magneto-inertial measurement units (MIMUs) are pivotal in tracking body movements and calculating the centre of gravity (CoG) for various applications. OMCS, such as Vicon and OptiTrack, utilise cameras and reflective markers to capture precise kinematic data, predominantly in controlled environments like research labs, though extensive setup is required (Schumann et al., 1995). Conversely, MIMUs, which integrate magnetometers and inertial measurement units, provide accurate CoG tracking in more dynamic, ecological settings, compensating for magnetic disturbances (Germanotte et al., 2021). Lastly, wearable exoskeleton sensors, equipped with strain gauges and load cells, play a critical role in adaptive postural support by providing real-time CoG feedback to maintain stability (Najafi et al., 2010). These diverse technologies cater to different environments and purposes, enhancing the precision and utility of motion capture systems. Table 2.12 summarises the sensors and systems for postural sway detection.

Table 2.12: Sensors and Systems for Postural Sway Detection

System	Mechanism	Application	Study
Force Plates	Measure ground reaction forces to calculate CoG.	Laboratory-based postural analysis.	(Goble et al., 2018)
Pressure Sensors	Measure pressure distribution under the feet.	Real-time postural monitoring.	(Walsh et al., 2020)
Optical Motion Capture	Track body movement using cameras and markers.	Detailed motion analysis in labs.	(Schumann et al., 1995)
MIMUs	Combine IMU and magnetometer data to estimate CoG.	Real-world CoG tracking.	(Germanotta et al., 2021)
Wearable Exoskeleton Sensors	Integrated strain gauges and load cells for force and torque measurement.	Postural correction in exoskeletons.	(Najafi et al., 2010)

Also, electromyography (EMG) is an essential tool for evaluating muscle activity in postural control systems, offering insights into how muscles contribute to correcting postural sway. By detecting electrical signals generated during muscle contraction, EMG helps assess neuromuscular control related to posture. It is frequently combined with other sensors, such as those used to track the centre of pressure (CoP), to provide a comprehensive understanding of postural stability. EMG is widely applied in rehabilitation settings, particularly for real-time monitoring of muscle function in individuals with conditions like stroke or Parkinson's disease. For instance, EMG data has been effectively combined with CoP measurements to analyse how muscle activation impacts postural stability (Warnica et al., 2014).

Postural sway measurements can be affected by several factors. Age and sex are significant determinants, with older adults generally displaying greater sway than younger individuals. Differences in sway performance between males and females are also noted under specific balance conditions (Goble, & Baweja, 2018; Voss et al., 2021). Visual and sensory inputs play a crucial role in balance control, as visual motion sensitivity and binocular visual fields are strongly correlated with postural stability (Wood et al., 2022). Additionally, experimental conditions—such as whether the individual has their eyes open or closed or is standing on a firm or foam surface—greatly influence sway measurements, with more challenging conditions leading to greater sway (Sturnieks et al., 2011).

2.9.2 Reliability and Validity of Sway Measurements

The test-retest reliability of postural sway measurements varies, with many traditional parameters showing low reliability across repeated assessments. This has led to calls for multivariate approaches to fully characterise balance performance (Dieën et al., 2010). Devices like the Swaymeter have demonstrated concurrent and convergent validity when compared to force plates, positioning them as reliable tools for assessing postural sway in both research and clinical settings, demonstrating good agreement with force platforms in measuring anteroposterior and mediolateral sway (Sturnieks et al., 2011).

Force platforms are recognised for high precision in CoP measurements. IMU-based methods, while more portable, depend on the reliability of their algorithms to correct sensor errors. Techniques like the Kalman Filter significantly reduce noise in IMU data, enhancing their validity (Maurer & Peterka, 2005). Thus, methods like the Kalman Filter have become a primary tool for filtering IMU data to improve accuracy by reducing sensor drift. Studies show it can outperform simpler algorithms like the Complementary Filter (McKee & Neale, 2019). Time-frequency analysis methods such as Fast Fourier Transform (FFT) also help identify dominant sway frequencies. These techniques are useful for assessing postural stability and differentiating between healthy and impaired subjects (El-Jaroudi et al., 1996). Besides that, more advanced control models, such as Model Predictive Control (MPC) and COP-Based Controller (COP-BC), are used to mimic the human postural control system, where these models consider sensory noise and neurological time delays, providing a more accurate representation of postural sway compared to traditional methods like the Intermittent Proportional Derivative (IPD) model (Jafari & Gustafsson, 2023). The findings here are compiled in Table 2.13.

Table 2.13: Findings on Methods of Postural Sway Detection

Category	**Study**	**Findings**	**Method**	**Processing**
Force Platforms	(Sturnieks et al., 2011)	Reliable and validated across multiple conditions.	Force platforms, Swaymeter	Not applicable
CoP Analysis	(Ramdani et al., 2009)	CoP parameters effectively discriminate sensory conditions (eyes open/closed).	Force platforms	Sample Entropy
Time-Frequency Analysis	(El-Jaroudi et al., 1996)	Spectral analysis identifies key sway frequencies to differentiate between health and impairment.	Time-frequency analysis (FFT)	Fast Fourier Transform (FFT)
Kalman Filter	(McKee & Neale, 2019)	Kalman Filter minimises sensor drift in IMU-based systems, improving accuracy.	IMUs	Kalman Filter
Postural Control Modelling	(Maurer & Peterka, 2005)	Simulation models of postural control show strong correlation with observed CoP measures in aging populations.	Multidimensional feedback modelling	Optimisation algorithms

2.9.3 Summary

Postural sway is evaluated using a range of parameters, such as path length, sway range, velocity, RMS, SampEn, and total sway area. These parameters are measured through advanced technologies like force plates, IMUs, and motion tracking systems. Age, sensory inputs, and test conditions are critical factors influencing sway outcomes. While traditional methods offer valuable insights, modern tools and multivariate assessments provide a more comprehensive understanding of balance control. Ensuring the reliability and validity of sway measurement tools is essential for accurately evaluating fall risk and balance impairments, particularly in vulnerable populations.

2.10 Postural Monitoring Algorithm for Postural Sway Detection

Postural sway detection is crucial for assessing balance and fall risk, traditionally measured using expensive and non-portable force platforms. Recent advancements have introduced more accessible and cost-effective solutions, such as wearable inertial sensors (IMUs), mobile applications, and virtual reality systems. Low-cost MEMS inertial sensors, including head-mounted and chest-based wearables, have been validated for postural sway analysis, showing high accuracy and portability (Pollind, & Soangra, 2020a, 2020b; Grafton et al., 2019; Meyer et al., 2023). Mobile applications like C3Logix™ offer comparable accuracy to force platforms, making them viable for field use (Miyashita et al., 2020). Neuro-Fuzzy inference systems using Discrete-Wavelet-Transform-based features further enhance stability assessments, even with noisy data (Andò et al., 2022). Additionally, virtual reality systems using HTC Vive trackers and wireless inertial sensors have demonstrated reliability in both clinical and athletic settings, particularly for detecting balance impairments in minimally disabled patients (Liang et al., 2020; Solomon et al., 2015). These advancements make wearable technologies valuable for both clinical and on-field assessments.

Overall, many different methods and algorithms have been developed to detect and analyse postural sway, including threshold-based methods and neuro-fuzzy systems. This review examines these approaches, focusing on their accuracy, reliability, and robustness against noise. The main algorithms or methods focused on are the threshold method and the neuro-fuzzy systems.

Threshold-based methods are simple and widely used, relying on predefined limits for time-based features to detect sway. However, they have limited robustness in noisy environments and show lower accuracy compared to more advanced methods (Andò et al., 2022a). For example, while threshold-based systems can effectively detect unstable postures, they tend to struggle in dynamic or noisy settings. A comparative study highlighted this, showing that threshold-based approaches, though easy to implement, are outperformed by neuro-fuzzy systems when classifying postural behaviours (Andò et al., 2023).

In contrast, neuro-fuzzy systems, which combine neural networks with fuzzy logic, provide more adaptive and noise-resistant detection (Andò et al., 2022a). Studies have demonstrated that neuro-fuzzy inference systems significantly outperform threshold methods, especially when data is noisy or contains variability. For example, using discrete-wavelet-transform (DWT)-based features further enhances the system's ability to detect instability with high accuracy and reliability (Andò et al., 2022a). Another notable advantage of neuro-fuzzy systems is their ability to classify complex postural behaviours with nearly 100% accuracy, distinguishing between stable standing, anteroposterior, and mediolateral sways (Baglio et al., 2023). Furthermore, neuro-fuzzy approaches have also proven highly effective when applied to inertial measurements, achieving a reliability index of around 95% in postural sway assessments (Andò et al., 2022b). This makes them well-suited for real-world applications where noise and variability are common challenges . By contrast, machine learning models have also shown superiority over threshold-based methods, providing greater accuracy and reliability when detecting postural instability. The details are tabulated in Table 2.14.

In summary, while threshold-based methods offer simplicity and ease of implementation, neuro-fuzzy systems are generally more effective for postural sway detection in dynamic, real-world settings. Their higher accuracy, combined with robustness against noise, makes them preferable for applications aimed at reducing fall risks and improving postural stability in clinical and rehabilitation environments.

Table 2.14: Study of Threshold-Based and Neuro-Fuzzy System Based Postural Detection

Study	Method	Key Features	Advantages	Limitations	Results/Conclusions
Threshold vs. Neuro-Fuzzy	Threshold-Based Methods & Neuro-Fuzzy Systems	Time-based feature thresholds; neural networks + fuzzy logic	Simple; easy to implement	Poor noise robustness; lower accuracy than advanced methods	Neuro-fuzzy systems were more robust and effective at distinguishing between stable and unstable postures (Andò et al., 2022a)
Machine Learning vs. Threshold	Threshold-Based & Machine Learning	Comparison between traditional thresholds and machine learning models	Machine learning offers higher accuracy	Threshold methods are less reliable	Machine learning approaches outperformed threshold-based methods in classifying postural sway behaviours (Andò et al., 2023)
Neuro-Fuzzy DWT Features	Neuro-Fuzzy with Discrete-Wavelet-Transform (DWT) Features	DWT-based features fed into neuro-fuzzy inference system	Highly accurate; noise-resistant	Complex implementation	Improved accuracy and reliability in detecting sway instabilities when using DWT-based neuro-fuzzy systems (Andò et al., 2022a)
Postural Behaviour Classification	Neuro-Fuzzy System for Posture Classification	Classification of different sway behaviours (anteroposterior, mediolateral, etc.)	Near 100% accuracy	Noise can still impact performance	Neuro-fuzzy approach showed excellent accuracy for classifying various postural sway behaviours (Baglio et al., 2023)
Neuro-Fuzzy for Inertial Measurements	Neuro-Fuzzy for Inertial Measurement-Based Postural Sway	Inertial sensor data processed using neuro-fuzzy algorithms	Highly reliable for real-world application	Computationally intensive	Superior reliability (~95%) when using inertial measurements for postural sway detection with neuro-fuzzy systems (Andò et al., 2022b)

2.11 Machine Learning Approaches to Postural Sway Classification

Postural sway represents the continuous micro-adjustments executed by the human body to preserve upright stability. Historically quantified using force platforms, contemporary approaches increasingly favour inertial measurement units (IMUs) due to their portability, cost-effectiveness, and real-time monitoring capabilities. The resultant IMU-derived data are inherently noisy, high-dimensional, and time-dependent, rendering them well-suited for machine learning (ML) approaches. These methods have shown growing efficacy in discerning balance profiles across healthy individuals, elderly populations, and patients with neuromotor disorders.

2.11.1 Classical Machine Learning Models for Sway Detection

The application of machine learning techniques to postural sway classification has centred around classical models such as Random Forest (RF), K-Nearest Neighbours (KNN), Support Vector Machines (SVM), and Naïve Bayes (NB). These models have demonstrated resilience to feature noise, the ability to generalize across subjects, and competence in handling non-linear relationships among input features. Table 2.15 summarises such approaches.

Table 2.15: Summary of Classical Machine Learning Models in Sway Detection

Model	Strengths	Limitations	Key Studies
Random Forest (RF)	High generalizability; robust to noise; good with imbalanced data	Requires many trees; can be computationally heavy	Gattinara et al. (2022); Prisco et al. (2025)
K-Nearest Neighbours (KNN)	Simple; good baseline performance	Computationally intensive in real-time; poor scalability	Ozdemir & Barshan (2014)
Support Vector Machine (SVM)	Effective in high-dimensional spaces; well-defined margins	Sensitive to kernel choice; tuning-intensive	Ozdemir & Barshan (2014)
Naïve Bayes (NB)	Fast; interpretable	Assumes independence among features	Less frequently applied in sway detection

2.11.2 Non-Stationarity and Overlap

One of the most persistent challenges in postural sway classification pertains to the reliable identification of the unstable (INST) class. Characterized by irregular, multi-directional fluctuations, INST signals often share spatial and spectral features with more stable sway patterns, thereby confounding many classifiers. Guo et al. (2022) observed that while accelerometer-based features were sufficient to differentiate between stable (STAB) and directional sway classes (DAP, DML), they failed to robustly isolate INST conditions, particularly when subjects experienced fatigue or external perturbations. To mitigate such ambiguities, Andò et al. (2023) introduced a neuro-fuzzy inference framework augmented with adaptive reliability indexes, which significantly improved classification performance in the presence of noise and signal overlap. Likewise, Andò et al. (2023) acknowledged a marked decline in classification accuracy when models trained on mimic trials were applied to real-world datasets, thereby exposing the limitations of conventional ML approaches in unstructured, clinical, or community settings.

2.11.3 Empirical Models versus Data-Driven Learning

Despite the ascendancy of learning-based models, rule-based approaches continue to hold relevance, particularly in embedded or safety-critical systems where interpretability, low latency, and minimal power consumption are prioritized. In static sway classification, threshold-based methods, typically tuned on mean ± standard deviation envelopes of directional features, have been successfully employed to delineate DAP and DML sway patterns. While such models lack adaptability to inter-individual variability, their deterministic nature and transparency are advantageous in contexts where decision traceability is required. Ozdemir and Barshan (2014) further emphasized that, under carefully defined parameters, rule-based classifiers could match or even exceed the performance of more complex models in wearable applications, particularly in fall risk detection scenarios where resource constraints are paramount.

2.11.4 Embedded and Real-Time Classification

The increasing demand for real-time postural sway monitoring has driven a shift toward model optimization and efficient on-device inference. Andò et al. (2024) demonstrated an MQTT-based pipeline enabling real-time IMU signal acquisition and processing on embedded microcontrollers, underscoring the feasibility of low-latency deployments. Nevertheless, the implementation of ML algorithms on resource-limited platforms frequently necessitates model compression, quantization, or the integration of specialized AI inference engines. Parallel developments by Ehara et al. (2025) illustrated the application of gradient boosting frameworks such as LightGBM for estimating joint angles, demonstrating the potential of such models to serve as lightweight alternatives in continuous biomechanical monitoring systems, including those related to postural stability.

2.11.5 Deep Learning and Temporal Signal Modelling

Although classical models remain predominant in wearable postural sway analytics, recent investigations have begun exploring the potential of deep learning architectures, particularly those capable of modelling temporal dependencies. Long Short-Term Memory (LSTM) networks and hybrid architectures such as CNN-LSTM have shown promising results in domains involving dynamic and non-stationary biosignals. Gu et al. (2025) introduced the CLTNet framework, combining convolutional, recurrent, and transformer layers to decode electroencephalogram (EEG) sequences with high temporal resolution. Such approaches are anticipated to offer enhanced performance in postural sway classification, especially for INST detection, where transitions between balance states are gradual and temporally entangled. These deep learning models are designed to capture complex sequential relationships and latent features that static classifiers cannot discern, thereby presenting a promising direction for future sway detection frameworks.

2.11.6 Summary of Literature and Research Gaps

The reviewed literature supports the growing role of ML in sway classification. Nevertheless, unresolved issues include: (1) difficulty in classifying the INST class due to overlapping features; (2) inter-subject variability; (3) reliance on single-sensor IMU configurations; (4) latency/resource challenges in real-time applications; and (5) absence of standardized, labelled datasets for benchmarking models. Table 2.16 summarises machine learning in sway detection.

Table 2.16: Summary of Key Studies Reviewed

No.	Reference	Key Insight
1	Andò et al., 2023	ML accuracy degraded in noisy, real-world sway trials.
2	Andò et al., 2024	Achieved real-time IMU-based sway monitoring via MQTT.
3	Ozdemir & Barshan, 2014	KNN performed well in fall detection; rule-based methods remain viable.
4	Prisco et al., 2025	RF and gradient boosting excelled in IMU-based ergonomic classification.
5	Gattinara et al., 2022	RF outperformed 51 classifiers in Parkinsonian sway detection.
6	Andò et al., 2023	Neuro-fuzzy inference enhanced classification under instability.
7	Guo et al., 2022	Feature-based classification effective for STAB/DAP; weak for INST.
8	Gu et al., 2025	CLTNet outperformed conventional models in decoding biosignals.
9	Ehara et al., 2025	LightGBM enabled low-latency joint-angle regression from IMU data.

2.12 Literature on Exosuit-Induced Muscle Unloading and Postural Control

A range of studies have explored the biomechanical effects of both passive and active exosuits on trunk muscle activation during static and quasi-static tasks. Kang and Mirka (2023a, 2023b) consistently reported that exosuits significantly reduce erector spinae (ES) and rectus abdominis (RA) activation, with unloading effects becoming more pronounced as trunk flexion angles increase beyond 20°. These reductions appear to be robust across symmetric and asymmetric stances, suggesting a generalizable neuromechanical adaptation to external support. Complementing these findings, Cholewicki et al. (2007) found that passive stiffness augmentation via orthoses leads to superficial muscle downregulation, supporting the notion that the central nervous system offloads muscle activity when external stability is provided. Smith et al. (2016) highlighted that excessive abdominal co-activation, particularly in the EO and

RA, can impair postural recovery, especially under visual or proprioceptive challenge, reinforcing the significance of the unloading observed during exosuit use. Importantly, active systems appear to outperform passive ones in muscle unloading capacity. Poliero et al. (2022) reported that an active lumbar support (XoTrunk) achieved approximately 41% EMG reduction in static contexts, compared to 16% in passive systems, underscoring the potential for algorithm-driven control in enhancing unloading efficacy. These findings collectively support the utility of wearable assistive systems, particularly active exosuits, in mitigating trunk muscle fatigue during prolonged static postures, with implications for both ergonomic and rehabilitative settings. Table 2.17 summarises the roles of trunk muscles during exoskeleton support.

Table 2.17: Trunk Muscle Unloading via Exosuit Support

No.	Source	Key Insight	Relevance to This Study
1	Kang & Mirka, (2023a)	Muscle unloading scales with trunk flexion angle; no short-term adaptation observed	Justifies observed EMG reduction trends during leaning or imbalance
2	Kang & Mirka, (2023b)	ES unloading consistent across symmetric/asymmetric postures	Supports generalizability of effect across all test conditions
3	Cholewicki et al., (2007)	Passive stiffness reduces superficial trunk EMG through CNS adaptation	Mechanistic basis for observed unloading in RA and ES-R
4	Smith et al., (2016)	High EO/RA activity impairs postural recovery; their reduction improves balance	Supports balance improvement hypothesis with lower superficial coactivation
5	Poliero et al., (2022)	Active exosuits outperform passive (41% vs. 16% EMG reduction in static tasks)	Validates active system design choice in current exosuit prototype

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This chapter outlines the technical processes and procedures employed in the development of a strap-based, pneumatic-powered, back-supported assistive exosuit for postural sway detection and correction. Specifically, it details the prototype's requirements and conceptual framework, the development process, and the anthropometric data incorporated into the design. Furthermore, the chapter elaborates on the conceptualization of the prototype's mechanical design, its system architecture, and the planned biomechanical assessments designed to evaluate the performance of the exosuit. The discussion also includes the selection of materials and electrical components utilized in the prototype's development, as well as the kinematic methods implemented to obtain postural sway metrics via inertial measurement units (IMUs). Moreover, the circuit design, developed using both breadboard and stripboard techniques, is explicated in detail. The chapter further addresses the design and implementation of a hard-coded graphical user interface (GUI) developed with Visual Studio Code, outlining its architecture and program flowchart. Additionally, this section reviews the libraries employed for programming the prototype components, the GUI system, algorithm development, and data analysis. It also describes the experimental protocols for three tests, sway data collection for classifier algorithm development, sway data validation for algorithm validation, and sway correction for overall prototype testing, implemented to assess the functionality and effectiveness of the exosuit in improving real-time postural sway and balance. Overall, the methodology presented herein offers a comprehensive insight into the project's technical framework and elucidates the rationale behind the key design decisions made throughout its execution.

3.2 Requirement/ Specification of Prototype

The developed prototype is a fabric-based, back-mounted active exoskeleton system designed to support real-time posture correction and reduce standing postural sway. This system was developed to fulfil several functional and design requirements aligned with its intended use in assistive balancing and postural adjustment scenarios. The design emphasises features such as being lightweight, fully portable, semi-concealable, durable, cost-effective, and adjustable to accommodate users of varying body shapes, sizes, and postural needs, in line with UN SDG 3: Good Health and Well-being and UN SDG 10: Reduced Inequalities. Additionally, it must provide accurate sensor readings, enable real-time feedback, and support basic Internet of Things (IoT) connectivity. To support portability and user comfort, the device is constructed using soft materials and compact components, making it lightweight and easy to wear across various environments, including at home or in research settings.

Although some components, such as pneumatic cylinders, remain externally visible, the overall structure is compact and thin, allowing concealment under loose clothing. The modular and adjustable design enhances fit and usability, while also allowing for component replacement or upgrading if needed. Durability is an essential requirement, as the system must withstand repeated use during posture training sessions. Cost-effectiveness is also prioritized, with components selected for affordability without compromising essential functionality. Functionally, the system offers real-time posture monitoring and correction by integrating an IMU to detect trunk motion and postural deviations. Corrective feedback is delivered through pneumatic actuation and is accompanied by immediate visual cues via an onboard OLED display. The pneumatic actuation is low powered and gradual, which biomechanically is in line for the spinal muscles, which are predominantly slow-twitch fibres (Chu, Lin & Chen, 2022; Liu et al., 2020).

A GUI is also implemented to support visual monitoring of system status and sensor data. Basic IoT connectivity allows for wireless data transmission, enabling remote observation and future potential for cloud-based data analysis or control. A known limitation of the system is its operating duration, which is constrained by pneumatic power requirements. The current battery-powered setup allows for approximately 15 to 30 minutes of active operation, which is considered adequate for short training sessions. Power efficiency is recognized as an area for future improvement. Notably, while passive exoskeletons for posture support exist, this prototype addresses a gap in the literature by introducing an active exoskeleton approach aimed specifically at managing postural sway via trunk-based actuation. To the best of our knowledge, no prior systems have employed an active solution targeting trunk strategy for sway correction, much less fabric-based, where most available solutions are passive supports or rigid orthoses, positioning this work as a novel contribution in the early-stage exploration of posture-assistive technologies.

3.2.1 Concept and Features of the Prototype

The back support postural sway exoskeleton is designed to improve balance and minimize postural sway, particularly during periods of quiet stance. By offering dynamic support to the back and trunk muscles, it addresses the needs of trunk stabiliser muscle groups, improving overall posture stability. The system is engineered to accommodate slow-twitch muscle fibres responsible for maintaining posture over time and not fast-twitch fibres, which are engaged in more active movements (Fitts, 1994). Key features of the exoskeleton include multi-axis support, where the exoskeleton can dynamically adjust to both anteroposterior (front-to-back) and mediolateral (side-to-side) sway, allowing the user to move naturally while receiving corrective support. This multi-axis capability enables the exoskeleton to engage when the user's posture deviates, realigning their centre of pressure (CoP) without restricting overall movement.

Besides, real-time postural correction is essential, the system uses with an IMU that constantly monitor the user's kinematic CoP parameters to detect any deviations from normal posture. Upon detecting abnormal sway, the exoskeleton's actuators are triggered to make real-time adjustments, minimizing the risk of falls or instability. This quick response mechanism ensures continuous postural support, especially in static situations where the user's balance might be compromised. Another main feature includes wearability and portability. A major focus in the design of the exoskeleton is ensuring it is both lightweight and comfortable. The frame is constructed from soft, flexible materials such as fabric straps that do not impede movement but provide necessary support. The wearable design ensures that users can go about their daily activities without feeling weighed down or restricted by the device.

A critical aspect of the exoskeleton's design is the postural sway detection system, which continuously monitors the user's CoP parameters and provides feedback for real-time adjustments (Menga & Ghirardi, 2018; Takeda et al., 2017; Layne et al., 2022). This is planned to be achieved through the precise placement of IMU which is positioned on the chest, where it can monitor upper body and trunk motion. This strategic placement ensures that sway is detected during quiet stance and other static postures. The IMUs provide data on acceleration, angular velocity, and orientation, offering a detailed understanding of how the body is moving relative to its CoP (Cinnera et al., 2023; Guidolin et al., 2021).

3.3 Development Process of Prototype

The development of the postural sway back support exoskeleton followed a structured and iterative process involving literature review, resource exploration, laboratory testing, evaluation, and refinement. Both hardware and software elements were addressed systematically to ensure a functional and reliable prototype. An overview of this process illustrated in Figure 3.1, and Gantt chart in Figure 3.2. The process began with conceptual development, where design considerations, key features, and mechanical sketches were created using SOLIDWORKS 2024 software. These guided the prototyping and material selection. During hardware development, mechanical requirements were analysed, and 3D models were prepared. Depending on feasibility, parts were either 3D-printed or fabricated through metalworking. All components underwent testing, with redesigns made as needed. Pneumatic cylinders were selected and subjected to load testing before full assembly. In the circuit development phase, basic circuit functions were tested and integrated incrementally. The MPU6050 sensor module was chosen as the sole sensor for capturing postural sway, providing accelerometer and gyroscope data, to attain and compute kinematic sway data such as displacement in anteroposterior (DAP), and medio-lateral (DML). It was tested for accuracy and integrated into the mechanical system.

For algorithm development, motion data from the MPU6050 was used to develop sway classification algorithms. Key parameters were identified and tested through iteration to achieve acceptable detection performance. These algorithms were implemented into the working prototype. Software development involved creating a modular system for real-time operation.

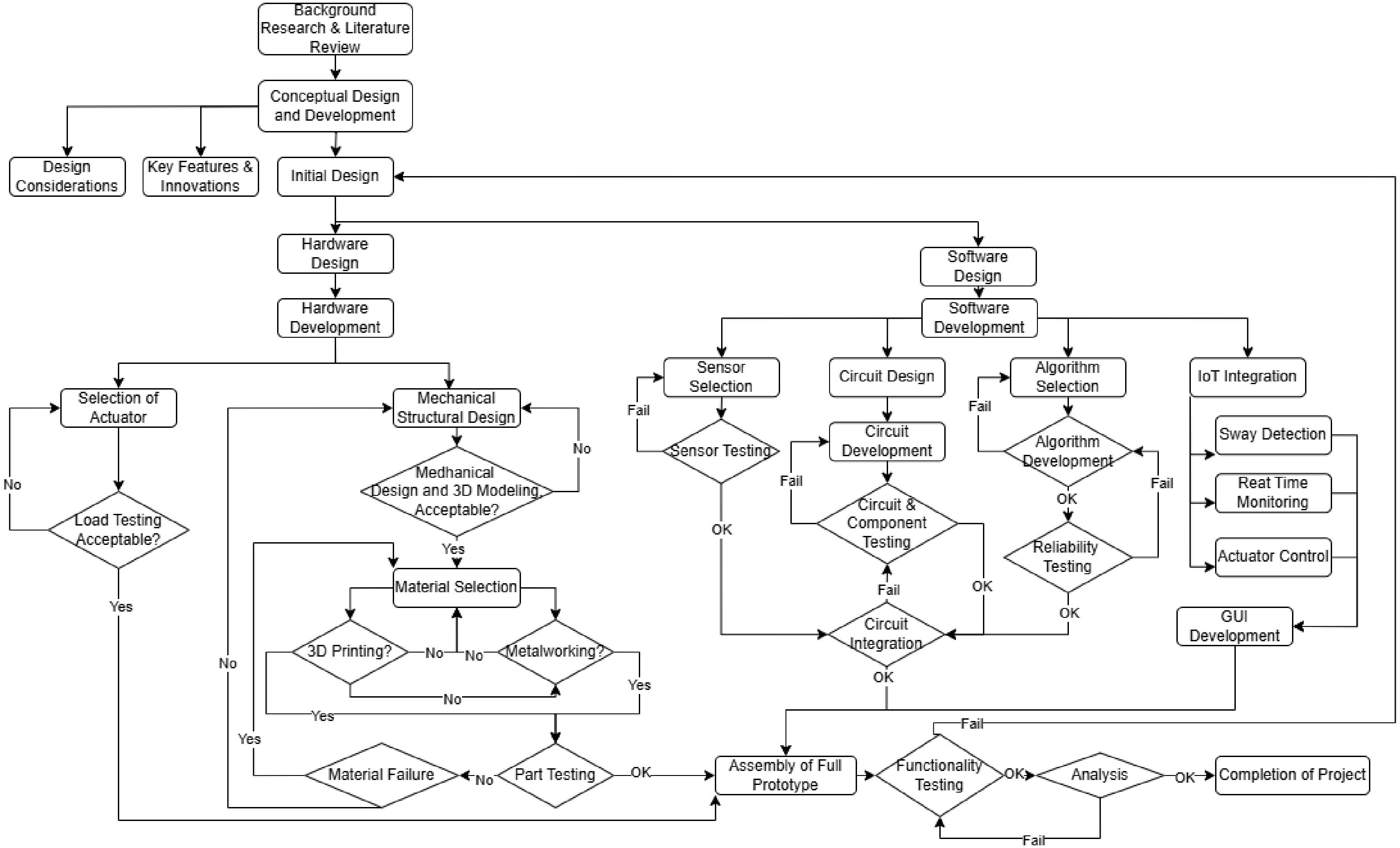


Figure 3.1: Flowchart of Conceptual Development and Methodology of Project

PROJECT PART 1													
Task	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**	**W8**	**W9**	**W10**	**W11**	**W12**	**W13**
Phase 1 Project Preparation													
Project Title Formation	■	■											
Scope Development	■	■	■										
Setting Objective	■	■	■										
Background Study	■	■	■	■									
Phase 2 Literature Review													
Postural Sway Types and Causes		■	■	■	■	■	■						
Current Treatments		■	■	■	■	■	■	■					
Existing Back-Support Exoskeleton		■	■	■	■	■	■	■					
Phase 3 Project Planning													
Initial Solution Development			■	■	■	■	■	■	■	■	■	■	
Cost Estimation and Budgeting								■	■	■	■	■	■
Material and Resource Selection								■	■	■	■	■	■
Risk Assessment Analysis											■	■	■
Selection of Actuator			■	■	■	■	■	■	■	■	■	■	■
Selection of Electronic Components					■	■	■	■	■	■	■	■	■
Phase 4 Conceptual Development													
Hardware Design and Technical Drawing									■	■	■	■	■
Structural Design									■	■	■		

Technical Drawing (CAD)									▨	▨	▨	▨	
Technical Drawing (SolidWorks)									▨	▨	▨	▨	▨
System / Algorithm Development									▨	▨	▨	▨	▨
PROJECT PART 2													
Task	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**	**W8**	**W9**	**W10**	**W11**	**W12**	**W13**
Phase 1 Circuit and Algorithm Development													
Sway Detection Circuit Development	▨	▨	▨	▨	▨								
Algorithm Development	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨			
Algorithm Selection	▨	▨	▨	▨	▨								
Preliminary Data Collection			▨	▨	▨	▨							
Parameter Selection			▨	▨	▨	▨	▨	▨					
Algorithm Testing and Validation						▨	▨	▨	▨	▨			
Development of Peripheral Circuits			▨	▨	▨	▨	▨						
Single Component Testing			▨	▨	▨								
Features Additions				▨	▨	▨							
Integration of Circuit				▨	▨	▨	▨						
Phase 2 Prototype Development													
Hardware Development	▨	▨	▨	▨	▨	▨	▨	▨					
Structural Design	▨	▨	▨										
Technical Drawing (CAD)	▨	▨	▨	▨									
Technical Drawing (SolidWorks)	▨	▨	▨	▨	▨								

	3D Printing and Finishing				▨	▨	▨	▨						
	Metalworking						▨	▨						
	Textile Assembly		▨	▨	▨	▨	▨	▨	▨					
Prototype Assembly					▨	▨	▨	▨	▨					
	Circuit Integration on Hardware				▨	▨								
	Strap and Textile Assembly				▨	▨	▨	▨						
	Complete Assembly						▨	▨	▨					
Prototype Iterations						▨	▨	▨	▨					
GUI Development								▨	▨	▨	▨			
Phase 3 Data Collection and Functionality Testing														
Protocol Development					▨	▨	▨	▨	▨					
Subject Recruitment							▨	▨	▨	▨	▨			
Data Collection of Functionality Tests									▨	▨	▨	▨		
	EMG Tests								▨	▨	▨	▨		
	CoP Tests								▨	▨	▨	▨		
Phase 4 Data Analysis														
Data Analysis											▨	▨	▨	▨
System Evaluation													▨	▨

Figure 3.2: Gantt Chart of Project

Motion data and feedback were displayed on an OLED screen attached to the device for immediate user access. Additionally, a GUI was developed for IoT-based remote monitoring, allowing external users to view sway status and system activity. In the final integration and testing phase, the complete prototype was evaluated for functionality, accuracy, and user interaction. Test results informed final refinements to ensure the system met performance expectations. This iterative approach enabled systematic improvement at each development stage. A summarised and simplified workflow of the prototyping is presented in Figure 3.3 below.

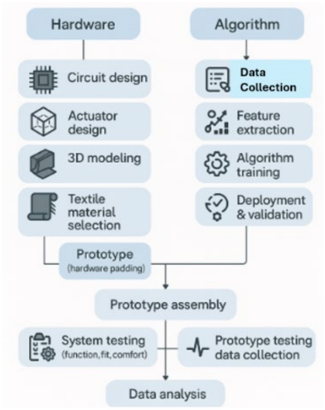


Figure 3.3: Simplified Flowchart of Prototype Development

3.4 System Architecture

The system architecture, illustrated in Figure 3.4, is centred around a control unit, the ESP32 microcontroller, which controls and processes all sensor inputs and actuator outputs. IMU sensors capture real-time kinematic sway parameters that reflect postural sway during standing, which are transmitted to the microcontroller for processing. The ESP32 is powered by a portable power bank, enabling the system to function independently without relying on a fixed power supply, allowing for mobile and wearable applications. Based on the processed sensor data, the microcontroller controls an actuation system comprising a pair of pneumatic cylinders. These cylinders apply linear force to pull adjustable straps that correct the user's posture, effectively mitigating anterior and lateral sway. The postural adjustments block diagram is illustrated in Figure 3.18 and explained in Section 3.6.1. This corrective action aims to minimise overall postural sway and enhance standing balance. A GUI, developed using StreamLit in Python and deployed through Visual Studio Code, provides a user-friendly platform for remote monitoring and interaction. Users can view real-time status updates, log and review their historical sway data, and get interactive visualizations. Additionally, an admin interface enables authorised personnel to access aggregated user data and add relevant comments for monitoring or therapeutic feedback. Complementing the remote interface, an OLED display mounted on the user's wrist provides immediate visual feedback on sway parameters and balance in real time. This integration of IMU sensing, pneumatic actuation, and IoT-based control and monitoring establishes a responsive and intelligent system capable of providing real-time postural correction and feedback, supporting both user autonomy and potential clinical oversight.

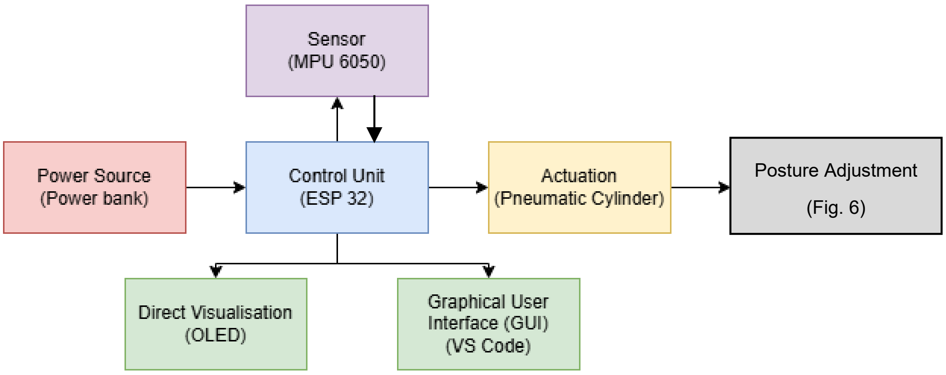


Figure 3.4: System Architecture of Prototype

3.5 Biomechanics Assessments and Considerations of Prototype

This section outlines the biomechanical rationale, analytical methods, and performance metrics used to evaluate the functional outcomes of the proposed posture-correcting exoskeleton prototype. The assessment framework integrates quantitative analyses of standing postural sway, based on Centre of Pressure (CoP) dynamics, and surface electromyographic (EMG) signals from key trunk musculature. The parameters and methods selected reflect established standards in biomechanical research and are intended to validate both the mechanical effectiveness and physiological relevance of the prototype.

The structural and functional design of the exoskeleton is informed by anthropometric and kinesiological principles to preserve user comfort, range of motion, and biomechanical efficiency. The device is intended to assist with postural control during quiet standing by supporting critical regions involved in balance maintenance, particularly the lumbar spine and pelvis. The anatomical focus includes the erector spinae, multifidus, internal and external obliques, and rectus abdominis, which are responsible for trunk stability and alignment. The mechanical support provided by the exoskeleton is designed to complement rather than replace muscular effort. Therefore, biomechanical evaluation focuses on whether the system can reduce excessive postural sway and alleviate muscular workload without restricting natural movement patterns. Table 3.1 presents a summary of the prototype's biomechanical objectives and corresponding design considerations.

Table 3.1: Biomechanical Functionality of the Proposed Exoskeleton

Aspect	Description
Postural Stability	Assists in minimizing CoP excursion via feedback and correction of sway.
Muscular Demand	Reduces trunk muscle activation during prolonged standing.
Joint Movement	Maintains physiological joint alignment and range of motion.

3.5.1 Standing Postural Sway Centre of Pressure (CoP) Parameters

CoP displacement serves as a fundamental biomechanical indicator of postural control and balance performance. In this study, CoP-related metrics were estimated from kinematic data acquired at the trunk level via IMU at the chest. The IMU provides three-axis acceleration signals, which can be processed to infer angular displacement and corresponding linear sway in both the sagittal and frontal planes. The orientation of IMU as well as the basic sway parameters such as anteroposterior displacement (DAP), mediolateral displacement (DML), pitch angle and roll angle, are depicted in Figure 3.5 which were adapted from multiple publications (Andò et al., 2022; Nehary, Rajan and Ando, 2024).

The inclination angles of the trunk in the sagittal and frontal planes, denoted as pitch and roll respectively, are computed from triaxial acceleration measurements and can be used to determine the orientation angles of the standing posture of user. These angles serve as proxies for anterior-posterior and mediolateral sway. These computations offer a simplified yet reliable means of quantifying postural orientation without requiring a full motion capture setup and can be determined by Equations 3.1 and 3.2:

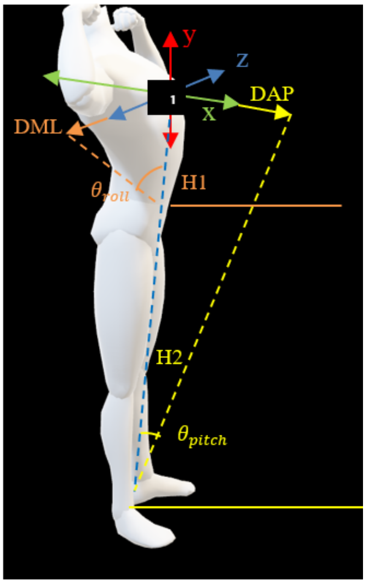


Figure 3.5: Equivalent IMU Nodes Positions on User and Representation of Basic Quantities for Reconstructing the AP and ML Dynamics

$$\theta_{pitch} = \arctan\left(\frac{A_z}{\sqrt{A_y^2 + A_x^2}}\right) \tag{3.1}$$

$$\theta_{roll} = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \tag{3.2}$$

where

A_x , A_y , A_z represent the acceleration components along the mediolateral, vertical, and anteroposterior axes, respectively. The resulting angles describe trunk inclination relative to gravity and provide input for estimating sway displacement.

Assuming the trunk rotates about a fixed base, the horizontal displacements of the CoP, which are the DAP and DML, can be estimated by projecting the pitch and roll angles over fixed vertical heights, H_1 and H_2, depicted in Figure 3.6, via Equations 3.3 and 3.4:

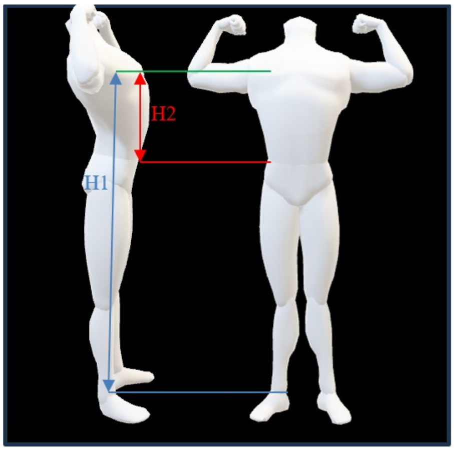


Figure 3.6: Representation of H_1 and H_2 to Reconstruct AP and ML Dynamics

$$DAP = \mathrm{H}_1\left(\frac{A_z}{\sqrt{A_y^2 + A_Z^2}}\right)$$

$$DAP = H_1 \tan(\theta_{pitch}) \tag{3.3}$$

$$DML = \mathrm{H}_2\left(\frac{A_z}{\sqrt{A_y^2 + A_Z^2}}\right)$$

$$DAP = H_2 \tan(\theta_{roll}) \tag{3.4}$$

where

DAP and DML represent the anterior-posterior and mediolateral displacements, respectively.

H_1 and H_2 denote the vertical distance from the IMU sensor to the assumed CoP, upper chest to ankle for H_1 and upper chest to waist for H_2.

These displacements are interpreted as estimations of postural sway at the CoP level and serve as the foundation for all subsequent sway metrics.

Next, the standard deviation of sway, where the variability of sway in each direction is calculated as the standard deviation of DAP and DML over a sampling period. These values provide a direct measure of postural stability. Greater standard deviations indicate increased sway and are generally associated with decreased postural control or increased neuromuscular effort, which can be computed with Equations 3.5 and 3.6:

$$S.D._{DAP} = \sigma_{AP} = \sqrt{\frac{1}{N-1}\Sigma_{i=1}^{N}\left(DAP_{(i)} - \overline{DAP}\right)^2} \tag{3.5}$$

$$S.D._{DML} = \sigma_{ML} = \sqrt{\frac{1}{N-1}\Sigma_{i=1}^{N}\left(DML_{(i)} - \overline{DML}\right)^2} \tag{3.6}$$

where

N = total number of samples

$\overline{DAP}$ and $\overline{DML}$ are the mean displacements

Following, mean sway velocity quantifies the rate of CoP displacement and reflects the dynamic behaviour of balance corrections. Higher sway velocities may indicate increased postural instability or compensatory movement patterns. The mean sway velocities of AP and ML axes can be computed via Equations 3.7 and 3.8:

$$Mean\ Velocity_{DAP} = \frac{1}{T}\Sigma_{i=1}^{N-1}\frac{\left|DAP_{(i+1)} - DAP_{(i)}\right|}{\Delta t} \tag{3.7}$$

$$Mean\ Velocity_{DML} = \frac{1}{T}\Sigma_{i=1}^{N-1}\frac{\left|DML_{(i+1)} - DML_{(i)}\right|}{\Delta t} \tag{3.8}$$

where

Δt = sampling interval

T is total trial duration

Path length is also an important parameter, which is the total path traversed by the estimated CoP trajectory, commonly depicted in the form of a stabilogram as depicted in Figure 3.6 (Ramachandran and Yegnaswamy, 2010). This parameter reflects the cumulative distance of sway, associated with balance control, computed as Equation 3.9:

$$Pathlength = \Sigma_{i=1}^{N-1}\sqrt{DAP^2 + DML^2} \tag{3.9}$$

The area of the prediction ellipse provides a bivariate measure of sway dispersion, as depicted in Figure 3.7. A reduced PEA during exoskeleton usage implies enhanced postural control, as sway is more tightly regulated. Assuming a Gaussian distribution of DAP and DML values, the area enclosing 95% of the sway trajectory is calculated as shown in Equation 3.10:

$$PEA_{95\%} = \pi \times a \times b \tag{3.10}$$

$$a = PSF^2 \cdot \sigma_{AP}, \qquad b = PSF^2 \cdot \sigma_{ML} \tag{3.10.1}$$

where

k = PSF=2.4477 is the prediction scaling factor for 95% coverage in this case.

a and b represent the semi-major and semi-minor axes of the ellipse.

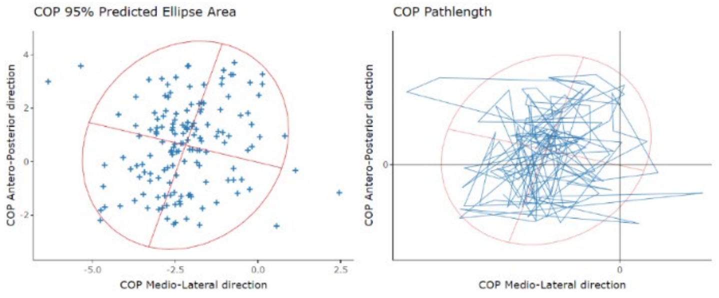


Figure 3.7: Sample Visualisation of $PEA_{95\%}$ and CoP Pathlength

It is important to note that the $PEA_{95\%}$ was chosen over the $CEA_{95\%}$ due to fundamental conceptual and statistical differences between the two. The PEA represents the region within which future individual CoP observations are expected to fall, providing a direct measure of overall postural sway dispersion. In contrast, CEA estimates the confidence region around the mean position of the CoP, and its area shrinks with increasing sample size due to reliance on sample variance and covariance matrices. As highlighted by Schubert and Kirchner (2014), this distinction is critical in posturography, CEA is based on inferential statistics of the mean, while PEA better reflects the true extent of sway variability experienced by an individual. PEA also avoids distributional assumptions such as bivariate normality and is more suitable for real-time applications and threshold-based postural classification, especially in embedded systems. Furthermore, Paillard and Noé (2015) emphasize the practical relevance of using sway area measures that directly reflect CoP dispersion rather than confidence around the mean, particularly in balance training and clinical contexts. As such, PEA offers a more appropriate and interpretable metric for evaluating balance control performance, especially in applied biomechanics and wearable system studies. Therefore, based on both theoretical justification and methodological recommendations in recent literature, PEA was selected as the primary metric for postural sway analysis in this study.

3.5.2 Trunk Stabiliser Muscles Electromyography (EMG)

To assess the biomechanical impact of the proposed exoskeleton on postural control, surface electromyography (sEMG) was conducted on four trunk stabiliser muscles, right rectus abdominis (RA), right external oblique (EO), and bilateral erector spinae (ES) at the L3–L4 level. These muscles were selected for their critical roles in maintaining upper body posture, contributing to spinal stability, and responding to trunk sway and perturbations in both sagittal and frontal planes and are depicted in Figure 3.8. The rectus abdominis and external oblique represent key components of the anterior abdominal wall, generating intra-abdominal pressure and counterbalancing spinal extensor activity. These muscles are particularly relevant in mediating forward sway and maintaining thoracolumbar alignment. The erector spinae, as primary spinal extensors, provide segmental stiffness and control over lumbar posture, especially during posterior sway and corrective movements. Together, these muscles constitute the active core stabilisation system, and their recruitment patterns provide insight into neuromuscular compensation during upright stance.

The inclusion of trunk muscle sEMG was justified based on both biomechanical and functional grounds. Although postural sway during quiet standing is traditionally attributed to distal (ankle) strategies, some studies suggest that proximal (hip and trunk) contributions become more relevant under certain conditions such as fatigue, instability, or constrained lower-limb feedback (Saffer et al., 2008). Importantly, the exoskeleton system developed in this study targets upper body sway via a chest-mounted IMU and delivers corrections based on thoracic displacement, making proximal muscle activity more relevant to its evaluation. This direct influence on trunk kinematics further supports the decision to focus on core musculature rather than distal control systems. Moreover, prior evidence indicates that increased postural difficulty or fatigue leads to elevated EMG activity in trunk muscles (Nakao et al., 2017), and that trunk muscle fatigue significantly alters CoP dynamics, especially CoP velocity, even in asymptomatic individuals (Ghamkhar & Kahlaee, 2019). These changes are especially important when considering populations at risk for chronic low back pain (CLBP), where compensatory overactivation of trunk stabilisers has been observed (Ringheim et al., 2015), though increased EMG alone is not a definitive marker of pathology.

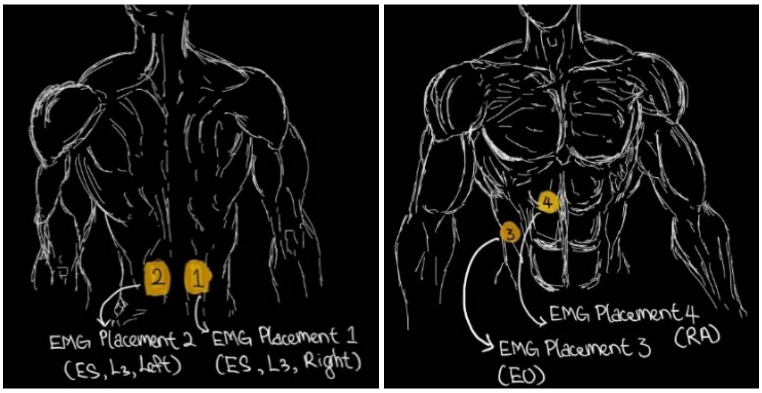


Figure 3.8: Placements of sEMGs on Trunk Stabiliser Muscles Considered

Given this context, the current analysis prioritised trunk muscles over ankle strategies to directly evaluate whether the exoskeleton reduced upper-body sway and muscle effort. Ankle musculature was not instrumented, as it was not the target of feedback or actuation. Furthermore, postural sway was assessed from the thoracic level, and thus muscular compensation at the trunk was more indicative of system effectiveness in mitigating sway. The hip joint's role, though biomechanically relevant, was also excluded from EMG assessment due to variability in recruitment strategies across individuals (Saffer et al., 2008) and lack of direct exoskeleton intervention at that level. This approach ensures that muscle activity data aligns with the functional objective of the exoskeleton, to stabilise the trunk by minimising excessive sway, and provides clear evidence of neuromuscular adaptation or unloading in response to device assistance.

3.5.3 Anthropometric Data Considerations and Calculations for Design

The biomechanical configuration of the exoskeleton was meticulously developed based on established anthropometric datasets to ensure ergonomic compatibility with average human body proportions. Key structural parameters, particularly spinal segment lengths, were incorporated to enable optimal alignment, comfort, and functionality. Rather than relying on individual vertebrae, the design considered functional spinal segments, which offer more biomechanically relevant divisions for wearable systems integration (Ko et al., 2004). Of particular importance was the determination of the exoskeleton's anchor point along the thoracic spine. Although the C7 segment, located near the base of the neck and at the level of the trapezius prominence, might appear suitable in terms of accessibility, it was deliberately avoided. The cervical region is anatomically more fragile and susceptible to injury due to its relatively lower load-bearing capacity and higher mobility demands. Thus, for both safety and biomechanical robustness, the exoskeleton's anchoring interface was positioned inferiorly at the T7 segment, corresponding approximately to the level of the inferior angles of the scapulae. This placement provides a stable foundation for load distribution while preserving cervical spine mobility and reducing risk of strain or impingement. Furthermore, the T7 level aligns with the thoracic pivot point of postural control, making it an ideal location for monitoring and mitigating trunk sway through sensor feedback and actuation. All dimensional parameters and proportional calculations used in the system's design, including segmental lengths and torso landmarks, are summarised in Table B-1, with reference to normative anthropometric data (Ko et al., 2004).

According to existing research, the torso-to-height ratio is generally around 30% or approximately 1/3 of the stature (Hall, 2012; Ramachandran et al., 2016). While different ethnicities exhibit deviations in these segment lengths, where Asians tend to have relatively longer torso and shorter limbs (Liu et al., 2020), the design assumes a torso ratio of 1/3 of total height for simplicity and practicality. In addition to torso length, other critical anthropometric measurements, such as shoulder breadth and hip breadth, are used to develop a modular exoskeleton that can accommodate a wide range of users, as depicted in Table B-2 (Gordon, 2006; Choi-Rokas and Garlie, 2014). These measurements are typically computed based on percentile data, allowing the design to be inclusive and adaptable. Although much of the data originates from American populations, where body dimensions tend to be about 5% larger than those of Malaysians on average, the exoskeleton's design has been recalibrated to better fit the average Malaysian body size. For instance, the median height of Malaysian males is approximately 1.68 m, while for females it is 1.57 m, both corresponding to the 20th percentile of U.S. stature data (Bong et al., 2012). To ensure inclusivity, the design range covers female heights from 1.53 m to 1.7 m (5th to 85th percentile) and male heights from 1.63 m to 1.84 m (5th to 90th percentile), allowing the exoskeleton to cater to a broad user base. In Table B-1 is a summary of anthropometric data tailored to Malaysian users. The application of these data is discussed in more detail in Chapter 3.6. Table B-2 presents a variety of anthropometric measurements for typical Malaysian males and females, with data on height, shoulder breadth, torso length, and other key physical characteristics to design a back support exoskeleton for trunk-based method of postural sway correction. It includes the mean and standard deviation (SD) for both genders, also percentile ranges for certain parameters, particularly relevant for product design and ergonomic considerations. The data illustrates how male and female body dimensions differ in key areas, which should be considered in design applications to accommodate a broad range of users.

3.6 Hardware Development

The development of the mechanical designs began with a comprehensive analysis of anthropometric data, focusing on torso dimensions and girth measurements. Statistical methods and mathematical modelling were employed to select key data points, ensuring the design parameters would be compatible with a wide range of users. The resulting designs were carefully tailored to accommodate varying statures and body sizes, enhancing the system's adaptability and usability. A back-based actuating system, featuring detachable anchors, was incorporated to provide modular support for different users.

The design process progressed through several stages, beginning with initial sketches and detailed technical drawings, followed by 3D modelling and prototyping. Software such as SOLIDWORKS and AutoCAD played a crucial role in creating precise digital models of the designs. Existing models from platforms like GrabCAD were referenced to accelerate the design process, with additional support from online tutorials. Multiple design iterations were conducted, allowing for continuous refinement and optimization. This iterative approach addressed design flaws, improved functionality, and enhanced the overall performance of the system. Throughout this process, meticulous documentation was maintained, ensuring transparency and reproducibility, which will facilitate future improvements and further developments.

3.6.1 CAD and Iterations

In developing the mechanical architecture of the back support exoskeleton aimed at mitigating postural sway, extensive emphasis was placed on constructing a bioinspired, structurally robust, and ergonomically adaptive framework. The exoskeleton frame was conceptualized through iterative design cycles, integrating principles from biomechanics, human-centred design, and modular engineering. The resulting structure comprises multiple interlocking and adjustable components, configured with bilateral symmetry to support functional balance and mechanical alignment. Comprehensive design schematics and assemblies, presented in Figures 3.10 to 3.17 and Figures C-1 to C-11 in Appendices, illustrate the full assembly and modular interconnections of the system. A key design objective was to ensure anthropometric adaptability and user comfort across a broad range of body sizes. This was achieved through adjustable components that can be fine-tuned to match the user's individual anthropometric dimensions, including torso length, shoulder width, and hip girth. By adopting a customizable fitting system, the exoskeleton maintains intimate contact with the user’s trunk while minimizing pressure points and ensuring consistent biomechanical alignment.

The mechanical structure draws inspiration from conventional back braces and safety harnesses, incorporating these principles to inform both actuator placement and force transmission pathways. The actuators are positioned in alignment with the anatomical paths of trunk-stabilizing muscle groups, particularly the ES and RA, which play a vital role in maintaining posture and counteracting excessive sway. This anatomical alignment ensures that assistive forces are applied efficiently, targeting regions of the trunk that contribute most significantly to midline stability. The layout was meticulously designed to avoid interference with natural joint articulations such as the shoulders and pelvis, thereby allowing unimpeded movement while ensuring effective postural correction. The 3D modelling process was conducted using industry-standard CAD platforms, including SOLIDWORKS and AutoCAD, to develop structurally sound and biomechanically informed components. The spine-aligned backplate, thoracic support, actuator mounts, and harness sub-assemblies were digitally prototyped to simulate natural spinal movement while delivering corrective mechanical assistance. Materials were carefully selected for these components: 3D-printed PLA or PLA+ was considered for initial prototypes due to its favourable balance between fabrication speed and mechanical durability, while aluminium was proposed for higher-load elements due to its superior strength-to-weight ratio.

The prototype frame is subdivided into four key mechanical subunits, the waistband assembly, which provides foundational support and anchors the lower structure; the thoracic backplate, that stabilizes the upper spine and acts as the main mounting structure for actuators; the harness system which encircles the chest and shoulders, maintaining actuator alignment and securing upper-body integration; and the actuator-holding assembly, accommodating pneumatic or servo actuators, optimized for force transmission along biomechanical vectors. To maintain dynamic freedom and user safety, the system incorporates multiple anchor points and pivot allowances, via strappings in contrast to rigid designs, ensuring that movement is guided, not restricted, across natural degrees of freedom. Strapping components were fabricated from high-durability woven textiles such as nylon, and their design was informed by the anthropometric data provided in Section 3.5.3. These straps support both load transfer and fit customisation. The data were simplified to prioritise key variables that directly affect strap routing and part integration, including Shoulder Width (SW), Chest Breadth (CB), Torso Length (TL), and Hip Girth (HG). These parameters were critical in defining the angles and lengths of the strap connections, as well as the placement of carabiners and padding.

To standardize fit and functionality, a dataset representing the 5th to 95th percentile of adult body sizes was analysed. The design of the strap system is guided by anthropometric parameters simplified for practical implementation and illustrated in Figures 3.10 and 3.11. The following ranges in Table 3.2 were established based on normative anthropometric data. To determine optimal strap angles, a simplified trigonometric model was applied, as depicted in Figure 3.9. The designed trapezius clearance (TC), measured from the backplate to the base of the neck and trapezius region, was defined in the range of 100–201 mm, based on anatomical data summarized in Table B-1. This ensures safe and unimpeded shoulder movement as well as lower risks of injuries bypassing the more fragile cervical spine area for more neutral, load bearing T7. The relationship is defined by Equation 3.11.

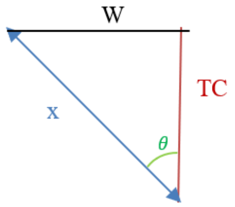


Figure 3.9: Simplified Trigonometric Model of Strapping Angle

$$\theta = \arctan\left(\frac{\text{Trapezius Clearance } (TC)}{0.5 \times W}\right) \tag{3.11}$$

where W is the acromial-to-acromial length, and $0.5 \times W$ simplified as clavicular length (CL). Thus, the computed effective range for strap angle, θ is 30° and 55°.

Table 3.2: Table of Anthropometric Data Analysed for Prototype Design

Parameter	**Range**
Body Height (BH)	1.53–1.88 m
Torso Length (TL)	30%–33.33% BH (~47–65 cm)
Shoulder Width (SW)	41–61 cm
Chest Breadth (CB)	30–38 cm
Shoulder-to-Waist (S-W)	30–42 cm
Hip Width (HW)	30–38 cm
Waistband Circumference	83–110 cm

The shoulder clearance offset from the spine was calculated to estimate required length of straps exceeding that of the torso length, using Equation 3.12:

$$x = \frac{CL}{\tan\theta} \tag{3.12}$$

with resulting values between 140–250 mm. To accommodate approximately 95% of the target population based on standard anthropometric datasets, the shoulder-to-pelvis length was assumed to range between 470–650 mm, while the shoulder-to-waist segment specifically falls within 420–550 mm.

A representative CAD design of the strap attachment configuration and routing strategy is presented in Figure 3.11, detailing the anchoring points, angular routing of the straps, and integration interfaces with the actuator modules. The corresponding physical realization of this system is shown in Figure 3.10, which validates the design through practical alignment with the CAD model and demonstrates its adjustability, structural integrity, and user comfort during wear. Chest depth estimations were derived based on 10%-20% of the vertical torso circumference, resulting in a range of 150-270 mm. To accommodate spinal curvature and individual variations, strap lengths were extended by an additional 10%-20 % of torso length and supplemented with 50-100 mm of slack. This resulted in an effective strap length of 550-900 mm, which was further adjusted, with padding and hardware allowances, to a standardized final length of approximately 1300 mm. The lower strap sections, ranging from 220-440 mm, were designed in accordance with the estimated torso length of 470-650 mm, factoring in 100-201 mm for shoulder clearance, an approximate 100 mm allocation for the backplate, and an additional 100 mm for the waistband structure.

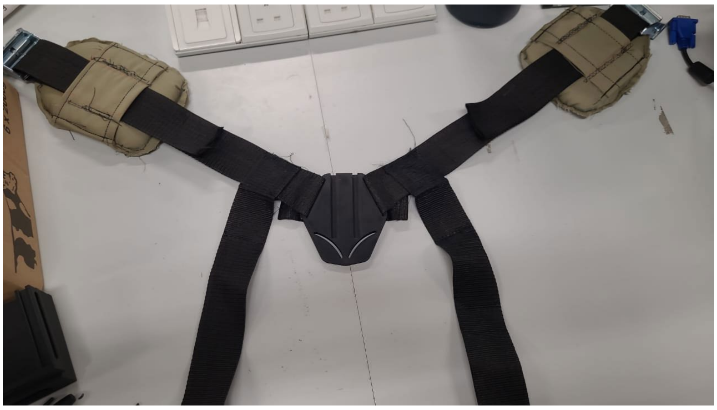


Figure 3.10: Actual Strapping Strategy of Prototype

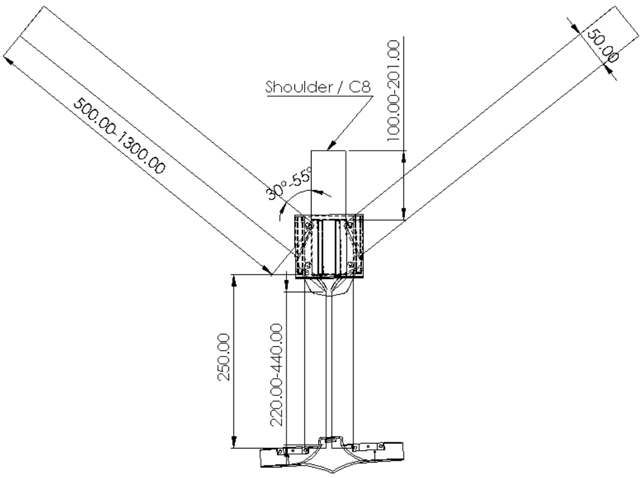


Figure 3.11: CAD Design of The Strap Attachment Configuration

To facilitate accurate detection of upper trunk postural sway, a custom-designed enclosure was developed for chest-mounted MPU6050 IMU module. The casing is compact and tailored specifically to accommodate the physical dimensions of the IMU, including the L-shaped header pins, measuring approximately 20 mm × 15 mm × 11 mm. The enclosure itself has a total height of only 15 mm, ensuring that it remains low-profile and easily concealable under typical clothing such as a t-shirt, sweater, or jacket, making it suitable for continuous wear in both experimental and real-world conditions. The enclosure features two M3 screw holes that allow secure fastening of the MPU6050 to the case using screws or bolts, thereby preventing sensor displacement during movement. Additionally, strategically positioned side openings enable the passage of 20 mm elastic straps. These straps are routed through the enclosure and fastened using ladder buckles, allowing the entire assembly to be securely anchored to the chest. This strapping mechanism is critical for preventing slippage, tilting, or drooping of the sensor, which could otherwise introduce errors in the sway detection algorithm. As this IMU module forms the core sensing component for the upper trunk posture assessment system, its correct positioning and firm attachment to the user's chest is essential to maintaining data fidelity. No separate cover was designed for this casing; instead, the MPU6050 module was soldered onto a custom-cut stripboard that fits snugly within the enclosure, serving as a makeshift lid. Wires are routed neatly along the side of the casing to connect with the main microcontroller unit housed separately, and the full assembly is detailed in Figure 3.12.

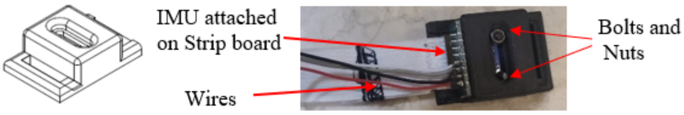


Figure 3.12: Chest IMU Holder and Strap

Further supporting modules include the development of a devboard holder casing, as illustrated in Figure 3.13. This housing is specifically designed to hold the ESP32 terminal expansion board and includes four M4 holes for secure mounting. A slotted track system is integrated into the design to accommodate a cover panel, transforming the unit into a compact box. This casing is then attached to a rigid backplate, shown in Figure 14, via similar slots and tracks, forming part of the thoracic module of the prototype. The entire unit, which also stores the SD card logging module, maintains a slim profile of only 33 mm, contributing to the overall wearable form factor. The backplate features multiple strap interface points, as shown in Figures 3.10 and 3.11, allowing stable and adjustable attachment to the user's upper back.

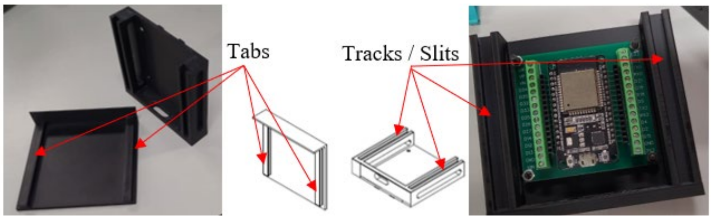


Figure 3.13: Devboard Holder with Tabs and Slots

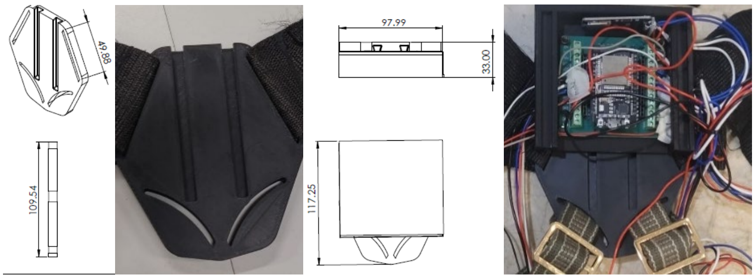


Figure 3.14: Attachment of Devboard on Backplate

To enhance user interaction and monitoring, a wrist-worn OLED module was developed, depicted in Figure 3.15, and assembled in Figure 3.16. The OLED circuit is encased snugly in a compact, watch-sized enclosure that can be secured around the wrist using neodymium magnets and elastic straps.

This module displays real-time visual feedback from the system, such as postural metrics or actuation states, allowing the user to monitor system performance immediately. The 17.5 mm height of the assembled module puts it slightly above the size of conventional wrist watches at 15mm but is still considered compact.

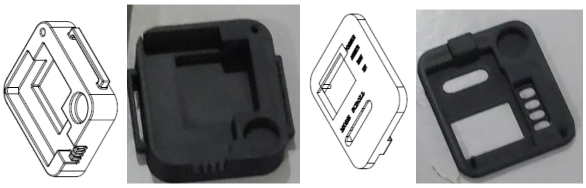


Figure 3.15: OLED Circuit Casing by Parts

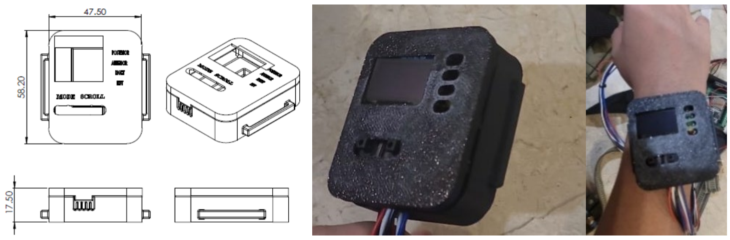


Figure 3.16: OLED Assembled Casing for Wrist

The waist-mounted component of the prototype, shown in Figure 3.17, contains the actuation mechanism. This unit integrates with adjustable waist straps to securely position the pneumatic actuators around the user's lower torso. The basic workflow of the actuation system is outlined in Figure 3.18, which presents the full block diagram of the postural adjustment mechanism. Together, these modular enclosures and attachment methods form an integrated wearable platform for real-time postural sway monitoring and correction. The design emphasizes mechanical simplicity, wearability, and data accuracy, enabling the system to function effectively in real-world usage scenarios.

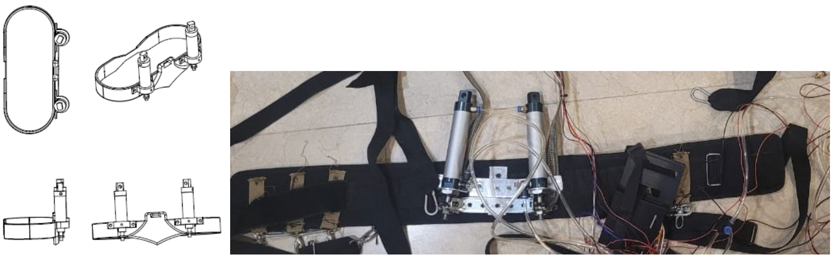


Figure 3.17: Waistband Assembly

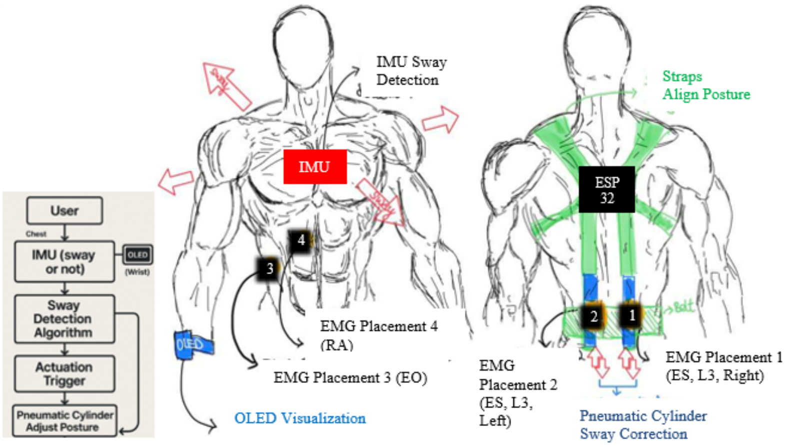


Figure 3.18: Block Diagram of Postural Adjustment of Prototype

The exoskeleton prototype comprises several essential components designed to deliver effective postural support while maintaining user comfort. The mechanical frame is ergonomically shaped and lightweight, allowing it to be worn on the back with minimal discomfort. It primarily uses fabric or polymer materials for the straps and padding, offering flexibility and comfort, while rigid sections, such as structural anchors, are made from aluminium and kept to a minimum to reduce weight and bulk. This frame provides the necessary structural base to secure the actuators and ensure stability during movement. The actuation system incorporates a pair of pneumatic cylinders, which generate linear pulling forces for posture adjustment. While the cylinders remain exposed, they are mounted in a way that avoids interfering with the natural movement. The overall design reduces the number of rigid components and simplifies the structure, helping to minimise the device’s form factor. As a result, the exoskeleton, though not entirely concealed, can still be worn discreetly under loose or layered clothing, balancing functionality with wearability.

3.7 Material and Component Selections

The development of the wearable posture-correcting exoskeleton necessitated careful consideration of mechanical, ergonomic, and biomechanical requirements to ensure optimal force delivery, user comfort, and device reliability. This section outlines the rationale behind the selection of key materials and components, as well as the biomechanical estimations that guided the actuator specifications.

3.7.1 Biomechanical Force Estimations

To determine the actuation requirements for postural correction of the upper trunk, torque estimations were derived based on basic biomechanical principles. The mass of the trunk segment accounts for approximately 41.6–50% of total body mass. Using the torque equation (Equation 3.13):

$$\tau = m \cdot g \cdot d \cdot sin(\theta) \tag{3.13}$$

where

τ is the torque required (Nm)

m is the mass of the trunk (kg)

g is the gravitational acceleration (9.81 m/s^2)

d is the distance from the pivot point (estimated as 0.265 m for H2, will be further explored in Section 3.9.4 and Figure 3.42)

θ represents the sway angle, where pitch angle corresponds to anteroposterior (AP) sway and roll angle corresponds to mediolateral (ML) sway. These angles are calculated using Equation 3.14, which is derived from Equations 3.3 and 3.4:

$$\theta_{pitch} = \arctan\left(\frac{DAP}{H_2}\right), \qquad \theta_{roll} = \arctan\left(\frac{DML}{H_2}\right) \tag{3.14}$$

Here, DAP and DML refer to the CoP displacements in the AP and ML directions, respectively, noted that all sway angle measurements here are in reference to fulcrum, thus using H2. According to literature, DAP values typically range from 3.67 mm to 17.66 mm, while DML values range from 5.22 mm to 24.44 mm (Ohlendorf et al., 2019; Goble & Baweja, 2018). These ranges are commonly used in postural control studies and are appropriate for both male and female participants aged 20 to 30 years, aligning with the demographic of the current study. Although CoP displacements are generally independent of subject height, some studies suggest a possible correlation with sex, where females often demonstrate slightly better balance performance. Nonetheless, the overall CoP-based balance scores remain comparable between sexes within this age group (Goble & Baweja, 2018; Becker et al., 2025). To simplify system design, the average, minimum, and maximum values of DAP and DML were adopted to compute the corresponding sway angles, which serve to define the correction limits. The resulting sway angle ranges are summarized in Table 3.3. Given that these sway angles fall within small ranges typical of healthy individuals, and to facilitate algorithm implementation, a simplified angular threshold was set. For AP sway, limits were extended to 5° posterior and 7° anterior, based on values reported in literature (Chaudhry et al., 2004). Additionally, a hard limit of ±15° was applied to both AP and ML sway to define the maximum expected range and ensure robustness in system performance.

Table 3.3: Computed Sway Angle Ranges Required for Correction

	$\boldsymbol{\theta_{pitch}}$ (°)	$\boldsymbol{\theta_{roll}}$ (°)
Min	0.7934	1.0680
Max	3.8126	7.3298
Mean	2.0531	2.8085
CI Range	1.7292 - 2.1611	2.5928 - 3.0241

For users weighing between 40 kg and 95 kg, the estimated required torque ranges from 17.78 Nm to 42.22 Nm. Within the average weight range of subjects recruited at UTAR (52–70 kg), torque requirements are between 23.11 Nm and 31.11 Nm. Since the exoskeleton uses two actuators, the torque per actuator is:

$$\tau_{per\ actuator} = \frac{\tau}{2} \tag{3.15}$$

To obtain the required linear force for actuation, torque is divided by the perpendicular distance to the backplate (r = 0.1 m):

$$F = \frac{\tau_{per\ actuator}}{r} \tag{3.16}$$

This results in required linear forces between 115.55 N and 155.55 N for average users, and up to 211.10 N for higher mass individuals. These values were used to inform the selection of pneumatic components.

Additionally, a pulley mechanism was established using linear pneumatic cylinders anchored at the fulcrum point (pelvis), routed over the shoulder (assume a joint or fixed guide), and anchored at the front waist using a carabiner. To determine the necessary stroke length for proper actuator function, a visualization was constructed, as shown in Figure 3.19, which includes a simplified stick diagram and directional vectors. Assuming the human body behaves as a rigid two-segment model with stationary lower limbs and no slack in the system, the required stroke length of the pneumatic cylinder can be estimated using the cosine rule, as expressed in Equation 3.17:

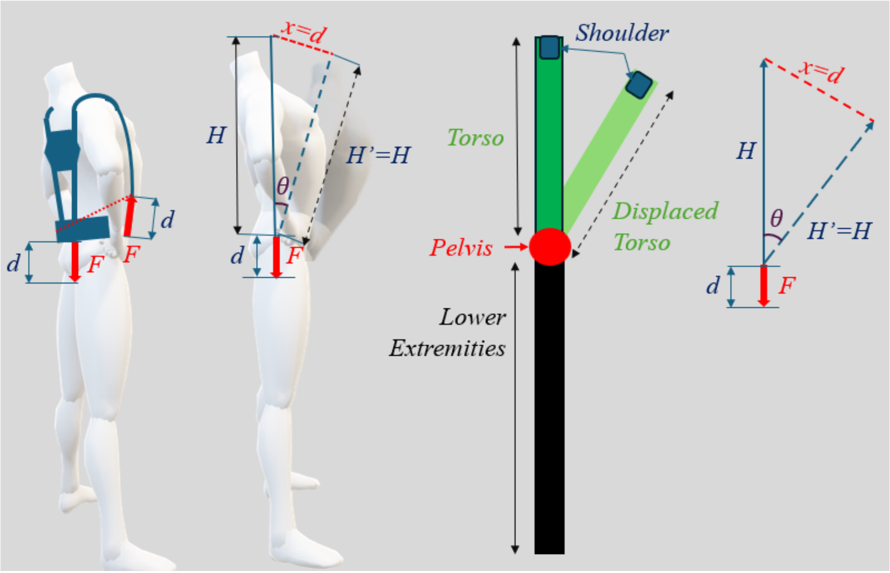


Figure 3.19: Schematic and Simplified Diagrams of Force and Stroke Length Required for Postural Sway

$$d^2 = H^2 + H'^2 - 2HH' \cos\theta = 2H^2(1 - cos\theta)$$
$$d = H\sqrt{2 - 2cos\theta} \tag{3.17}$$

where θ is the postural sway angle of ±15, H is the vertical distance from the chest to the waist (as defined in H2). Substituting the measured value of H, the required stroke length d is calculated to be over 69.18 mm, which defines the minimum actuator stroke necessary for the system to function effectively

3.7.2 Pneumatic Components

The pneumatic actuation was driven by the Mi Portable Electric Air Compressor 2 or MiPump 2 (Figure 3.20), a compact air pump weighing 490 g, of dimensions 123 mm×75.5 mm×45.8 mm, and powered by a 2000 mAh lithium-ion battery. The device operates within a pressure range of 0.2 to 10.3 bar and produces approximately 80 dB of noise. The maximum flow rate at 0 bar is 15.0 L/min; however, under typical operating loads of 2–6 bar, effective flow rates were estimated between 3 to 7 L/min. These estimates were supported by indirect reference to automobile tire inflation data, where inflating a 40 L tire in approximately 8 minutes implies a delivery rate of roughly 5 L/min.

Figure 3.20: Mi Portable Electric Air Compressor 2

The selected pneumatic cylinder (Figure 3.21) features a bore diameter of 32 mm and a stroke length of 75 mm, and up to 10 bars of pressure. In the operation of the prototype, pressures ranging from 2 to 6 bars were considered to not overburden the portable air compressor. To determine the force exerted under these conditions, the piston area should be calculated via Equation 3.18:

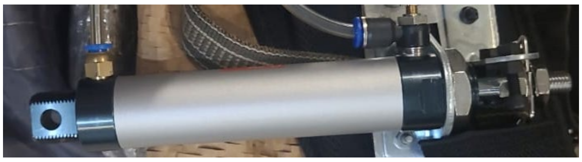


Figure 3.21: MAL Mini Aluminium Pneumatic Cylinder

$$A = \pi r^2 = \pi(0.016)^2 \approx 8.042 \times 10^{-4} m^2 \tag{3.18}$$

where

A is effective piston area

r is radius of piston (bore), which 0.5 bore size

From Equation 3.19, the force exerted can be determined:

$$F = P \times A \tag{3.19}$$

Thus, at 2, 4, and 6 bar operating pressures, the extension forces were estimated at approximately 160.84 N, 321.7 N, and 482.5 N, respectively, accommodating even extreme weighted individuals (>95kg), with safety factor of 2.3 at 6 bars and 1.5 at 4 bars. Considering the presence of a 10 mm diameter piston rod (rod radius = 0.005 m), the effective retraction area was slightly reduced, yielding retraction forces of about 145.2 N at 2 bars, from new effective retraction area from Equation 3.20:

$$A_{effective} = \pi r_{piston}^2 - \pi r_{rod}^2 \approx 7.26 \times 10^{-4} m^2 \tag{3.20}$$

Besides, the internal stroke volume per cylinder was calculated to estimate flow rates or stroke frequencies via Equation 3.21:

$$V = A \cdot Stroke = 8.042\ \times 10^{-4} \cdot 0.075 = 60.3mL \tag{3.21}$$

The air volume required per stroke was calculated to be approximately 60.3 ml. Given MiPump2's flow rates, each cylinder could operate at 0.967 Hz (2 bar), 0.691 Hz (4 bar), and 0.415 Hz (6 bar), respectively, under dual-cylinder conditions, which are deemed sufficient for corrective postural actuation during slow upper trunk sway.

3.7.3 Materials for Structural Components

Various grades of 3D-printed polymers were selected to meet the functional and structural requirements of different components in the prototype. PLA+ was employed for rigid and load-bearing parts such as the backplate module, Figure 3.14. This material was chosen for its enhanced mechanical strength and improved thermal resistance compared to standard PLA, making it suitable for securing straps for high tension. Flexible PLA was utilized in areas that required elastic deformation, including slotted regions for cable routing and snap-fit mechanisms, allowing repeated attachment and detachment without material failure such as the ESP32 holder and covers, Figure 3.13. Standard PLA was reserved for components subjected to minimal mechanical stress, such as the housing for the MPU 6050 (Figure 3.12) and OLED (Figure 3.16), where structural rigidity was not a primary concern. All components were fabricated using a cubic infill pattern at 25% density, providing an optimal balance between weight reduction, mechanical strength, and printing efficiency.

To improve user comfort, reducing risks of skin irritation or pressure-related discomfort, high-density foam wrapped in fabric was integrated into the shoulder and back regions of the exoskeleton. This padding effectively prevents direct contact between the user's body and the 50 mm wide nylon webbing straps, significantly enhancing wearability during prolonged use, illustrated in Figure 3.22. For structural support and secure integration of the pneumatic actuation system, the waist module was designed using a weightlifting-style belt constructed from ultra-high-density foam reinforced with durable nylon for strappings. This belt served as a load-bearing base for the attachment of pneumatic cylinders. To ensure mechanical reliability and resistance to high tensile loads, aluminium mounting plates and brackets were used to affix the cylinders to the belt via bolts and nuts. L brackets are also used as stoppers to control the activation of pneumatic cylinders, depicted in Figure 3.23. This configuration provided both structural integrity and user comfort, striking a balance between rigid support and ergonomic wearability.

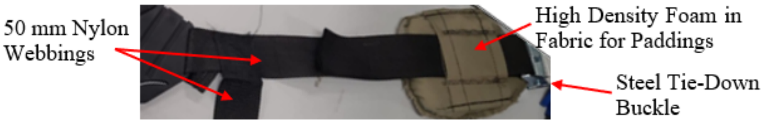


Figure 3.22: Fabric Attachments and Paddings of Prototype

To ensure both modularity and secure fastening in the wearable system, multiple D-rings were integrated into the waist belt using a combination of sewn nylon webbing and mechanical bolting. These D-rings functioned as anchor points for carabiner hooks, which were attached to the terminal ends of the shoulder straps, Figure 3.23. This design facilitated easy donning and doffing while allowing the user to fine-tune the strap tension between the upper and lower harness segments. Nylon webbing straps of 20 mm and 50 mm widths were selected for their high tensile strength, flexibility, and resistance to abrasion, making them suitable for both static load-bearing and dynamic adjustment purposes. Steel components, including adjustable ladder buckles, slide buckles, and tie-down buckles, were used in place of plastic alternatives to avoid mechanical failure under repeated high-tension loading. These buckles enabled user-specific customisation of strap lengths and overall harness fit, contributing to the adaptability of the prototype across users with varying body types. Standard bolts and nuts were employed throughout the assembly to ensure structural integrity and reusability. Collectively, these fastening and modular design elements support a robust, customizable, and user-friendly wearable system that can accommodate the demands of real-time upper trunk postural sway correction while maintaining a minimal form factor suitable for daily wear.

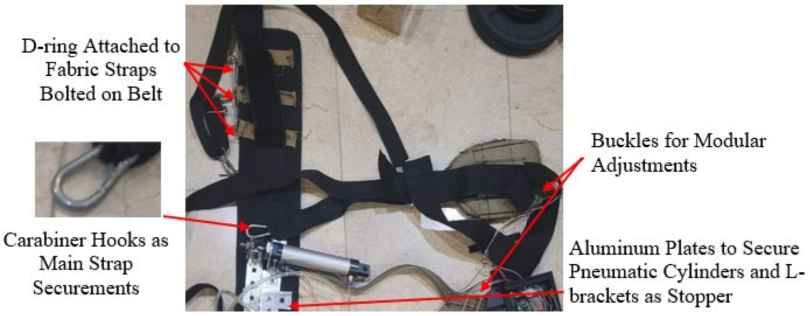
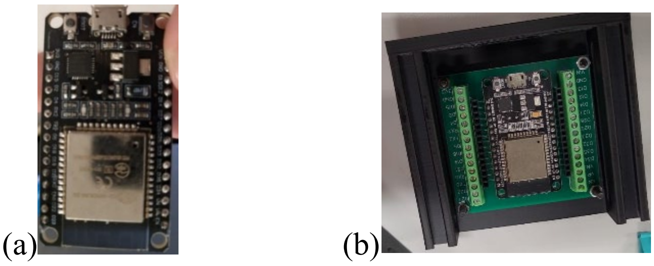

D-ring Attached to Fabric Straps Bolted on Belt
Buckles for Modular Adjustments
Carabiner Hooks as Main Strap Securements
Aluminum Plates to Secure Pneumatic Cylinders and L-brackets as Stopper

Figure 3.23: Fastening and Securement Strategies of the Prototype

3.7.4 Circuit Components

The electronic system was built around the ESP32 microcontroller, selected for its superior performance compared to standard Arduino boards. The ESP32 offers integrated Wi-Fi (2.4 – 2.5 GHz) and Bluetooth connectivity, on-board clock (40 MHz crystal), higher processing speed, and greater memory capacity, all while maintaining a low cost. The ESP32 Devkit V1 (30 Pins) with ESP32-WROOM-32 chip, illustrated in Figure 3.24 (a), was selected for its dual-core processing and ample memory, including 520 kB SRAM and 4 MB external flash (Espressif Systems, no date). This allows efficient real-time processing of chest sway data for postural sway detection, wireless communication for real time monitoring via GUI, and direct code execution from flash. Its RTC memory supports low-power modes, making it suitable for wearable applications. These features make it essential for real-time data acquisition and wireless communication in wearable biomedical applications. To streamline circuit assembly and improve reliability, an ESP32 terminal board was employed, Figure 3.24 (b). This accessory allows for solderless connections, simplifying prototyping, reducing troubleshooting time, and providing more secure and stable electrical connections compared to conventional Dupont jumper wires, which are prone to loosening and signal inconsistency under motion.



(a) (b)

Figure 3.24: (a) ESP32 Devkit V1, 30-Pin Model, (b) with Terminal Block

For trunk or chest sway detection, an MPU6050 was selected, shown in Figure 3.25. This low-cost (approx. RM10) 6-degree-of-freedom sensor integrates both a 3-axis accelerometer and a 3-axis gyroscope, enabling accurate real-time monitoring of angular velocity and acceleration which can be used to derive upper trunk sway. Its compact size and compatibility with the ESP32 make it ideal for wearable implementations.

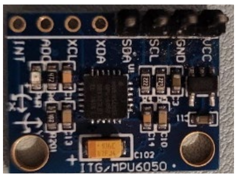


Figure 3.25: MPU 6050

To control pneumatic actuation, a 4-channel relay module, shown in Figure 3.26, was used to interface between the microcontroller and solenoid valves, detailed in Chapter 3.8.3 below. The relays enable the ESP32 to switch the high-current loads required by the solenoids, ensuring safe and reliable actuation of the pneumatic cylinders in response to detected postural deviations.

Figure 3.26: A 4-Channel Relay

A compact 0.96" SSD 1306 OLED display, as shown in Figure 3.27, was integrated into the system to deliver real-time visual feedback to the user. It displays key information such as stability status, sensor readings (in numerical or graphical form), and current orientation, which improves usability during testing and operation. The 0.96" size was chosen to fit a compact, watch-like form factor, while the OLED technology offers high contrast and clarity, enabling easy readability even with small fonts.

Figure 3.27: 0.96" SSD 1306 OLED

Power for the solenoid valves was supplied by four 3.7V lithium-ion AA batteries housed in a battery holder, selected for their high energy density, rechargeability, and compact size. To deliver the required 12V for the solenoid valves, an LM2596 buck converter, in Figure 3.28, was used to regulate the output voltage efficiently and prevent overvoltage damage. Meanwhile, the ESP32 and its peripheral circuits were powered separately via a power bank connected directly to the ESP32. For data logging purposes, an SD card module was included in the circuit. This allowed continuous storage of sensor data, which is crucial for post-processing, performance evaluation, and further refinement of sway classification algorithms. The use of onboard data storage ensured that the system could function independently in real-world settings without requiring constant connectivity.

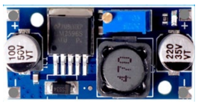


Figure 3.28: LM2596 Buck Converter

3.7.5 Calibration of Inertial Measurement Unit

In an ideal scenario, when the IMU is placed on a flat surface, the x and y-axis values should register as 0, while the z-axis should reflect the gravitational force, approximately 9.81 m/s². However, slight deviations often occur due to factors such as sensor imperfections or environmental conditions. To address this, calibration of the IMU is necessary before taking accurate measurements. The calibration process involves averaging the IMU readings over a set period and adjusting them to match the expected values for each axis. This step ensures that the sensor readings align more closely with true physical forces. The calibration can be efficiently performed using the MPU6050_light library, which automates the process and ensures accurate sensor performance. The flowchart depicting the IMU calibration process is shown in Figure 3.29.

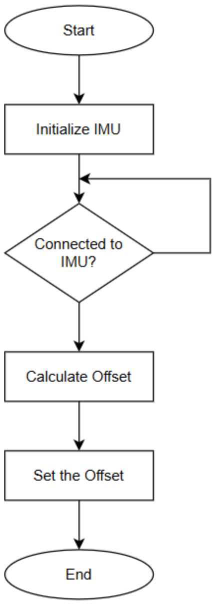


Figure 3.29: Flowchart Showing the Process in Calibrating an IMU.

3.8 Circuit Development

The circuit development phase was fundamental to the integration of sensor data acquisition, real-time processing, actuation control, and data logging within the wearable exoskeleton system. A compact and modular electronic system was designed to ensure compatibility with the mechanical structure while minimizing bulk and preserving user mobility. The objective was to create a robust embedded control system capable of capturing postural sway data in real time, making classification decisions, and actuating pneumatic components responsively to support posture correction. The development involved selecting appropriate microcontrollers, sensors, actuators, display units, and power management components. Particular attention was given to the balance between performance, reliability, and cost, as the system had to operate continuously and accurately in a dynamic, wearable environment. Key considerations included real-time responsiveness, wireless communication capabilities, ease of integration with the mechanical harness, and the ability to log data for further analysis. All components were selected and integrated with the goal of ensuring safe and efficient operation of the exoskeleton during testing and use. The circuit was built on a terminal board to enable clean, modular wiring while allowing future adjustments or upgrades.

3.8.1 MPU 6050 Circuit Design for Chest Sway Detection

The MPU6050 inertial measurement unit (IMU) was employed to capture real-time motion data for chest sway detection. The sensor was connected to the ESP32 via a terminal board, with soldered wire connections through a strip board to ensure mechanical stability and noise reduction. The pin configuration was such VCC to 3v3, GND to GND, SCL to GPIO22, and SDA to GPIO21 on the ESP32, evident in Figure 3.30. This configuration is shown integrated within the custom ESP32 enclosure in Figure 3.12. To interface with the sensor, the MPU6050_light.h library was utilized due to its lightweight design and performance in embedded systems which provided 6-axis raw data, including linear acceleration (ax, ay, az) and angular velocity (gx, gy, gz). From these values, basic orientation parameters such as pitch and roll were computed, along with sway displacement values in the anteroposterior (DAP) and mediolateral (DML) directions, as detailed in Section 3.5.1. These real-time raw and computed values formed the input for the sway detection algorithm, which further processed them into higher-level sway parameters to classify and correct postural instability. All sensor and algorithmic operations were embedded within the ESP32 system for efficient onboard processing. To enable accurate timestamping of real-time sensor data, the ESP32's onboard real-time clock was synchronized using Network Time Protocol (NTP). The time.h library was included to support timekeeping functions. Two NTP servers, pool.ntp.org and time.nist.gov, were used for redundancy, while the time zone was set to UTC+8 with no daylight offset. This synchronization ensures precise timing for IMU-based sway detection and allows reliable time-based logging and analysis of movement data, which is critical for applications involving temporal postural sway classification.

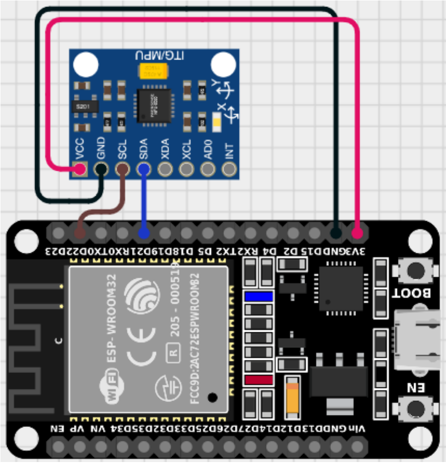


Figure 3.30: Circuit Connection of Chest IMU to ESP 32

3.8.2 SD Module Circuit Design for Data Logging

To facilitate reliable real-time data logging, an SD card module was integrated into the ESP32 system using the Serial Peripheral Interface (SPI) protocol. The SdFat.h library, along with SPI.h, was selected for its optimized performance, extended compatibility with large-capacity SD cards, and efficient file handling—especially suitable for embedded systems with limited memory. Compared to the standard SD.h library, SdFat.h offers faster access and greater control over the file system, making it ideal for time-sensitive applications. The SD card was used to store IMU raw data and derived postural sway parameters in .csv format. This allows for structured, timestamped offline analysis, repeatability in testing, and validation of the detection algorithm. Connections were made to the ESP32's default SPI pins, in Figure 3.31, which are MISO (D19), MOSI (D23), SCK (D18), and CS (D5). The module was securely housed within the main ESP32 enclosure to maintain a compact and integrated form factor. A global logging system was implemented using variables such as loggingEnabled, csvFileName, and lastLogTime, with a defined logging interval of 500 milliseconds to regulate data sampling. When enabled, the system logs IMU data and derived sway parameters into a .csv file, providing structured and time-synchronized datasets essential for offline analysis, performance evaluation, and algorithm validation.

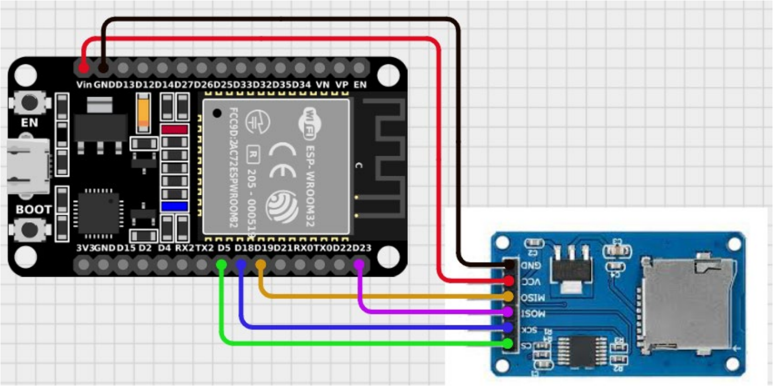

Vin GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
BOOT
ESP-WROOM32
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
GND
VCC
MISO
MOSI
SCK
CS

Figure 3.31: Circuit Connection of SD Card Module to ESP 32

3.8.3 Relay and Solenoid Valve Circuit Design for Actuation

To enable automatic actuation of the pneumatic cylinders, a relay-based switching system was developed using a 4-channel relay module connected to two solenoid valves as shown in Figure 3.32. These valves regulate airflow to the cylinders, enabling directional stabilization based on real-time posture feedback. Power to the solenoid valves was supplied by four 3.7V Li-ion batteries housed in a battery holder, chosen for their high energy density and reusability. These were configured in series and fed into a buck converter (LM2596), which stepped the voltage down to a stable 12V supply required by the solenoid valves. The COM and NC (Normally Closed) terminals of the relay channels were connected to the solenoid valves to ensure they remain inactive by default and are only triggered when the relays are activated. To ensure proper electrical flow and safe actuation, the positive output (+) from the buck converter was connected to the COM (Common) terminal of the relay channels. The Normally Closed (NC) terminals of the relays were then connected to the positive terminals of the solenoid valves. Meanwhile, the negative output (–) from the buck converter was connected directly to the negative terminal of the solenoid valves.

This configuration means that the valves remain off (circuit open) when the relays are inactive, and are only powered when the corresponding relay is triggered (sets COM to NC path as closed). By controlling the HIGH/LOW state of the GPIO pins on the ESP32, the system selectively activates solenoids for specific directional control based on postural instability. This wiring not only prevents unnecessary energy drain but also adds a layer of safety by defaulting to the "off" state unless an explicit signal is sent by the microcontroller. The relays were controlled via GPIO pins D27 and D33 of the ESP32 (defined as RELAY3 and RELAY4 respectively). These were initialized as output pins and set to LOW (inactive) during startup. The relay control logic was implemented in the updateRelays() function, which activated specific relay channels based on the user’s current orientation status (LEFT, RIGHT, ANTERIOR, POSTERIOR, INSTABILITY, or STABLE), as detected through the MPU6050. This configuration enabled targeted stroke of the pneumatic cylinders to assist with balance correction dynamically. The relay module and wiring were secured onto the waistband via bolting to maintain a compact, wearable system.

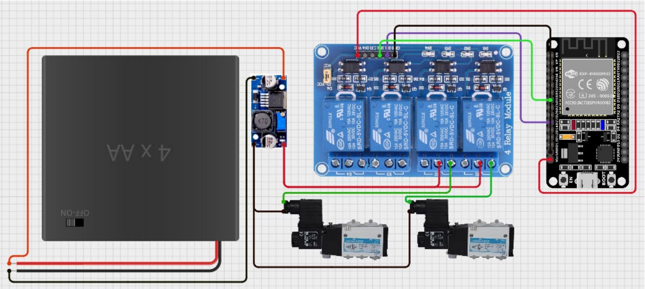


Figure 3.32: Relay and Pneumatic Component Circuit Connections

3.8.4 OLED Display Circuit Design for Quick Visuals

A 0.96” SSD1306 OLED display was integrated into the system to provide clear, high-contrast visual feedback to the user. This includes stability status, real-time IMU readings, and system modes, enhancing both usability and monitoring during operation. The OLED module was controlled using the I2C protocol, connected to the ESP32’s SCL (D22) and SDA (D21) pins, sharing the same I2C bus as the MPU6050 IMU for efficient pin usage, connection in Figure 3.33. To support interface navigation, two tactile pushbuttons were added and connected to digital pins D14 and D4. These buttons enable user interaction with the display, allowing switching between numerical readouts, graphical plots, and system menus. The input logic was debounced in software to ensure reliable operation. For the software interface, the Adafruit_GFX.h and Adafruit_SSD1306.h libraries were selected due to their reliability, extensive documentation, and built-in support for drawing graphics and handling fonts. They also support memory-constrained microcontrollers like the ESP32 while maintaining good performance and responsiveness. The display and buttons were compactly housed within the main ESP32 casing, maintaining a small form factor while providing intuitive, user-friendly interaction for testing and monitoring.

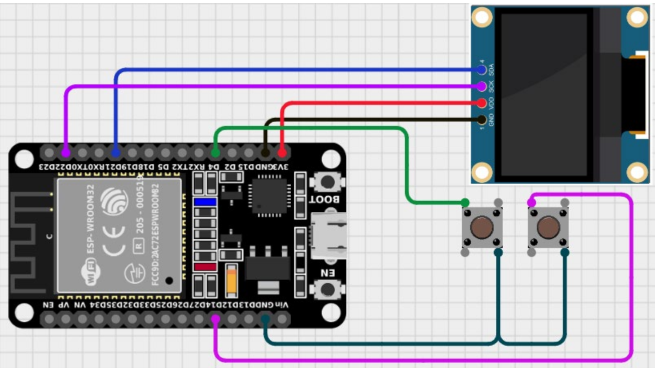


Figure 3.33: OLED Display Circuit Connection

3.8.5 Circuit Integration

The complete circuit integration was carefully structured to ensure reliable communication, efficient power management, and modularity within a compact embedded system as depicted in Figure 3.34. The ESP32-WROOM-32 microcontroller served as the central hub, interfacing with multiple peripheral components through both I2C and SPI communication protocols. The MPU6050 IMU and the SSD1306 OLED display shared the same I2C bus, connected to the ESP32’s default I2C pins (GPIO 21 for SDA and GPIO 22 for SCL). Potential I2C conflicts were mitigated through device-level address management, as the MPU6050 and OLED used unique default addresses (0x68 and 0x3C respectively), ensuring seamless simultaneous communication without interference. These connections were routed through a terminal board and soldered to ensure low-resistance, noise-resistant signal paths.

In parallel, the SD card module operated independently on the SPI protocol, utilizing GPIOs 23 (MOSI), 19 (MISO), 18 (SCK), and 5 (CS), which prevented any cross-talk with the I2C bus. The SdFat library was selected over the standard SD library due to its enhanced compatibility with the ESP32 architecture and its support for non-blocking, efficient file operations necessary for real-time data logging. Digital I/O pins were allocated for additional modules such as the 4-channel relay (GPIOs 27 and 33), and user interface buttons (GPIOs 14 and 4). The entire assembly, including the ESP32, IMU, SD module, and wiring, was securely housed within a custom enclosure to maintain mechanical stability and ensure reliable operation during motion and testing.

Power distribution was managed via a dual-source strategy, namely, a 5V power bank supplied the ESP32 and its peripherals, while a regulated 12V output from a buck converter powered the solenoid valves via relay switching. Ground lines across the power and logic circuits were commonly tied to maintain a consistent reference voltage, minimizing the risk of floating grounds or erratic behavior. This integrated configuration enabled concurrent real-time data acquisition, control, and feedback operations, forming the functional backbone of the wearable exoskeleton system.

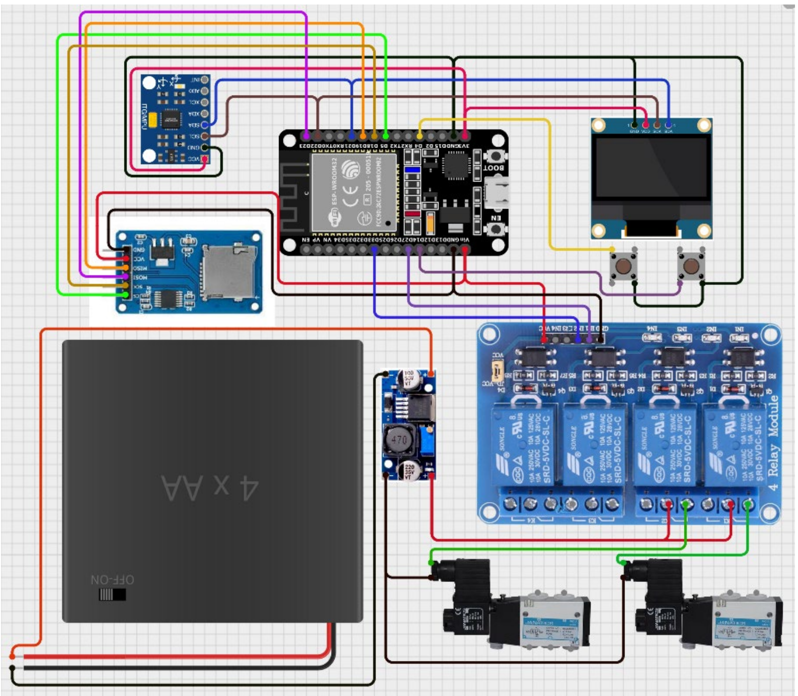


Figure 3.34: Integrated Circuit Design

3.8.6 Graphical User Interface (GUI) Development

To enhance the functionality and use interactions of the back-support strap-based pneumatic exosuit for static posture correction, an IoT-enabled software framework was developed. This system allows for real-time visualisation, remote monitoring, and session logging of biomechanical data obtained during use. The overall architecture adopts a modular approach, comprising four major tiers: the frontend interface, backend logic, database layer, and hardware integration unit, depicted in Figure 3.35 as block diagram.

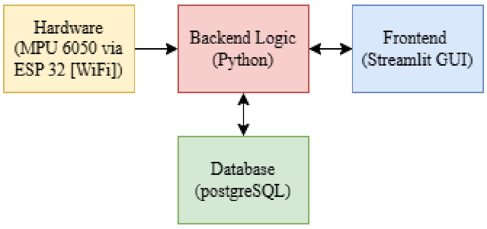


Figure 3.35: Block Diagram of System Architecture

The frontend is constructed using Streamlit, an open-source Python library, to facilitate rapid development of interactive dashboards and interfaces. This environment supports both patient and administrator interactions through a role-based access control system. The backend is implemented in Python 3.11 and is responsible for handling logic control, session management, and external communication with microcontroller hardware. PostgreSQL, managed and queried using pgAdmin 4, serves as the relational database system, providing high-performance, ACID-compliant data storage for user credentials, metadata, session logs, and raw sensor data. Data flow is bi-directional between the hardware and software layers, with real-time movement data streamed from the ESP 32-equiped chest IMU via Wi-Fi to the backend for parsing, analysis, and visualization. Development was carried out in Visual Studio Code, offering integration between Python modules, PostgreSQL, and frontend components.

The frontend design utilizes Streamlit due to its capability to transform Python scripts into shareable web apps without requiring extensive HTML or JavaScript knowledge. Python serves as the primary backend language, enabling efficient data processing and communication with the microcontroller. The PostgreSQL database is selected for its ACID-compliance, scalability, and support for complex query operations. Libraries such as psycopg2 are used to handle database connections, while pandas manages data manipulation. Visualization tasks rely on matplotlib and seaborn, which provide versatile options for creating both static and interactive plots. The ESP32 platform is used for its lightweight and portable configuration, transmitting IMU data via Wi-Fi through socket communication. Table 3.4 justifies the methods selected.

Table 3.4: Technology Justification Summary

Component	Technology	Justification
Frontend	Streamlit	Simplifies deployment of interactive GUI dashboards using only Python.
Backend	Python 3.11	Enables robust logic control, data processing, and integration flexibility.
Database	PostgreSQL	Scalable, ACID-compliant SQL engine with robust indexing and query support.
DB Connector	psycopg2	Secure and efficient PostgreSQL adapter with parameterized query support.
Data Handling	pandas	High-performance data transformation and time-series handling.
Visualization	matplotlib, seaborn	Facilitates advanced data plotting and real-time visual feedback.
IMU Interface	ESP 32 (Wi-Fi)	Lightweight embedded solution for real-time data acquisition and streaming.

User authentication is handled through a dedicated Python script that validates login credentials against stored PostgreSQL records. At this stage, backend queries the users table using parameterized queries to avoid SQL injection. Upon successful verification, session states are updated to reflect the user's role. Patients and administrators use the same login page, but content rendering is conditional, restricting patients to their own data while granting admins full access to all user records and system configurations. Passwords are hashed using Argon2, and session inactivity triggers an auto-logout protocol for enhanced security. Once authenticated, patients are directed to a dashboard that aggregates session counts, live trial metrics, and recorded data via pandas. The dashboard is designed to refresh every five seconds, displaying real-time values streamed from the IMU sensor. Users can initiate new trials, which are stored in structured formats (JSON or CSV) and uploaded to the SQL database. Historical data retrieval is enabled through parameterised SQL queries to support filtering and analysis. The navigation sidebar offers clear access to trial initiation, historical records, and secure logout features. On the other side, administrators access a more advanced dashboard that includes real-time visualizations of login activity, session metrics, and patient comments. Navigation features enable administrators to view user logs, manage patient accounts, reset passwords, and moderate feedback. The backend enforces route protection, session validation, and detailed auditing of administrative actions. Filtered views allow for rapid inspection of patient records, while layered navigation supports drill-down access to individual trials and raw sensor data. Figure 3.36 shows the flowchart of main code and both dashboards.

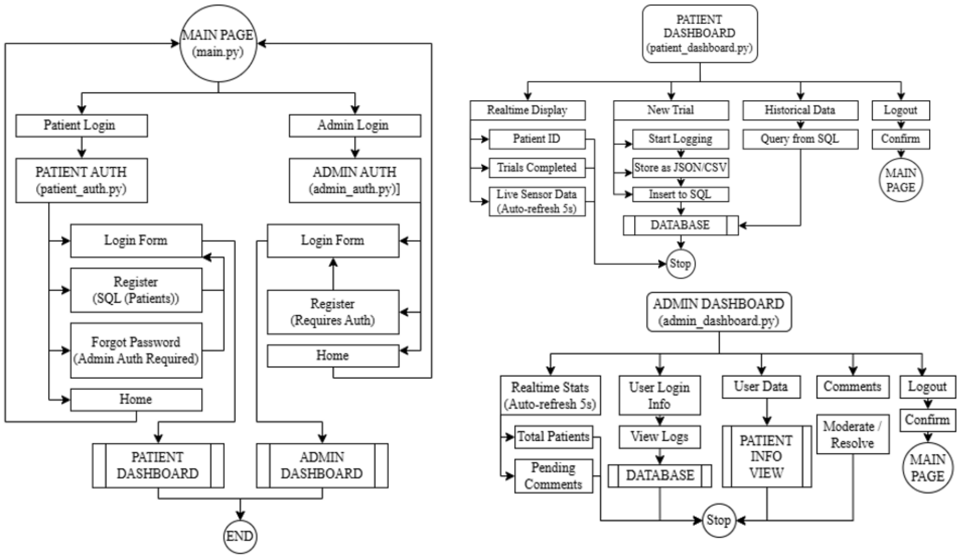


Figure 3.36: GUI Flowchart for Main Page and User Dashboards

The data visualisation module, depicted in flowchart in Figure 3.37, generates time-series plots and interactive charts, with matplotlib and seaborn, representing metrics such as DAP, DML, accelerometer data and other computed CoP parameters. These visualizations help users interpret the effectiveness of the exosuit under in real time and make comparisons with historical data. Patient records can be searched and filtered by timestamp, trial ID, or username. Tables are rendered with pagination and expansion features for clarity and convenience, implementing parameterised SQL queries to ensure injection safety. Linked views allow users to explore specific datasets in more detail, including gauge charts and raw data logs.

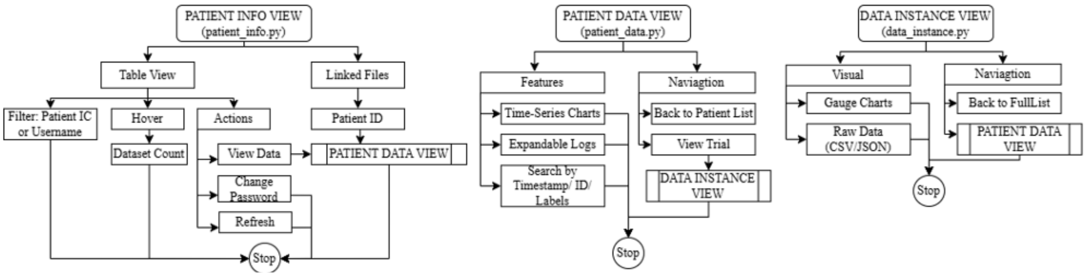


Figure 3.37: Data Visualization Flowchart

The dual-user architecture accommodates patients and administrators through modular, role-specific flows. Streamlit serves as the unifying framework for frontend interaction, while PostgreSQL manages data persistence. Each user interaction is captured in a session, with routing logic ensuring restricted access to sensitive modules. Confirm prompts and session-based protection minimize the risk of accidental data exposure. A full-stack security model includes Argon2 hashing, audit trails, and automatic logout after 10 minutes of inactivity as depicted in the diagram in Figure 3.38.

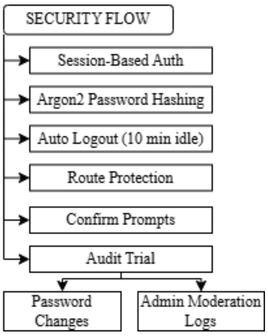


Figure 3.38: Security Features of GUI

The PostgreSQL database depicted in Figure 3.39 uses connection pooling to optimize performance and ensure consistent access under concurrent loads. It consists of schemas for users, administrators, trial data, and comments. All records are transactionally handled to avoid partial writes or corrupt entries. Backup protocols and error logging are built into the system to ensure data integrity and support post-hoc analysis in the event of a system failure. The ESP 32 and MPU605 circuit was configured to stream accelerometric and gyroscopic data over a local Wi-Fi network. The microcontroller transmits data in real-time to the Python backend, which parses and logs it into the SQL database. These values are simultaneously rendered on the frontend, allowing clinicians to track postural sway, trial behaviour, and subject activity. Security concerns related to embedded-to-server communication were addressed by obfuscating IP addresses and limiting communication to local network scopes during development. Hardcoded credentials, tokens, and API keys were excluded from the repository to reduce the attack surface. Future iterations may include mutual

TLS authentication or encrypted payload transmission using AES to further strengthen communication channels.

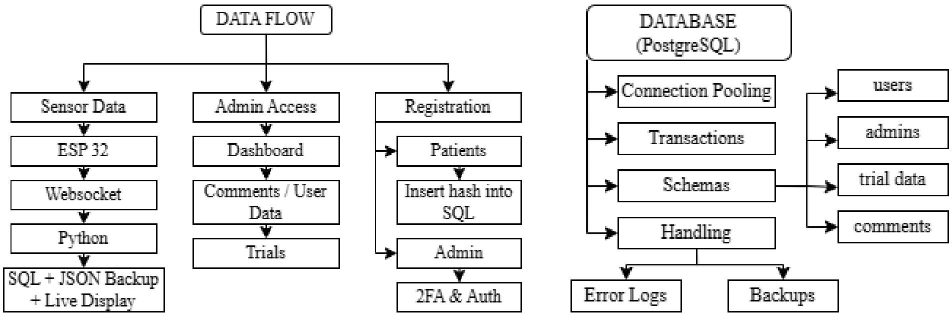


Figure 3.39: Data Flow and Database Flowchart

In summary, the IoT-enabled monitoring system for the pneumatic exosuit combines a responsive user interface, robust backend processing, and secure data infrastructure to facilitate comprehensive posture tracking and remote assessment. The use of open-source tools and modular programming practices ensures scalability, maintainability, and user-centric interaction, thus elevating the utility of the exosuit within clinical and rehabilitative contexts.

3.8.7 Assembled Prototype

The developed prototype provides active back support using pneumatic actuation triggered by real-time postural sway. It consists of a soft wearable frame, chest-mounted IMU, ESP32-based control unit, pneumatic actuators, a custom sway detection algorithm, and user interfaces including an OLED display and GUI, illustrated in Figure 3.40. Constructed mechanically from fabric straps and buckles, the system is compact and lightweight. Pneumatic components are mounted at the waist, with actuators fixed via aluminium plates, while lighter parts of main circuit on thoracic spine as well as IMU strapped on chest. Donning is like wearing a backpack, with added leg straps for stability. The system supports users up to 95 kg (safety factor 1.3), operating at 4–6 bar via a portable miPump2 compressor. Audible hissing and motor noise are present during activation. On startup, the IMU is calibrated on a flat surface; once donned, real-time sway is monitored, triggering actuation as needed. The OLED displays system status and allows mode selection, while data can be logged via SD card or GUI. Current circuitry is functional but may be miniaturized through future IC integration.

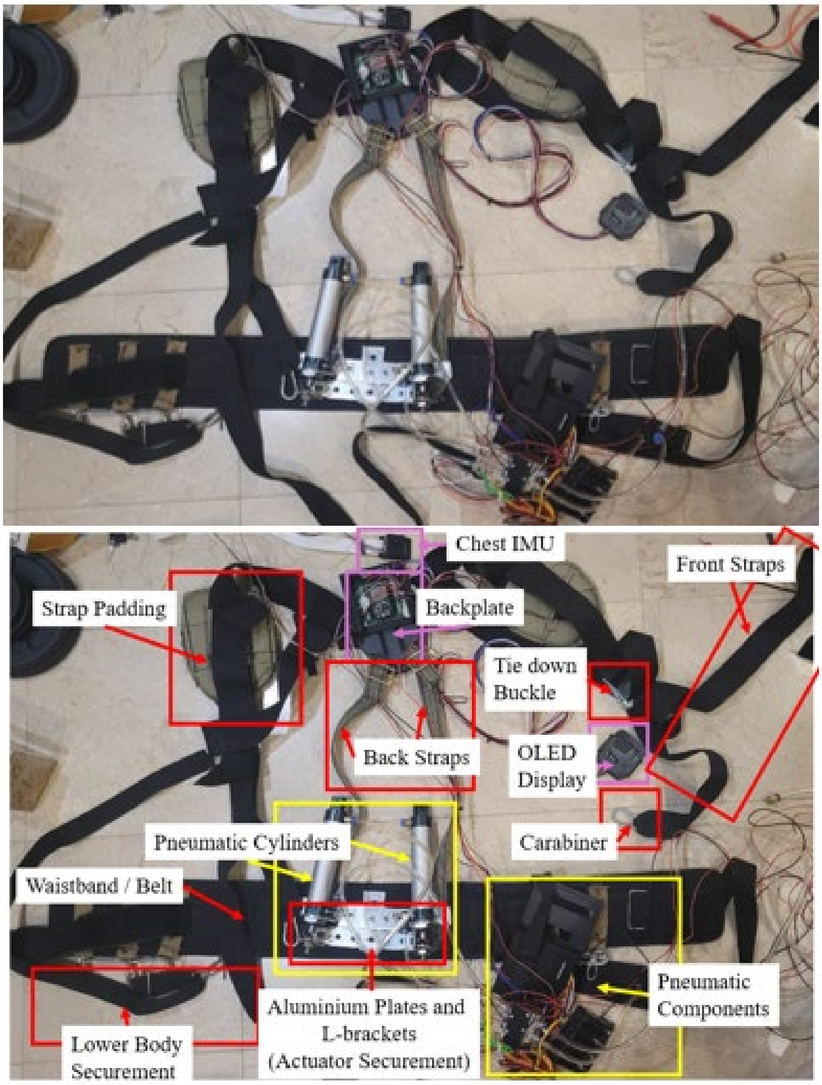


Figure 3.40: Final Assembled Prototype of Back-Supported Strap-based Pneumatic Exoskeleton for Standing Postural Sway Correction

3.9 Algorithm Development

This section presents the methodological development of a postural sway classification algorithm, designed to detect distinct balance conditions in users equipped with the proposed exosuit system. The classified output was intended to inform the actuator control logic for real-time support and intervention. The algorithm was trained and evaluated using real-world data collected from 37 participants under systematically varied postural perturbation conditions. Each sample was labelled according to experimental ground truth, allowing for supervised classification into four predefined sway categories: stable, anteroposterior sway, mediolateral sway, and unstable. These classifications aimed to support balance rehabilitation or assistance protocols. The algorithm design proceeded through a structured pipeline: initial data preprocessing, feature extraction, exploratory clustering, supervised classification model evaluation, and finally the development of an empirical rule-based threshold algorithm optimized for microcontroller deployment, shown in Figure 3.41. This development drew significant conceptual influence from the standing sway detection framework proposed by Ando et al. (2023), which implemented IMU-based kinematic classification on humanoid platforms. However, the current work extended their approach into real-world human applications and adapted it for computationally efficient embedded deployment.

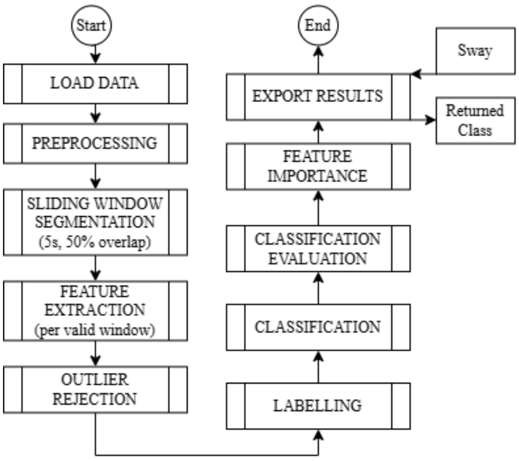


Figure 3.41: Pipeline of Algorithm Development (.mat) Code

3.9.1 Algorithm Design and Pipeline

This study developed an automated framework for postural sway classification and feature analysis using time-series accelerometry data. The methodology integrates signal preprocessing, multi-level outlier detection, domain-specific feature extraction, sway condition labelling, and both threshold-based and machine learning classification. The entire analytical pipeline was implemented in MATLAB R2023b, consisting of approximately 2500 lines of modular code, appended in Appendix D. The framework was designed to accommodate real-world sensor noise, inter-subject variability, and embedded hardware constraints, with reproducibility and interpretability as central principles.

The classification pipeline was adapted from the wearable AI-based postural sway detection system proposed by Ando et al. (2023), which achieved over 98% classification accuracy using accelerometer and gyroscope data processed on an embedded AI microcontroller. Their architecture incorporated sensor fusion, statistical and frequency-domain feature extraction, overlapping window segmentation, ensemble-based machine learning (Random Forests as primary, k-NN as fallback), and real-time embedded inference. In the current study, several structural elements of Ando’s framework were retained. These include the core feature extraction logic encompassing RMS sway, directional displacements, and velocity-based features, as well as the classifier suite for benchmarking purposes. Cross-validation and class-specific threshold tuning were also preserved. However, due to hardware limitations, specifically the use of a general-purpose ESP32 microcontroller in place of Ando’s dedicated STMicroelectronics AI core, the ensemble learning model was replaced with a rule-based threshold classifier. This substitution emulated the decision boundaries of the original model through interpretable, empirically derived thresholds, enabling real-time execution on resource-constrained devices.

While the foundational logic is inherited from Ando et al. (2023), key methodological differences distinguish the present study. Notably, their experiments relied on synthetic sway patterns generated via a robotic platform, minimizing data variability and noise. In contrast, this study employed real human subjects (n = 37), introducing natural fluctuations due to physiology, movement inconsistency, and sensor misalignment. Next, their hardware facilitated on-device machine learning inference, whereas our ESP32-based system imposed stricter computational, memory, and energy constraints. Also, the real-world signals collected here were subject to various noise sources, including muscle tremors, fatigue-induced instability, skin motion artifacts, and sensor drift, all of which necessitated enhanced preprocessing and outlier detection. These distinctions explain the lower classification performance compared to prior results and support the methodological pivot to a threshold-based model optimized for deployment in exoskeletal assistive systems.

Recent literature further contextualizes this approach. Machine learning models such as k-NN, SVM, and Random Forests have been shown to achieve high classification accuracy (typically above 90%) in fall or sway detection tasks when applied to well-curated datasets with minimal inter-subject variability (Turan & Barshan, 2021). However, performance tends to degrade, often below 70%, in clinical or real-world settings where data heterogeneity is pronounced (Gattinara et al., 2022). As such, rule-based systems remain a viable alternative, offering transparency, computational efficiency, and deterministic behavior suited to embedded control, particularly in healthcare or rehabilitation contexts (Ando et al., 2023).

3.9.1.1 Data Acquisition

Raw sensor data were filtered to isolate low-frequency components associated with postural sway. A zero-phase second-order Butterworth filter (0.01–0.6 Hz) was applied to attenuate high-frequency noise while preserving physiological sway dynamics, consistent with prior findings in balance assessment (Ando et al., 2023). Outlier detection and correction were performed using robust statistical thresholds and visual inspection of signal traces. In our case, raw accelerometery data were collected from a wearable inertial sensor mounted at the subject's chest. Each trial was recorded under specific sway conditions detailed in Section 3.11. Data were stored as individual .csv files, systematically organised by subject ID and condition. Anthropometric parameters, including body height, chest-to-ankle height (H1), and chest-to-waist height (H2), were extracted from an Excel sheet and matched to each subject via both exact and fuzzy name logic. Where missing values were encountered, gender-specific standard estimates were imputed based on trends illustrated in Figure 3.42 and supplementary Graphs A-2 and A-3.

3.9.1.2 Preprocessing and Filtering

All raw accelerometer signals were resampled to a uniform frequency of approximately 29.4 Hz ($\Delta t = 0.034$ s). A second-order zero-phase Butterworth bandpass filter with cut-off frequencies set at 0.01 Hz and 0.60 Hz was applied to each axis. This range was selected to preserve the low-frequency sway components while attenuating motion artefacts, sensor drift, and physiological tremors. The filtering was implemented using a forward–reverse filtfilt function to eliminate phase distortion, with validation conducted through frequency-domain response analysis. The Butterworth design's maximally flat passband characteristics make it suitable for biomechanical signal processing, particularly for sway motion isolation. Graph A-1 shows a sample processed data,

3.9.1.3 Multi-Level Outlier Detection

A hierarchical outlier detection scheme was employed to improve data integrity at multiple levels. First, at the signal level, spikes were removed using z-score filtering ($|z| > 3.5$), wavelet decomposition with a Daubechies-4 ('db4') basis to detect transient anomalies, and a Median Absolute Deviation (MAD) approach with a 0.5-second sliding window. Outliers were replaced with NaN and subsequently interpolated. Second, at the window level, a 5-second sliding window (with 50% overlap) was applied across the filtered signals. Windows with more than 30% missing or outlier-filled samples were discarded to maintain data reliability. Third, at the feature and subject levels, extracted features were screened using z-scores. Individual windows exceeding a z-score of ±3 in any feature were flagged, while subjects with mean feature vectors surpassing ±2 were treated as outliers. Linear interpolation was used to impute missing values prior to final feature extraction, minimizing potential bias. This multi-tiered approach was essential for mitigating the effects of short-duration disturbances caused by abrupt motion, device tension, or pneumatic actuation, which can disproportionately affect downstream classification performance.

3.9.1.4 Feature Extraction

Within each valid window, DAP and DML were computed as described in Equations 3.3 and 3.4, respectively. These were used to extract a comprehensive set of biomechanically relevant sway features. Statistical features included root mean square sway (D_RMS) and displacement range (DR). Geometrical features comprised the 95% predicted ellipse area (PEA_95) and rectangular sway area (Rs). Kinematic descriptors included total sway path length, mean and maximum sway velocities, and Euclidean decomposition metrics reflecting the directional changes in DAP and DML. These features were selected based heavily on insights of prior research such as Ando et al. (2023).

Displacement magnitude was computed as the Euclidean norm of DAP and DML. Instantaneous velocity was derived from the temporal derivative of the displacement magnitude, and decomposition was defined as the rate of directional change between DAP and DML components. All features were exported into individual files and consolidated into a master dataset for classification and thresholding. These features are well-established in biomechanical literature and collectively capture the amplitude, directionality, and dynamics of postural sway, providing robust inputs for both supervised and heuristic classification methods. Multiple time-domain features were computed from CoP and accelerometer signals. These included root mean square (RMS), sway range, features selected based on their relevance to postural instability detection and precedent in wearable sensor applications (Ozdemir & Barshan, 2014). Feature sets were continuously refined across iterations based on classifier performance and deployment feasibility.

3.9.1.5 Stability Classification and Labelling

Two complementary methods were employed for sway condition labelling. The primary method used heuristic labels derived from filename conventions, which assigned sway types into four categories: 0 (Stable), 1 (Antero–Posterior), 2 (Medio–Lateral), and 3 (Unstable). As a fallback, K-means clustering with k = 4 was applied to a subset of standardized features (D_RMS, AP_range, ML_range, and CEA_95) to uncover latent sway patterns in cases of inconsistent or missing labels. Cluster centroids were then manually matched to the appropriate classes based on dominant feature trends. This dual strategy ensured flexibility and robustness in class labelling, accommodating both structured and exploratory analyses across variable data sources.

3.9.1.6 ROC-Based Threshold Optimization

To enhance the interpretability and deployment readiness of the classification scheme, Receiver Operating Characteristic (ROC) analysis was used to identify optimal feature thresholds. A Leave-One-Class-Out strategy was applied, and Youden's J statistic was used to determine thresholds that maximized class separation. Key metrics such as True Positive Rate (TPR), False Positive Rate (FPR), and Area Under the Curve (AUC) were recorded for each feature and class. Additionally, a Relative Importance (RI) score was computed for each feature, for its discriminative value using Equation 3.22, as well as the accuracy (Q%) using Equation 3.23 (Ando et al., 2023):

$$RI\% = 100(\frac{\frac{1}{N}\Sigma_{i=1}^{N}\left|JF_{i,q} - Jth_q\right|}{\max\left(JF_{i,q} - Jth_q\right)}) \tag{3.22}$$

$$Q\% = 100[1 - \frac{\Sigma_{i=1}^{N}\gamma i}{N}] \tag{3.23}$$

where, N is number of considered patterns or classes; $JF_{i,q}$ is value of q feature for I pattern; Jth_q is threshold related to q feature. This formulation accounts for the proximity of a feature's performance to the optimal threshold across the dataset. By extracting these interpretable cutoffs and corresponding importance values, the classifier outputs become directly usable in low-power, real-time applications, without the need for opaque black-box decision layers.

3.9.1.7 Classifier Training and Hyperparameter Tuning

Five supervised learning models, Decision Tree, k-Nearest Neighbours (k-NN), Support Vector Machine (SVM), Ensemble (Boosted Trees), and Naive Bayes, were trained and optimised. Each model underwent three-fold hyperparameter tuning followed by five-fold cross-validation for performance evaluation. Key tuning parameters included the maximum number of splits and leaf sizes (Decision Trees), the number of neighbours and distance metrics (k-NN), kernel selection and box constraints (SVM), number of boosting cycles and learning rates (Ensemble), and distribution assumptions per feature (Naive Bayes). This multi-model approach ensured broad algorithmic coverage, balancing generalizability, computational load, and interpretability. Cross-validation safeguards against overfitting while providing realistic performance estimates.

3.9.1.8 Data Export and Visualization

All outputs, including cleaned datasets, feature tables, outlier logs, ROC curves, and classifier evaluation plots—were saved in structured formats for post-analysis. Visualizations were exported in both .png and MATLAB .fig formats to support thesis documentation and supplementary review. Although models like Random Forest showed promising accuracy during benchmarking, their memory footprint and non-deterministic behavior rendered them unsuitable for real-time deployment on embedded platforms. By synthesizing the design logic of Ando et al. (2023) high-accuracy model with practical system constraints, this study yielded a threshold-based classification pipeline that preserved essential biomechanical rigour while enabling interpretable, real-time execution in exosuit-assisted postural control applications.

3.9.2 Classifier Selection

Although the dataset was pre-labelled, an initial attempt was made using unsupervised k-means clustering to explore natural groupings in the feature space. However, the clustering results showed low consistency with the actual labels (≈ 27.5%), indicating that the feature distribution did not support unsupervised separation. Subsequently, a supervised classification approach was adopted. Among these, Random Forest and SVM achieved the highest mean classification accuracy at 63.77% and 63.70%. respectively. However, its runtime complexity and memory demands rendered it unsuitable for real-time embedded use. KNN also performed comparably (63.0%) but similarly imposed constraints on embedded compatibility. Besides, given the limitations of machine learning models in terms of execution time, memory footprint, and real-time deployment feasibility, a simplified threshold-based model was formulated. This approach utilised manually derived decision rules on a reduced feature set, informed by iterative visual analysis and domain-specific heuristics. The rule-based model achieved an improved classification accuracy of approximately 70%, outperforming all tested machine learning models while meeting the computational constraints of the ESP32 microcontroller. The final model offered interpretability, low power consumption, and robust real-time performance in embedded applications. Logic based on Ando et al (2022), where the thresholds of AP sway (DAP and ax) and ML (DML and az) were derived from the vast dataset, via their means + 2 SD of each feature (Appendix A).

3.9.3 Classifier Evaluation and Rationale

Despite the presence of labelled training data, k-means clustering was initially employed to evaluate intrinsic separability in the feature space. This unsupervised method yielded an accuracy of only 27%, affirming that sway categories were not linearly separable without explicit supervision. Subsequent supervised classifiers yielded varying results, summarised in Table 3.5. Model selection was guided by three primary criteria: classification performance, computational complexity, and deployment feasibility. The strengths and limitations of each model are summarised in Table 3.6, along with their alignment to methods described by Ando et al. (2023).

Table 3.5: Supervised Classification Model Performance

Model	**Accuracy (%)**
Random Forest	63.77
K-Nearest Neighbours (k=15)	63.09
Decision Tree	60.03
Naïve Bayes	58.11
Logistic Regression	55.96
Linear Discriminant Analysis	55.64

Table 3.6: Classifier Rationale

Classifier	**Rationale**
KNN	Captures local non-linear decision surfaces
Decision Tree	Easily interpretable, rule-extractable
Random Forest	High accuracy via ensemble learning
Naïve Bayes	Efficient; suitable for high-dimensional data
LDA	Assumes linear separability; interpretable
Logistic Regression	Baseline for linear models

Although Random Forest and KNN achieved the highest accuracy, their computational burdens made them suboptimal for embedded implementation. Furthermore, the dataset may be small (n=37), non-stationary features, inter-class imbalance, and overlapping class boundaries constrained the generalisation capacity of all supervised models. These challenges mirror observations by G. Prisco et al. (2025) and Ando et al. (2023), who both identified limitations in applying traditional ML classifiers to dynamic postural contexts without advanced preprocessing and multimodal sensor fusion.

3.9.4 Algorithm Preliminary Data Analysis

To enhance generalisability across individuals, subject-specific anthropometric normalisation was incorporated into the preprocessing stage. Parameters representing H1 and H2 body segment lengths, averaged in Table 3.7 and Table B-3 the gross measured data, were estimated via linear regression from known height values. Regression analysis, shown in Figure 3.42, confirmed a linear relationship between height and estimated segment length, H1, having R2 of 0.85, and H2 being nearly constant, justifying their use for normalisation in feature computations. Graphs A-2 and A-3 shows the plots by sex.

Table 3.7: Mean Anthropometric Data of Subjects

SEX	Height (m)	H1 (m)	H2 (m)
Female (n=7)	1.613 (± 0.0647)	1.073 (± 0.1025)	0.234 (± 0.0339)
Male (n=29)	1.729 (± 0.0509)	1.169 (± 0.0563)	0.271 (± 0.0206)
Overall	1.707 (± 0.0706)	1.150 (± 0.0763)	0.264(± 0.0269)

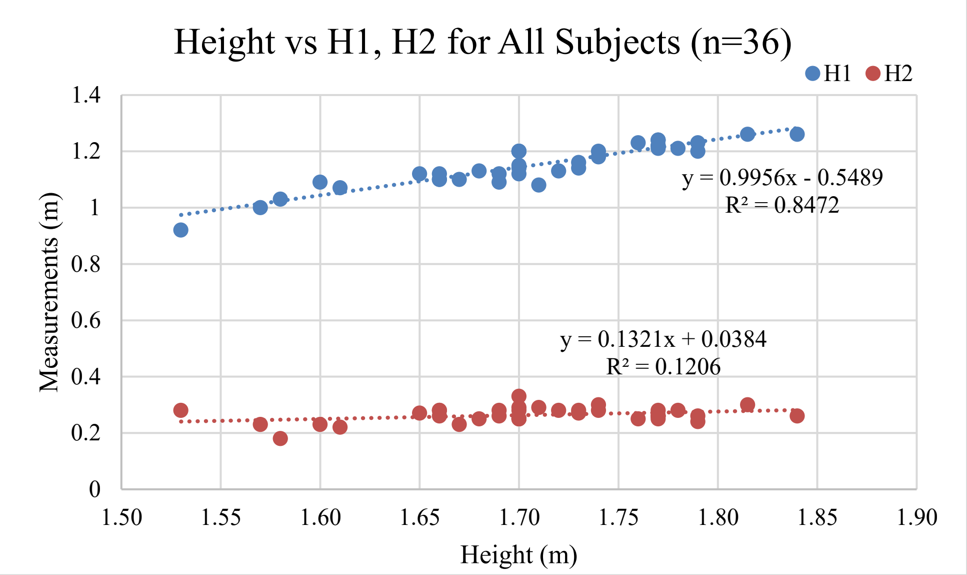


Figure 3.42: Correlations of Height of Subjects with H1 and H2 Measurements

3.9.5 Modifications of Developed Algorithm

Following the limited classification accuracy and significant implementation overhead associated with machine learning-based models, a final algorithmic modification was undertaken. A deterministic, rule-based classification scheme was developed using empirically derived thresholds on select postural sway features. This transition was motivated by the need for a lightweight, interpretable, and real-time compatible algorithm suitable for deployment on resource-constrained microcontroller platforms. The final model employed fixed thresholds applied to key features, including DAP, DML, and accelerometer components ax and az. These thresholds were not arbitrarily chosen; rather, they were iteratively refined through real-time validation, comparison with ground truth labels, and informed by domain-specific literature and ROC curve analysis using metrics such as Youden's Index.

This empirical model demonstrated a classification accuracy of approximately 70%, surpassing all previously tested machine learning classifiers on the same dataset. Notably, it provided consistent performance across subjects, with reduced variance and stable behaviour in diverse testing conditions. The final approach offered several strategic advantages. Firstly, its computational simplicity facilitated real-time processing on ESP32-based microcontrollers without the need for specialised libraries or external computation. Secondly, unlike traditional machine learning models, it required no training phase and could operate deterministically, reducing risks in safety-critical rehabilitation applications. Thirdly, its structure allowed for intuitive interpretability, supporting clearer communication of system decisions to end-users and clinical practitioners.

The algorithm's development followed three main methodological iterations. The initial approach utilised k-means clustering to explore natural groupings in the feature space; however, it yielded poor alignment with true class labels, achieving only 27% classification accuracy. Subsequently, a second approach involved implementation of supervised learning classifiers, including KNN, RF, DT, SVM, NB, LDA, and Logistic Regression. These models were evaluated through 5-fold cross-validation with individually tuned hyperparameters. Although Random Forest yielded the highest accuracy (63.77%), it was computationally intensive and unsuitable for embedded deployment. The final iteration, therefore, adopted a rule-based framework leveraging threshold comparisons for state classification. The transition to this method was driven by multiple considerations: hardware limitations (e.g., limited memory and processing power of ESP32 boards), the need for fast and interpretable decision logic, and empirical findings suggesting that a well-tuned thresholding approach could match or exceed the classification accuracy of more complex models within this application context.

While the threshold-based method lacks generalisability to external datasets without recalibration, its performance within the controlled experimental environment proved reliable and efficient. The use of ROC-based threshold optimisation and iterative domain-informed tuning provided a strong balance between simplicity and classification fidelity. The method proved especially suitable for real-time postural correction tasks, where latency, consistency, and interpretability are prioritised over black-box generalisation. In summary, although various classification paradigms were explored, including unsupervised and supervised models, the final empirical approach emerged as the most practical and reliable within the constraints of embedded deployment. Its deterministic nature, computational efficiency, and interpretability offer substantial advantages for assistive technologies targeting postural correction in static balance tasks.

3.10 Data Collection and Selection Criteria

This section details the methodology adopted to evaluate postural sway under various task and support conditions across three sequential experiments. These include: (1) preliminary sway data acquisition to establish baseline sway profiles for classification algorithm training, (2) validation of sway measurement consistency under increased task difficulty, and (3) assessment of the active back support exoskeleton during balance tasks. Each experiment is framed with equipment justification, participant selection criteria, and procedural overview.

3.10.1 Preliminary Sway Data Collection Protocol (n = 36 Subjects)

The objective of the first experiment was to collect kinematic data characterising postural sway across multiple induced sway conditions, enabling the training of a sway classification algorithm. For this, we employed an MPU6050 IMU and the SONY MOCOPI motion capture system. The MPU6050, widely recognised for its accuracy and cost-effectiveness in wearable movement tracking, was mounted on the sternum to capture trunk accelerations and angular velocities. The MOCOPI system provided full-body spatial data for validating IMU-based measurements. Participants also performed tasks on a wobble board to simulate dynamic instability and provoke distinct sway patterns for analysis. Eligibility for participation was defined by exclusion criteria designed to eliminate confounding biomechanical factors as depicted in Table 3.8, and trial conditions in Table 3.9. Total 36 subjects of 1.707 (± 0.0706) cm were included.

Table 3.8: Exclusion Criteria

Criteria	Description
Height	Within 15th–90th percentile Malaysian (155–183 cm)
BMI	Non-extreme (17.0–28.0)
Injury History	No recent injuries affecting ankle, hip, or back stabiliser muscles

Table 3.9: Trial Conditions and Number of Trials

Condition	Trial Duration	Reps	Notes
Still Stance (STAB)	60 sec	3	Stable
Anteroposterior (AP)	60 sec	3	Forward-backward sway
Mediolateral (ML)	60 sec	3	Side-to-side sway
Unstable (UNST)	60 sec	3	Random/unstable sway

Participants were first screened and briefed before providing written informed consent. The IMU was securely attached to the upper sternum, while six MOCOPI sensors were positioned according to manufacturer specifications as depicted in Figure 3.44. After a familiarization phase, participants performed four randomized stance trials: a still stance (STAB), anteroposterior sway (AP), mediolateral sway (ML), and an unstable wobble board stance (UNST). Each trial lasted 60 seconds and was repeated three times, with one-minute rest periods between trials, all according to flowchart Figure 3.43. All kinematic data were logged concurrently from both IMU and MOCOPI systems.

3.10.2 Preliminary Sway Data Validation Protocol (n=5 subjects)

The second experiment aimed to validate the reliability of sway detection across standard balance tests of increasing difficulty. The same hardware setup was used as in Experiment 1. The trial design was based on clinically accepted postural control tests, including eyes-open and eyes-closed conditions in both bipedal and single-leg stance. Inclusion criteria mirrored those from Experiment 1 to ensure data consistency. Each participant was instrumented with an IMU and MOCOPI sensors. They completed four conditions in randomized order: normal stance with eyes open, normal stance with eyes closed, single-leg stance with eyes open, and single-leg stance with eyes closed. Each trial lasted 30 seconds and was repeated three times. The resulting data provided validation for sway feature stability across progressively challenging tasks, serving as a benchmark for classifier robustness.

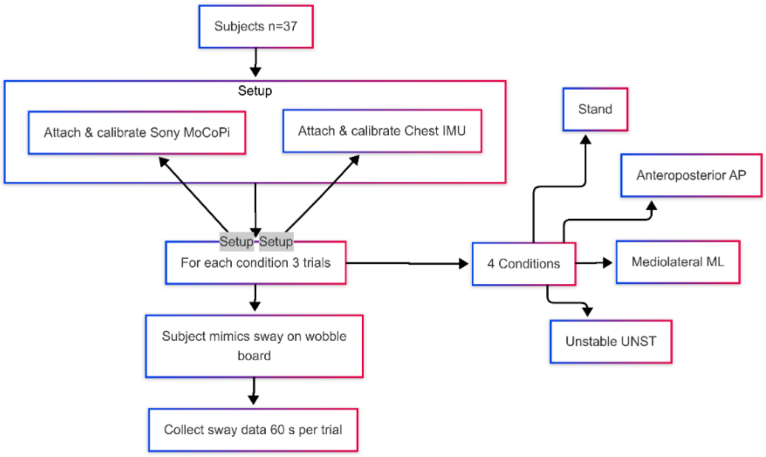

Subjects n=37
Setup
Attach & calibrate Sony MoCoPi
Attach & calibrate Chest IMU
Setup-Setup
For each condition 3 trials
Subject mimics sway on wobble board
Collect sway data 60 s per trial
4 Conditions
Stand
Anteroposterior AP
Mediolateral ML
Unstable UNST

Figure 3.43: Flowchart of Sway Data Collection

3.10.3 Postural Sway Exoskeleton System Testing Experiment Protocol – Back Support Exoskeleton (n=15)

The final experiment investigated the effects of a lightweight active back support exoskeleton on postural stability during static balance conditions. The centre of pressure (CoP) was measured using a Nintendo Wii Balance Board, which has been validated for intra-subject CoP comparisons in time-domain sway analysis (Ando et al., 2022; Bartlett et al., 2014; Leach et al., 2014). Although not suited for clinical diagnostics, its utility in research-grade postural analysis has been demonstrated in multiple studies. Complementary kinematic data were collected using the same IMU and MOCOPI configuration. Muscle activation patterns were assessed using a Delsys Trigno sEMG system. Surface electrodes were preferred for their non-invasive nature, and monopolar configurations were employed due to their superior intermuscular coherence compared to bipolar setups (Mohr et al., 2018). Exclusion criteria remained consistent with prior experiments. Participants first completed a preparation phase, during which anthropometric data were recorded and sEMG sensors were applied to the tibialis anterior, medial gastrocnemius, and optionally, the erector spinae and abdominal muscles. The exoskeleton was then calibrated and donned, and donning time was recorded to assess practical usability.

Each participant completed balance trials in four stance conditions: normal stance with eyes open, normal stance with eyes closed, tandem stance with eyes open, and tandem stance with eyes closed (limited to 25 seconds for safety), shown in Figure 3.44. These were tested across three device states: exoskeleton powered ON, exoskeleton powered OFF, and no exoskeleton. Each configuration was repeated three times, yielding 36 total trials per participant according to Figure 3.45. CoP data were synchronized with IMU, MOCOPI, and sEMG recordings. Following data acquisition, sensors were removed, and participants were invited to provide feedback. Signal processing involved outlier removal using a Hampel filter, signal smoothing with a fourth-order Butterworth low-pass filter, and root mean square (RMS) feature extraction from the EMG data, including mean and standard deviation metrics.

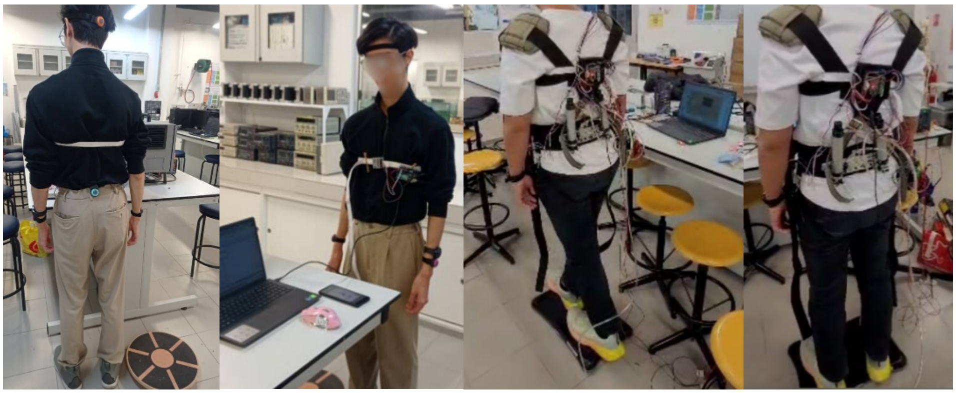


Figure 3.44: Sway Data Collection (left); Functional Test (Right)

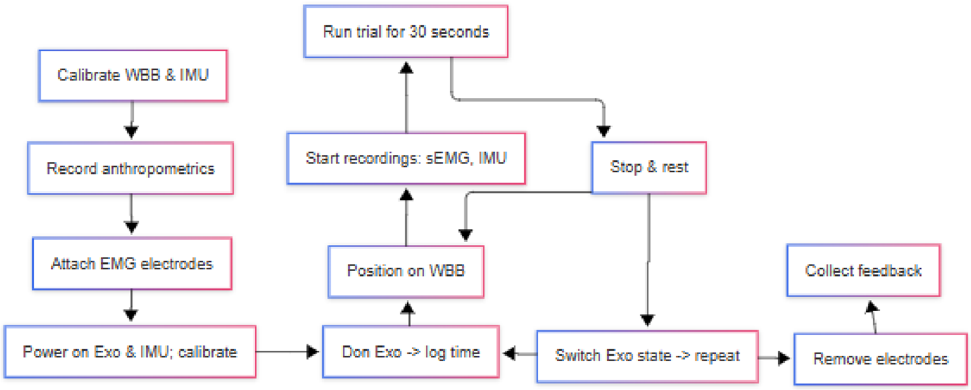


Figure 3.45: Flowchart of Functionality Test

3.11 Project Budget

The prototype was developed with a total expenditure of RM404.71 detailed in Table 3.10. Given the limited funding, component selection was guided by cost-efficiency and availability. Major expenses were allocated to pneumatic hardware, including solenoid valves, tubing, and a portable air compressor. Additional costs covered a basic microcontroller, inertial measurement units (IMUs), wearable straps, and lightweight structural supports. Open-source software frameworks were used for interface development to avoid licensing costs. Components were sourced from local suppliers and online platforms to minimise shipping fees. Due to budget constraints, alternatives to higher-grade materials such as carbon fibre were employed, and electric actuators were not implemented. These trade-offs affected system performance in areas such as weight, battery life, and actuator precision but allowed the construction of a functional prototype sufficient for preliminary testing and demonstration.

Table 3.10: Project Budget

Item	QTY	Price (RM)
(A) Pneumatics		
Xiaomi Pump 2	1	159.00
MAL Mini Air Pneumatic Cylinder Aluminium Bore 16mm 20mm 25mm 32mm Single Rod Double Acting Stroke (25mm, 75mm) -Ext Warranty @ 1.24	2	57.22
Solenoid Valves	2	20.90
SL Pneumatic Throttle Valve 1/8 1/4 3/8 1/2 Male Thread Fitting Air Flow Speed Controller 4-10mm Tube	2	4.90
PC Pneumatic Fitting Push Fit Hose Tube Connector Male Thread M5 M6 1/8 1/4 3/8 1/2 Air Quick joint	4	3.08
PU Tubes (1m)	2	2.44
(B) Electronics	**Total (A)**	**247.54**
3.7V18650 Lithium-Ion Rechargeable Battery 4860Mwh Large Capacity Long-Lasting Handheld Megaphone Amplifier	1	24.26
ESP32	1	15.00
10,000mAh Powerbank	1	13.00
SD Card Module and SD card	1	10.00
6DOF MPU 6050 GY-521 3 Axis Gyro Accelerometer Sensor Module Arduino	1	9.90
ESP32 BASE (EXPANSION PINS OR TERMINAL BLOCK) FOR 30P & 38P	1	7.97

OLED	1	7.00
LM2596 Buck Converter	1	4.95
(C) Hardware	**Total (B)**	**92.08**
Weightlifting Belt	1	17.00
5mm Nylon Webbing (1m)	4	16.80
25mm Zinc Alloy Press Buckle Small Hardware Tightening Buckle	2	10.80
1pc MAL/CDJ2B-LB Holder Air Cylinder Bracket Mounting Support Bore 16mm 20mm 25mm 32mm 40mm	2	9.36
Metal Japanese Buckle Bag Strap Button Three-Speed Buckle Flat Wire Alloy Bag Strap Adjustment Luggage Hardware Accessories	2	6.23
3mm thick Aluminium plate	1	4.50
L bracket	2	0.40
	Total (C)	**65.09**
Grand Total (RM)		**404.71**

3.12 Summary

This project focused on the development and preliminary validation of a strap-based, back-supported pneumatic exoskeleton designed to assist standing posture and reduce postural sway. The methodology encompassed the mechanical design, system integration, and experimental validation with human participants. The exoskeleton frame was designed using SOLIDWORKS and fabricated from lightweight materials, primarily fabric and straps, along with custom 3D-printed components, to ensure user comfort and ease of donning. Pneumatic actuators were incorporated at the trunk region to provide active postural support. A sway detection algorithm was developed to trigger corrective actuation based on body sway, using data from an MPU-6050 IMU sensor. Actuation control was handled by a microcontroller-based system capable of manual and predefined pneumatic valve control.

To enhance usability, a full-stack graphical user interface (GUI) was developed using Python Streamlit for the frontend and PostgreSQL (pgAdmin 4) for the backend database. Additional features such as offline SD card data logging and real-time OLED display were implemented to improve portability and user convenience. Experimental testing involved human participants performing quiet standing trials under four test conditions: normal stance with eyes open (EO), normal stance with eyes closed (EC), tandem stance with EO, and tandem stance with EC. Each condition lasted 30 seconds and was repeated three times. A Wii Balance Board was used as a low-cost alternative to force plates for measuring Centre of Pressure (CoP) parameters to quantify postural sway. Trigno Delsys surface EMG sensors were placed on key trunk stabilizers—external obliques (EO), erector spinae (ES), and rectus abdominis (RA)—to assess muscle activation trends, although EMG was not integrated into the real-time system. IMU and EMG data were analysed offline to evaluate balance control and muscular effort with and without exosuit assistance.

The project adhered closely to the planned Gantt chart with minimal deviation. The prototype was developed within a budget of RM 500, with a final cost of RM 404.71, making it a cost-effective solution. Overall, the methodology prioritized low-cost development, safety during trials, and the practical feasibility of pneumatic exoskeleton systems for supporting static postural stability in rehabilitation and assistive applications.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the final prototype, encompassing its mechanical design, electronic circuitry, and graphical user interface (GUI). It also details the results and validation of the standing postural sway detection algorithm implemented in the system. In addition, surface electromyography (sEMG) and centre of pressure (CoP) tests were conducted to evaluate the prototype’s performance on human subjects. A comfort and practicality assessment, including donning and doffing time as well as user feedback on wearability, was also performed to assess the system's usability in real-world conditions.

4.2 Full Prototype

The developed prototype integrates pneumatic actuators to provide active back support, utilising a strap-and-fabric-based framework designed for compactness and ease of wear. Control is achieved via an ESP32-based system, while postural sway is detected in real time using a chest-mounted MPU6050 IMU. The system is complemented by a custom control algorithm, sway detection logic, and a GUI to facilitate user interaction and feedback. Figure 4.1 illustrates the fully assembled prototype as worn by a study participant.

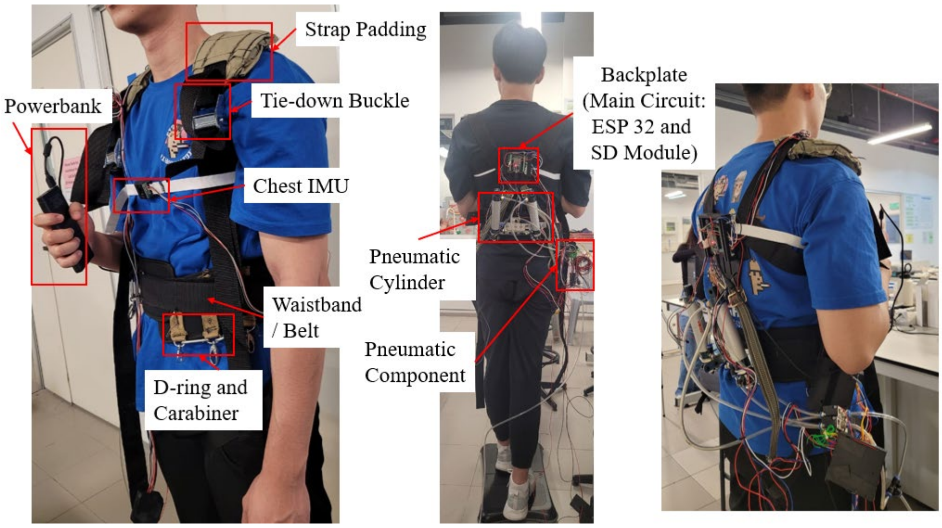
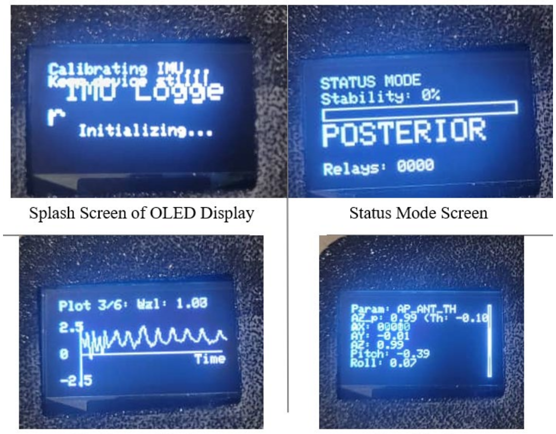


Figure 4.1: Final Prototype of Back Support Pneumatic Exoskeleton on Subject

To don the exoskeleton, the user first wears the chest-mounted IMU, followed by the vest-like harness. The system is secured using adjustable belts and straps to ensure fit and comfort. During setup, the ESP32 is powered via a portable power bank, and the IMU (housed in a protective casing) is briefly placed on a flat surface for calibration prior to use. Once donned, the system initialises automatically. Real-time sway data is acquired and logged either through an onboard SD card or via the GUI interface. This data is simultaneously used as input for the feedback control algorithm, which triggers linear actuation of the pneumatic cylinder upon detection of excessive postural sway. To enhance user accessibility, an OLED display is attached to the wrist. It provides visual status updates and feedback, including system mode, live sway plots, and raw sensor data. Navigation is enabled via mode and scroll buttons, with the interface initialised by a splash screen, as shown in Figure 4.2. The prototype, in its current form, demonstrates cohesive integration of sensing, control, and actuation, offering a compact solution for real-time postural support.



Splash Screen of OLED Display

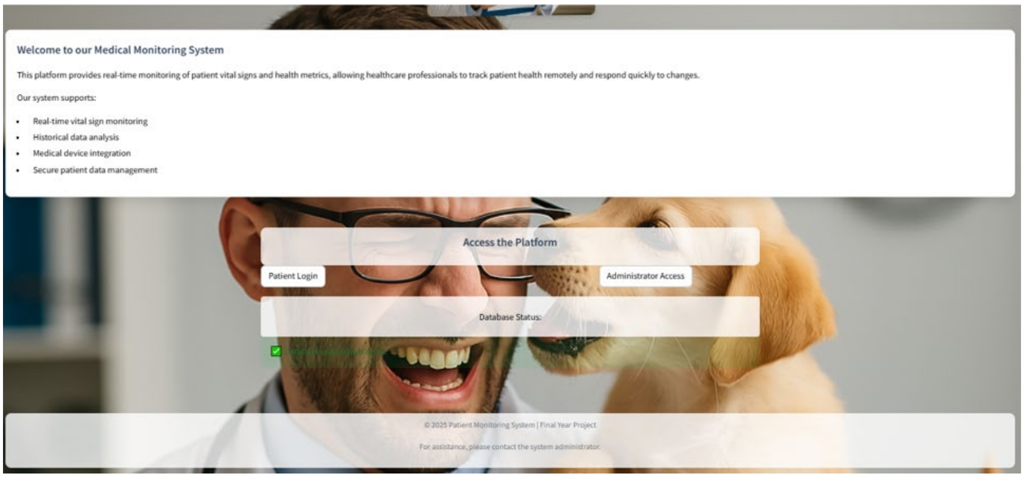
Status Mode Screen

Plot Screen Display

Data Display

Figure 4.2: OLED Display Screen

The user experience of the system was evaluated through practical interaction flows that reflected typical data entry and retrieval tasks User interface main page is shown in Figure 4.3. During testing, the login process was stable and responsive, with authentication consistently completing in under 200 milliseconds, with 2 options of logins, namely user mode (Figure 4.4), where new users can register and admin mode (Figure 4.5), requiring admin authentication. Once logged in, users navigated the dashboard using a sidebar layout (Figure 4.6), which was generally described as intuitive and easy to follow. Feedback suggested that the interface required minimal effort to learn, making it accessible even for first-time users.



Welcome to our Medical Monitoring System

This platform provides real-time monitoring of patient vital signs and health metrics, allowing healthcare professionals to track patient health remotely and respond quickly to changes.

Our system supports:

- Real-time vital sign monitoring
- Historical data analysis
- Medical device integration
- Secure patient data management

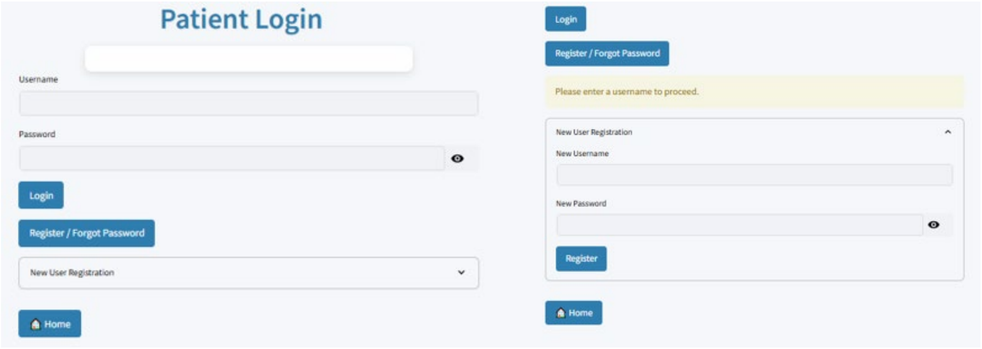
Access the Platform

Database Status:

© 2025 Patient Monitoring System | Final Year Project

For assistance, please contact the system administrator.

Figure 4.3: User Interface Main Page



Patient Login

Username

Password

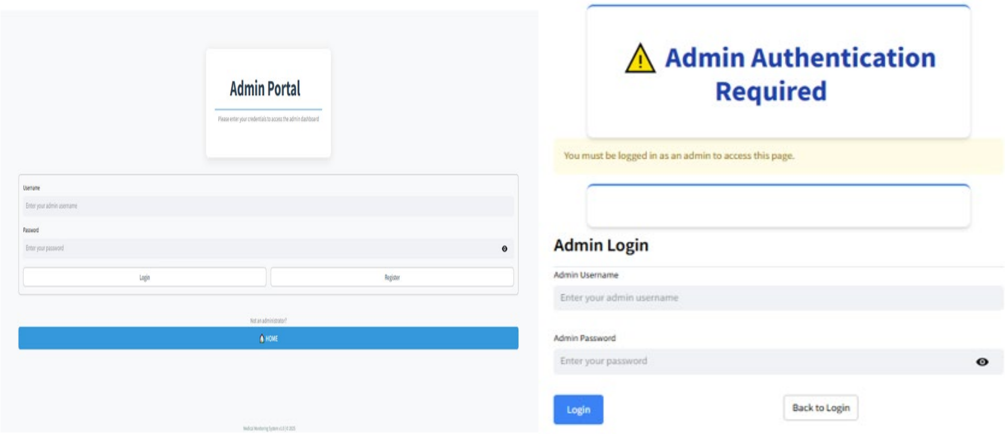
Please enter a username to proceed.

New User Registration

New Username

New Password

Figure 4.4: Patient Login and Registration Page



Admin Portal

Please enter your credentials to access the admin dashboard

Username

Enter your admin username

Password

Enter your password

Not an administrator?

Medical Monitoring System v1.0 | © 2025

⚠ Admin Authentication Required

You must be logged in as an admin to access this page.

Admin Login

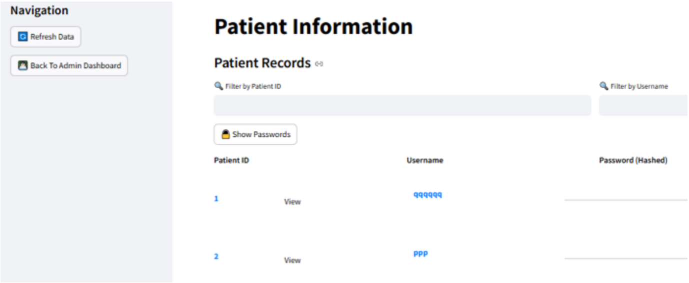
Admin Username

Enter your admin username

Admin Password

Enter your password

Figure 4.5: Admin Login Page



Patient ID		Username	Password (Hashed)
1	View	qqqqqq	
2	View	PPP	

Figure 4.6: Navigation Sidebar of GUI

Record querying functions, including filtering by patient ID, session date, and condition, worked reliably and returned results accurately across all test cases. The data visualisation tools were useful for quick interpretation, bar and box plots allowed comparison between conditions (Figures 4.7 and 4.8), heatmaps showed how signals changed over time, and line charts helped track recovery or progression across sessions, shown in Figure 4.9. Real-time IMU data streaming, running at around 10 Hz, was successfully integrated, providing live updates on orientation and movement parameters, as shown in Figure 4.10. Figure 4.11 shows the trail tracking section of dashboard, which were displayed clearly on the Streamlit dashboard and were useful for monitoring posture or motion in real time.

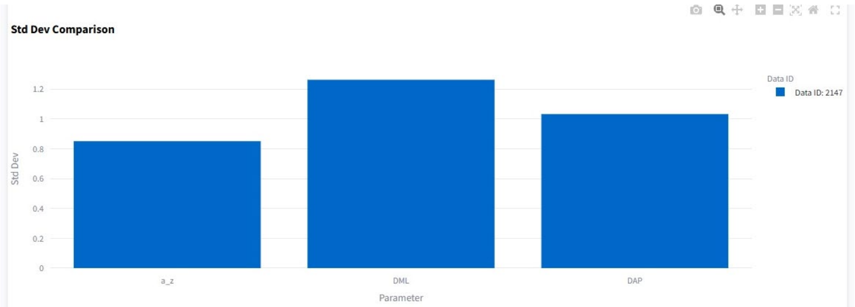


Figure 4.7: Barchart Comparison of Standard Deviation Via GUI

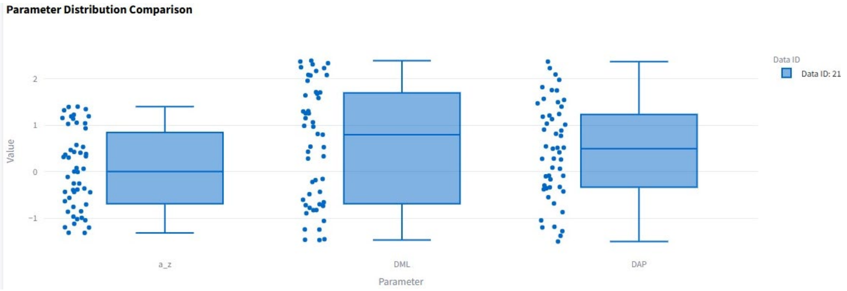
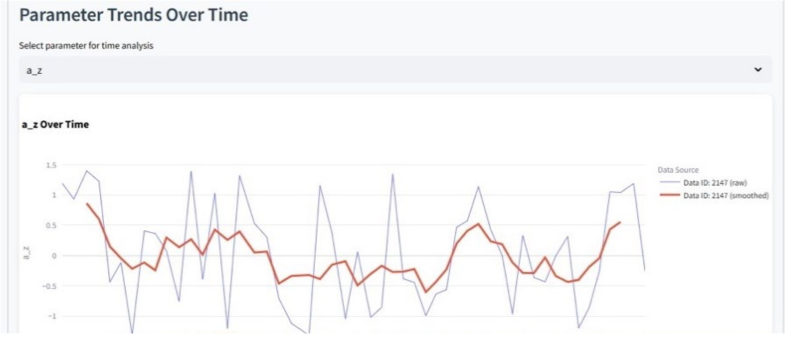


Figure 4.8: Boxplot Visualisation of Distribution Via GUI



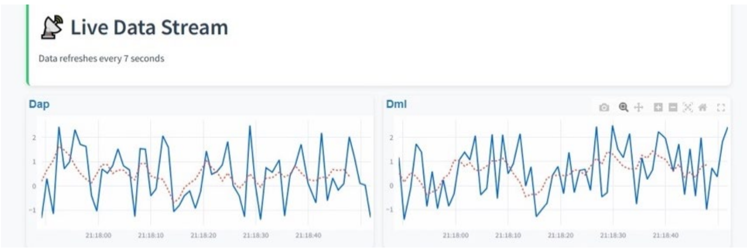
Parameter Trends Over Time

Select parameter for time analysis

a_z

a_z Over Time

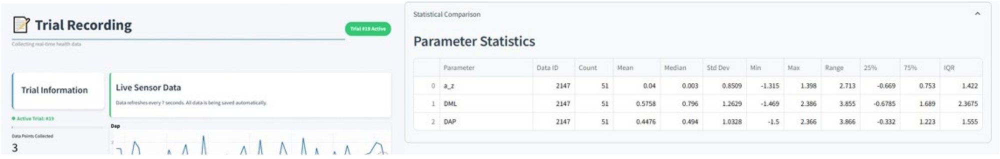
Figure 4.9: View of Historical Data Over Time By GUI



Live Data Stream

Data refreshes every 7 seconds

Figure 4.10: Real-time Data Visualisation on GUI



Trial Recording

Collecting real-time health data

Trial #19 Active

Trial Information

● Active Trial: #19

Data Points Collected

3

Live Sensor Data

Data refreshes every 7 seconds. All data is being saved automatically.

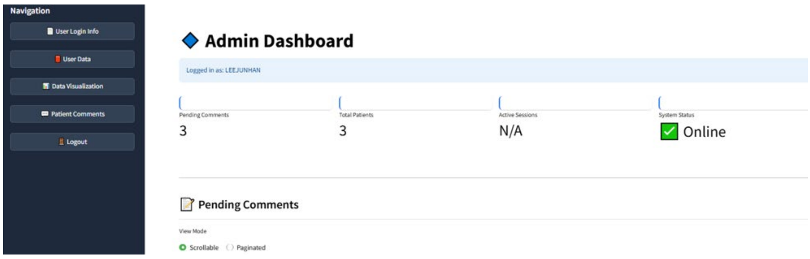
Statistical Comparison

Parameter Statistics

	Parameter	Data ID	Count	Mean	Median	Std Dev	Min	Max	Range	25%	75%	IQR
0	a_z	2147	51	0.04	0.003	0.8509	-1.315	1.398	2.713	-0.669	0.753	1.422
1	DML	2147	51	0.5758	0.796	1.2629	-1.469	2.386	3.855	-0.6785	1.689	2.3675
2	DAP	2147	51	0.4476	0.494	1.0328	-1.5	2.366	3.866	-0.332	1.223	1.555

Figure 4.11: Trial Dashboard Display and Features on GUI

Administrators, via their dashboards (Figure 4.12) were able to add comments (Figure 4.13), annotate data, and view multiple sessions together (Figure 4.14), which helped in reviewing patient history and comparing outcomes. While some interface elements could benefit from further refinement, the system performed reliably and was effective for both data capture and visualisation in real-world testing scenarios.



Admin Dashboard

Logged in as: LEEJUNHAN

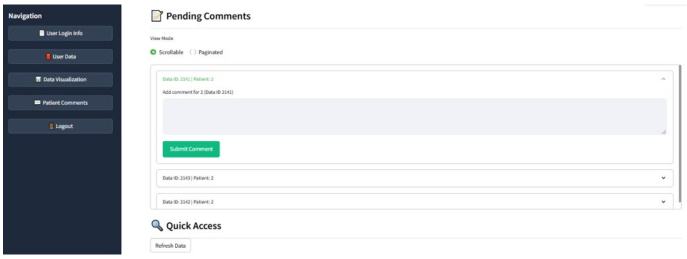
Pending Comments	Total Patients	Active Sessions	System Status
3	3	N/A	✓ Online

Pending Comments

View Mode

● Scrollable ○ Paginated

Figure 4.12: Admin dashboard



Pending Comments

View Mode

● Scrollable ○ Paginated

Data ID: 2141 | Patient: 2

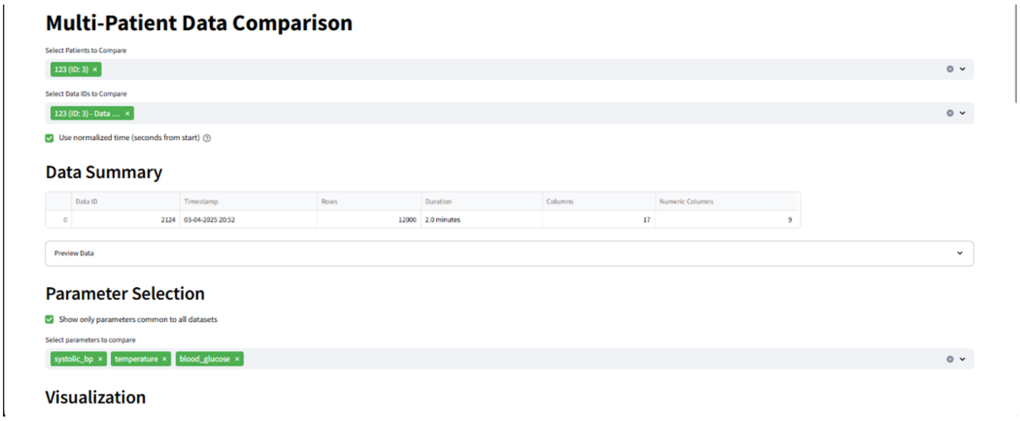
Add comment for 2 (Data ID 2141)

Data ID: 2143 | Patient: 2

Data ID: 2142 | Patient: 2

Quick Access

Figure 4.13: Comment Function on Patient File By Admin



Multi-Patient Data Comparison

Select Patients to Compare: 123 (ID: 3)

Select Data IDs to Compare: 123 (ID: 3) - Data ...

☑ Use normalized time (seconds from start)

Data Summary

	Data ID	Timestamp	Rows	Duration	Columns	Numeric Columns
0	2124	03-04-2025 20:52	12000	2.0 minutes	17	9

Parameter Selection

☑ Show only parameters common to all datasets

Select parameters to compare: systolic_bp, temperature, blood_glucose

Visualization

Figure 4.14: Multi-Data Comparison Function By Admin

4.3 Algorithm Performance

This section evaluates the feasibility of distinguishing postural sway types using centre-of-pressure (CoP)-derived features from a chest-mounted accelerometer. The classification task aimed to differentiate among four predefined sway categories, Stable (STAB), Anteroposterior (DAP), Mediolateral (DML), and Unstable (INST), using a strategy informed by existing biomedical signal processing literature and constraints imposed by embedded machine learning systems. Key considerations included feature dimensionality, model complexity, and computational efficiency.

A total of 37 participants completed three repetitions of each sway condition, yielding 444 labelled trials. From these, non-overlapping 5-second windows were applied using a sliding window approach, resulting in approximately 130,000 labelled samples. While a finer segmentation (e.g., 250 ms windows) could have produced a higher-resolution dataset with over 3.7 million data points, it was excluded due to training and validation time constraints. The selected windowing approach offered a practical balance between temporal resolution and computational tractability. To prevent data leakage and to ensure generalisability across individuals, stratified 5-fold cross-validation was implemented. This ensured that all samples from any given subject were isolated to either training or validation folds, not both.

4.3.1 Feature Landscape and Data Behaviour

The feature set included standard metrics commonly used in postural sway analysis: Root Mean Square (RMS) of acceleration, sway velocity, anteroposterior (AP) and mediolateral (ML) range, and ellipse area. These features have demonstrated utility in quantifying biomechanics risk and instability (Prisco et al., 2023). Visualisation of processed signal curves revealed clear inter-class distinctions. STAB was characterised by flat, low-amplitude traces. In contrast, DAP and DML demonstrated uniaxial oscillations, dominated by the anterior-posterior axis (Ax) and mediolateral axis (Az), respectively. INST, however, presented erratic, multi-directional bursts resembling hybrid patterns of DAP and DML with intermittent, unpredictable deviations, as depicted in scatter plot in Figure 4.15.

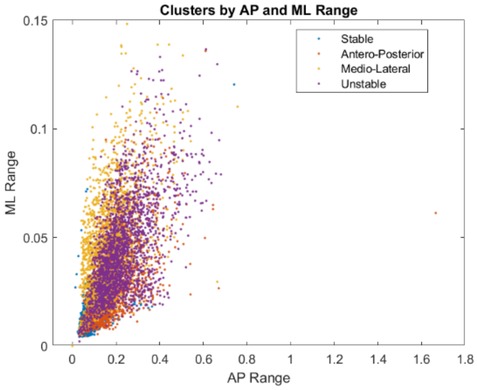


Figure 4.15: Scatter of Labelled Data by Ranges

Signal envelopes and scatterplots provided further insight. RMS and directional signals showed clear trends: DAP was primarily Ax-dominant, DML showed lateral Az excursions with limited axial interference, while INST had high RMS values and abrupt, noise-like shifts across both axes. Figure 4.17 visualises the scatter distributions of directional ranges, highlighting overlaps, particularly between DAP, DML, and INST. The INST class exhibited the widest multidimensional spread with no discernible centroid, while STAB, DAP and DML are easily distinguishable, reinforcing its complexity and classification difficulty. High intra-class variability, especially in DAP and INST, was observed and is attributable to subject-specific compensation mechanisms, such as trunk or upper limb engagement. Additionally, the single chest-mounted IMU limited sensitivity to lower-limb dynamics, which are crucial in sway detection. This aligns with findings by Guo et al. (2022), who emphasised the role of sensor placement, suggesting that pelvic or lower-limb-mounted IMUs may offer superior discriminative power in sway classification tasks.

Feature distribution plots further confirmed that STAB was the most separable class, while DAP, DML, and INST exhibited overlapping feature spaces. This was particularly problematic for multi-class classifiers. The ambiguity of the INST class was reflected in ROC analyses conducted with a K-means clustering baseline. While the area under the curve (AUC) was high for STAB; INST, DAP and DML consistently returned low AUC values, underscoring its weak and fragmented feature identity, shown in Figure 4.16. Subsequent tests were carried out which determined removing either INST or DAP and DML does in fact significantly improve the AUC values.

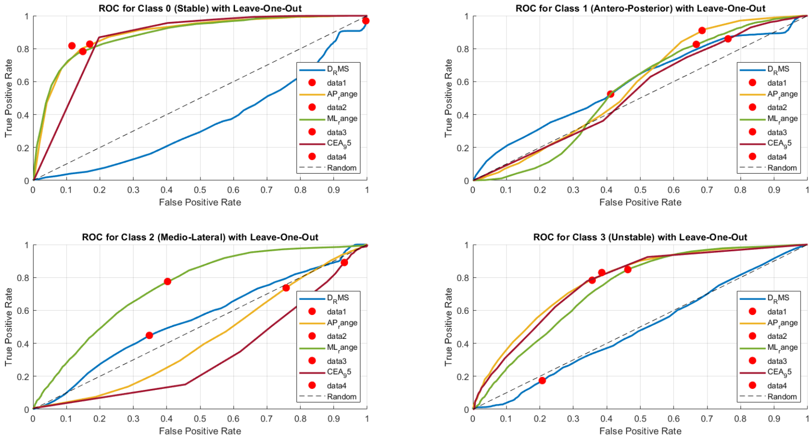


Figure 4.16: ROC Curves of K-means Classifier per Class

K-means clustering with four clusters (matching the true number of sway classes) was used to explore intrinsic feature space separability. STAB formed a tight, distinct cluster, validating its relative uniformity. However, other classes, especially INST, exhibited low inter-cluster purity. DAP and DML formed moderately cohesive but overlapping directional clusters, whereas INST points were dispersed across multiple centroids. These findings collectively suggest that INST, as a sway condition, lacks a cohesive and distinguishable feature identity, posing challenges to both unsupervised and supervised learning approaches. The dense class overlap, particularly among dynamic sway types (DAP, DML, INST), is a compound result of sensor limitations, physiological variability, adaptive motor responses, and intrinsic noise in human balance behaviour.

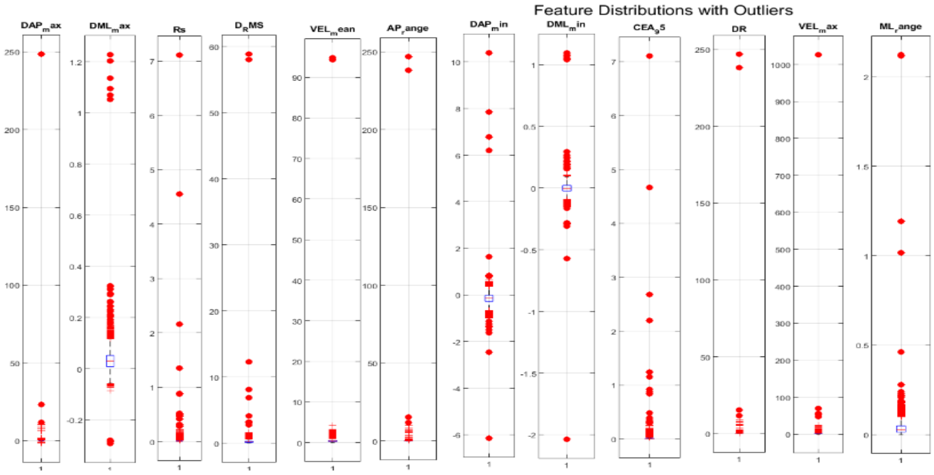


Figure 4.17: Feature Distribution of Sway Parameters

4.3.2 Classifier Evaluation and Cross-Validation Results

Despite extensive hyperparameter tuning, the overall accuracy of machine learning models plateaued around 63–64%, constrained by inter-class feature overlap and the limitations of single-sensor input. This reflects an inherent ceiling imposed by the signal characteristics and subject-level variability rather than model architecture alone. The model-wise cross validation accuracy is shown in Figure 4.18 and Figure 4.19 shows its distribution over the 5 cross validations (K-fold 5), details in Figure C-12. Hyperparameter optimisation was performed using grid search methods, with results visualized via heatmaps to identify performance peaks across parameter combinations. For instance, K-Nearest Neighbours (KNN) yielded optimal performance at k = 15, whereas Random Forest (RF) achieved best results with 100–150 estimators and constrained tree depth, minimizing overfitting. These observations reinforce that class feature overlap inherently limits the benefit of tuning, and model stability is heavily affected by inter-subject variability.

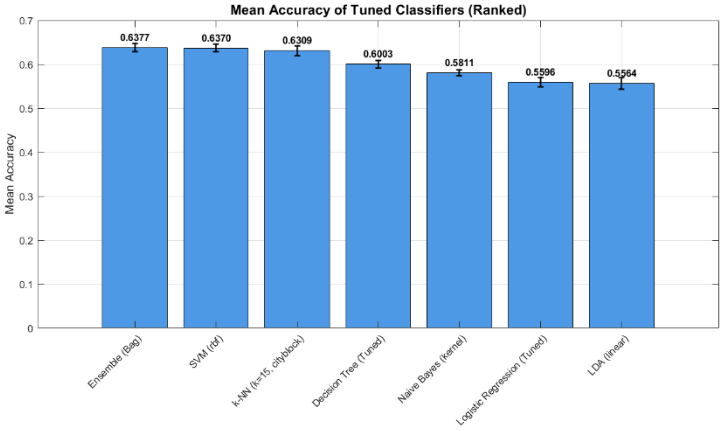


Figure 4.18: Mean Accuracy of Tuned Classifiers

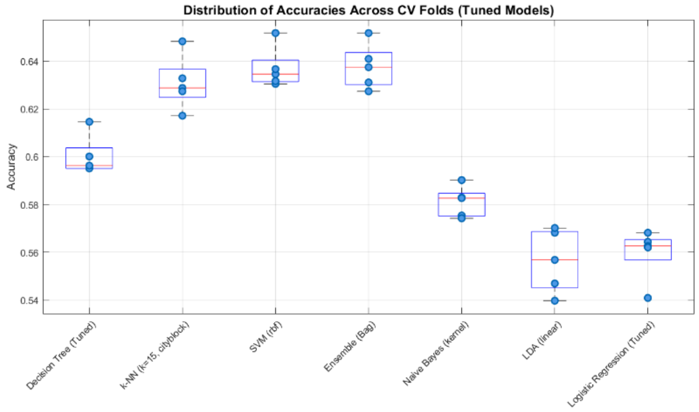


Figure 4.19: Cross Validation Accuracy Distribution per Model

Among all evaluated models, Random Forest achieved the highest cross-validated mean accuracy of 63.77%, leveraging its ensemble framework to improve robustness against noisy and variable data. This aligns with prior findings demonstrating RF's effectiveness in classifying balance impairments via postural sway features (Sun et al., 2019). Notably, bagging methods outperformed boosting methods, with AdaBoostM2 and Bagged Trees producing stable outcomes, while GentleBoost and LogitBoost failed to converge or resulted in 0% accuracy, depicted in Figure 4.20. The mediocre performance of boosting algorithms is due to their sensitivity to misclassified samples, which becomes problematic in high-overlap, imbalanced class environments.

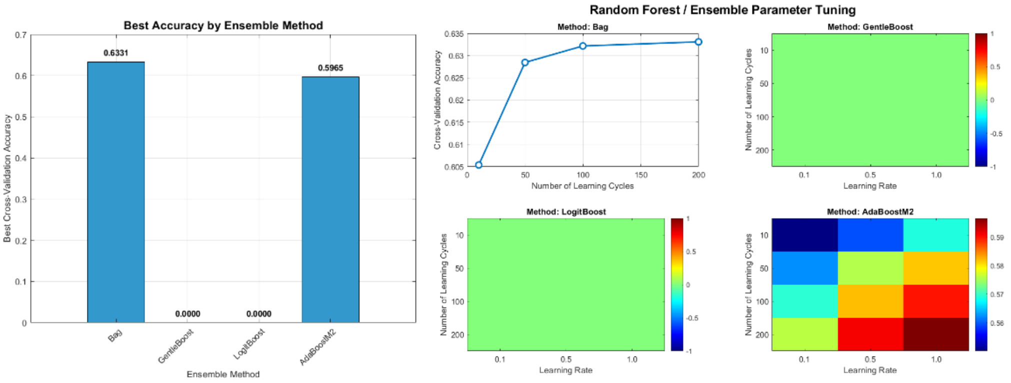


Figure 4.20: Tuning for Random Forest

Support Vector Machine (SVM) followed closely with a cross-validated accuracy of 63.70%. Its strength lies in constructing high-dimensional separating hyperplanes and maximizing classification margins, making it relatively robust to overlapping distributions and class imbalance. However, its performance is highly dependent on careful kernel and regularization parameter selection, especially in the presence of noisy or non-separable data. K-Nearest Neighbours (KNN) achieved a mean accuracy of 63.09%, performing best with Euclidean and Cityblock distance metrics depicted in Figure 4.21. These results suggest that absolute displacement measures are more informative for postural sway classification than angular or correlation-based metrics. While KNN exhibited consistent classification behavior across all classes, it was computationally intensive, which may limit real-time deployment, especially in embedded systems with resource constraints.

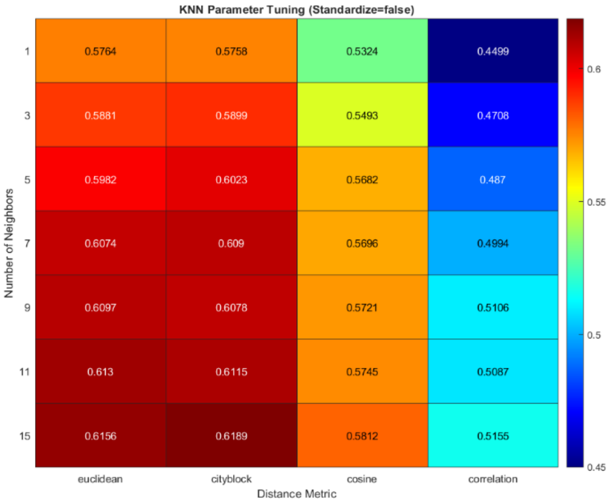


Figure 4.21: KNN Hyperparameter Tuning

Naïve Bayes (NB) classifiers revealed performance discrepancies based on distributional assumptions. Multinomial and Multivariate NB variants performed poorly (~58.11%), due to the continuous and non-discrete nature of the input features. Gaussian NB yielded marginal improvements, while a hybrid approach combining normal distribution with kernel density estimation reached nearly 53% accuracy shown in Figure 4.22. Despite lower overall performance, this suggests that non-parametric density models may still offer utility in overlapping feature spaces with non-Gaussian behaviour.

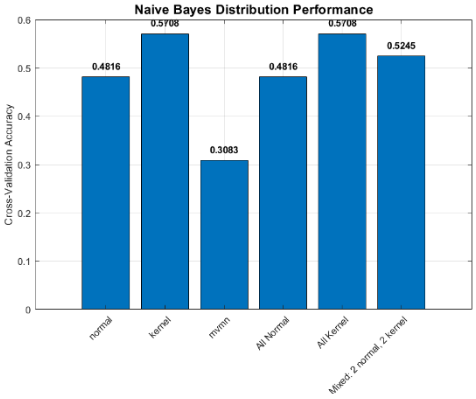


Figure 4.22: Naïve Bayes Classifier Tuning

Decision Trees, although interpretable, achieved only 60.03% accuracy. Despite pruning, they remained prone to overfitting due to the noisy and variable nature of the dataset. Nonetheless, their feature-based decision thresholds, especially for RMS and AP range, aligned with clinically intuitive postural markers as seen in Figure 4.23. Logistic Regression and Linear Discriminant Analysis (LDA) performed in the 55–57% range and struggled most with INST class detection, consistent with their limitations in modelling nonlinear and multi-axial behaviours as depicted in Figure 4.24.

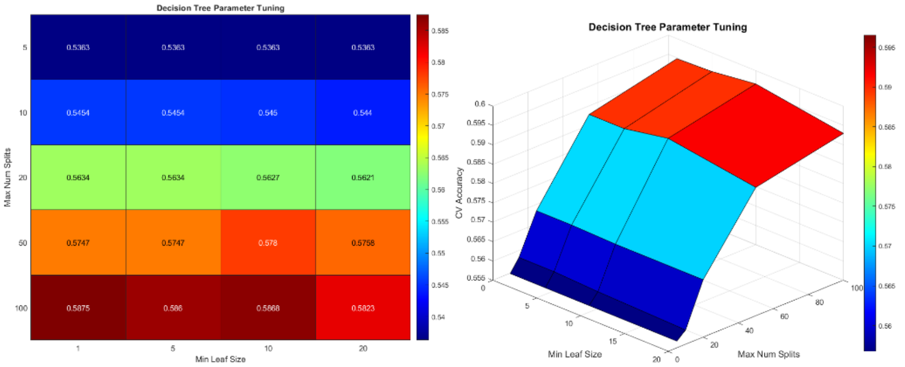


Figure 4.23: Decision Tree Parameter Tuning

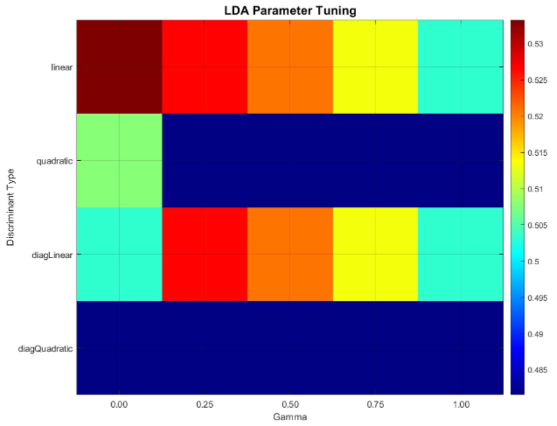


Figure 4.24: LDA Tuning

Among all approaches, the empirical rule-based algorithm, despite being manually tuned, heuristic-driven and non-adaptive, achieved the highest accuracy of approximately 70%. It leveraged compound logic, integrating basic CoP data thresholds with raw accelerometer data to effectively distinguish between classes, especially crucial in detecting the elusive INST class. This model was particularly advantageous for embedded real-time classification, given its deterministic behaviour, low computational demand, and transparent structure, despite its lack of learning ability.

4.3.3 Empirical Model and Threshold Tuning (DAP/DML Ax, Az)

The empirical classifier was developed based on threshold conditions extracted from ensemble mean ± standard deviation envelopes of Ax and Az accelerometer components, as well as displacement-based features for DAP and DML, as visualized in Graphs A-4 to A-17, where Graphs A-4 to A-9 shows distinction of the parameters by class, and Graphs A-10 to A-17 the ensemble means with SD for threshold tuning. Specifically, DAP was identified when Ax exceeded 1.5 standard deviations above the mean and DAP displacement was similarly elevated, provided that Az remained below one standard deviation and DML displacement was suppressed. Conversely, DML was characterized by dominant Az values exceeding their threshold while Ax remained within or below normal limits. INST classification was triggered when both axial thresholds were exceeded simultaneously or when multiaxial surges and erratic fluctuations occurred outside the typical bounds for DAP and DML. These compound logic rules were further reinforced using RMS and ellipse area constraints to reduce false positives. The resulting model demonstrated strong stability across subjects and was notably more effective than machine learning classifiers in detecting unstable sway episodes, where feature ambiguity often led to misclassification. Although lacking adaptability and learning capacity, the model's deterministic structure, low computational footprint, and transparency render it ideal for embedded or wearable systems requiring real-time postural assessment. The success of this heuristic approach underscores the value of integrating domain expertise and physiologically informed rules, particularly in scenarios where sensor constraints and temporal variability limit the effectiveness of purely data-driven methods.

4.3.4 INST Class Performance and Diagnostic Challenges

The INST (Unstable) class posed persistent classification challenges across all models due to its ambiguous and overlapping feature behavior. While prior sections established its multidirectional and erratic signal profile, the deeper issue lies in INST's temporal unpredictability, with abrupt axis shifts, amplitude surges, and intermittent stillness that occasionally mimicked STAB. These dynamic fluctuations made INST highly prone to misclassification, particularly as DAP or DML, in both linear classifiers and ensemble models. Another contributing factor was the use of a single chest-mounted IMU, which limited sensitivity to lower-body compensations and fine-grained balance adjustments. As noted by Andò et al. (2022), such sensor placements are inherently disadvantaged in capturing full-body sway dynamics, especially when subjects deploy hip or upper-limb strategies for balance correction. These compensations introduce non-stationary signal noise, reducing the model's ability to differentiate true instability from intentional movement variability.

While STAB, DAP, and DML showed more class-consistent feature patterns, due to their directional dominance and constrained intra-class variability, INST exhibited significant feature and temporal overlap with all three. This is consistent with mimic-based postural failure findings by Andò et al. (2023), which demonstrated that instability states often defy clean categorical boundaries. These collective findings suggest that INST classification challenges are not merely artifacts of this system's limitations, but reflect a broader problem in modelling human instability episodes.

4.3.5 Class-Specific Feature Mapping and Model Interpretations

Among the four sway classes, STAB demonstrated the most distinct and compact feature representation, characterized by low RMS, minimal sway range, and a small ellipse area, traits that enabled high classification accuracy across all models. DAP and DML also exhibited reasonably strong separability, particularly through their axis-specific features (i.e., elevated AP_range for DAP and ML_range for DML), consistent with biomechanical literature on directional sway (Guo et al., 2022). INST, in contrast, lacked any stable or exclusive feature pattern. While high RMS and irregular sway velocity were observed in many instances, these features also appeared in DAP and DML cases, limiting their diagnostic specificity. Even though the Reliability Index (RI) initially suggested high per-feature consistency for INST, similar to STAB, the RI failed to capture the inter-class confusion caused by INST's broad spread in feature space, Figures 4.25 and 4.26. For example, RMS reliability for STAB was unexpectedly low (~27.5%), despite strong classification performance, whereas features like ellipse area showed higher RI but were not discriminative enough to isolate INST in a multiclass context. This initially suggested that INST, despite its classification challenges, demonstrated some internal consistency in its feature distribution.

However, contrary to this interpretation, model performance significantly improved when the INST class was excluded from the classification task. Upon removing INST and re-training the models in a reduced three-class (STAB, DAP, DML) scenario, all remaining classes exhibited better accuracy, precision, and inter-class separability—even though no additional features were introduced or removed. This implies that the presence of the INST class contributed considerable noise and confusion within the model's decision boundaries. This paradox, where INST appears reliable in RI evaluation yet destabilizes overall classification, highlights a critical distinction between intra-class consistency and inter-class separability. While INST may have consistent internal patterns, those patterns heavily overlap with both DAP and DML in the shared feature space, undermining its practical discriminability in multiclass settings. Although visualizations were not included for this analysis, the improved performance metrics across all three remaining classes reinforce the conclusion that INST acts as a confounding factor in both feature space and classifier learning. This finding underscores the complexity of modelling unstable postural behavior and further supports the case for either recharacterizing INST with additional temporal features or treating it as a separate anomaly detection problem rather than a strict classification target.

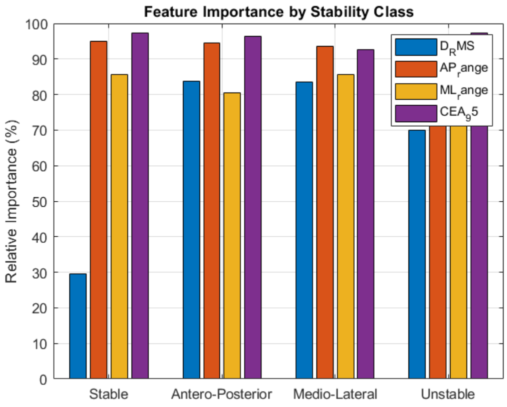
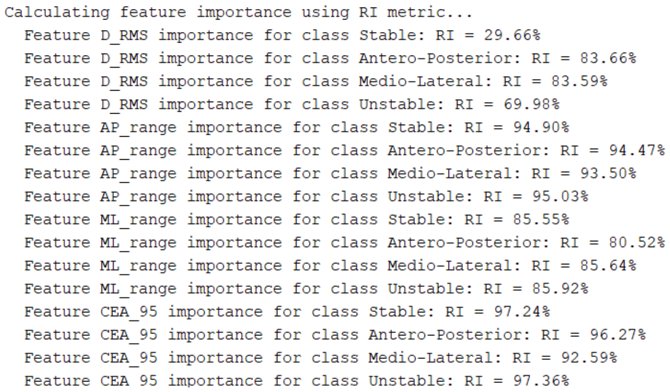


Figure 4.25: RI% of Features per Class



```
Calculating feature importance using RI metric...
  Feature D_RMS importance for class Stable: RI = 29.66%
  Feature D_RMS importance for class Antero-Posterior: RI = 83.66%
  Feature D_RMS importance for class Medio-Lateral: RI = 83.59%
  Feature D_RMS importance for class Unstable: RI = 69.98%
  Feature AP_range importance for class Stable: RI = 94.90%
  Feature AP_range importance for class Antero-Posterior: RI = 94.47%
  Feature AP_range importance for class Medio-Lateral: RI = 93.50%
  Feature AP_range importance for class Unstable: RI = 95.03%
  Feature ML_range importance for class Stable: RI = 85.55%
  Feature ML_range importance for class Antero-Posterior: RI = 80.52%
  Feature ML_range importance for class Medio-Lateral: RI = 85.64%
  Feature ML_range importance for class Unstable: RI = 85.92%
  Feature CEA_95 importance for class Stable: RI = 97.24%
  Feature CEA_95 importance for class Antero-Posterior: RI = 96.27%
  Feature CEA_95 importance for class Medio-Lateral: RI = 92.59%
  Feature CEA_95 importance for class Unstable: RI = 97.36%
```

Figure 4.26: Matlab Snippet of Feature Importance for Classes

When INST was excluded from the classification problem, all three remaining classes (STAB, DAP, DML) saw improved performance. This supports the conclusion that INST's inclusion introduces confounding overlap that disrupts model decision boundaries, despite appearing reliable in isolation. Random Forest models partially alleviated this issue by modelling complex feature interactions, but even they struggled with recall for INST, underscoring its intrinsic ambiguity. These observations collectively suggest that static classifiers, relying solely on summary features, are insufficient for capturing the dynamic instability seen in INST. Future work should explore temporal modelling architectures, such as Hidden Markov Models (HMMs) or Long

Short-Term Memory networks (LSTMs), which are better suited to handle the nonlinear transitions and state-dependent behaviours inherent in human instability. Coupling such models with multi-sensor systems may further improve detection by capturing a more holistic representation of postural dynamics.

4.3.6 Inter-Subject Variability and Empirical Model Insights

Significant inter-subject performance variability was observed, particularly in the INST classification. In some participants, INST was never detected, while in others, false positives occurred even under stable conditions. This heterogeneity was confirmed via subject-level quantile performance plots, reinforcing that sway classification is not solely a feature engineering or model selection issue, but one deeply influenced by individual biomechanics and behavioural compensation. Factors such as differing balance strategies (e.g., ankle vs. hip), sensor-to-movement misalignment, and variable body mechanics all contributed to inconsistent classification accuracy. These limitations align with Gattinara et al. (2022), who noted that ML-based classifiers, while effective under controlled or disease-specific conditions, Parkinson's to be exact, tend to falter when exposed to subject-level variability and treatment-induced movement differences. Figure 4.27 shows the Q% per subject indicating high variance between subjects.

An empirical threshold-based model, developed using logical rules on compound CoP-derived features (e.g., directional dominance in Ax and Az), showed relatively better stability across subjects. Although not cross-validated due to its heuristic nature, this model performed consistently, particularly in detecting directional sway. Its success may lie in its interpretable decision boundaries and reduced reliance on complex feature transformations. These observations suggest that while machine learning offers scalability and pattern recognition capabilities, integrating domain-informed rule-based heuristics—especially in subject-agnostic systems—can provide robustness in real-world deployment settings. This hybrid approach warrants further investigation, particularly for wearable balance monitoring applications under diverse population settings.

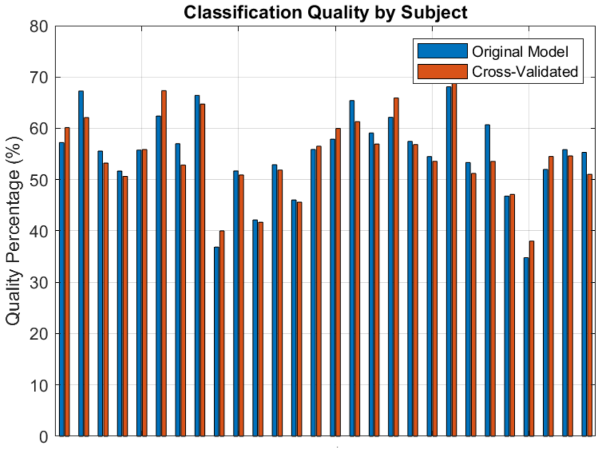


Figure 4.27: Q% of K-means by Subject

4.3.7 Summary and Observations

In conclusion, Random Forest, and K-Nearest Neighbours (KNN) were the most consistent machine learning classifiers, outperforming boosting-based ensembles, and linear models in cross-validation. Naïve Bayes variants performed less reliably overall, though kernel-based variants showed minor improvements. The threshold-based empirical model recorded the highest validation accuracy (~70%), based on static rules derived from observed class-specific feature patterns. While it lacked adaptability, it was straightforward to implement and maintained stable performance across trials.

INST classification remained the main source of error, with high confusion due to overlap with DAP and DML features. The lack of consistent axis dominance and the irregular temporal nature of instability contributed to poor model separation. Single-IMU input and subject variability further limited classifier generalisability. Compared to frameworks like Andò et al., differences in performance are likely due to environmental noise (e.g., sensor placement inconsistencies, movement artifacts), limited data volume, class imbalance—particularly under-sampled INST data—and absence of real-time interaction or mimic-loop training. Overall, machine learning models reached a performance ceiling under current data and setup constraints. The rule-based classifier, while static and manually tuned, remained the most reliable under validation conditions. Temporal models or multimodal sensor input may be necessary to improve future classification of dynamic sway behaviour, especially for instability detection.

4.4 Exosuit Comfortability and Practicability

To evaluate the wearability and usability of the developed exosuit, donning and doffing times were recorded and analysed across multiple trials. These metrics are essential for determining the practicality of wearable assistive devices, especially in real-world settings where time efficiency and user independence are critical. Table 4.1 summarises the donning and doffing times across all subjects with time by subject appended in Table B-4. The average don time was 199 seconds (SD = 88 s), and the average doff time was 68 seconds (SD = 17 s). Upon removal of statistical outliers, primarily due to hesitation or inexperience in early trials, the adjusted mean times improved to 182 seconds (SD = 60 s) for donning and 66 seconds (SD = 16 s) for doffing. This corresponds to a mean improvement of 8.92% for donning and 2.99% for doffing, indicating increased user confidence and efficiency after repeated use. While full-body industrial exoskeletons often report donning times of up to 10 minutes and doffing times under 5 minutes, back-support exoskeletons, being lighter and simpler, typically require only around 35 seconds to don and 7 seconds to doff Chung et al., 2024). The prototype exosuit in this study, although slightly slower, falls within a reasonable range given its early-stage construction and design limitations. Notably, the exosuit used in this study was assembled as a proof-of-concept with a limited budget (RM404), resulting in a semi-manual, strapping-based harness that lacked quick-release mechanisms or rigid frames.

Table 4.1: Summary of Don Doff Timing of Users

	Don Time (s)	Doff Time (s)
All Subjects		
Mean	199	68
SD	88	17
Removed Outliers		
Mean	182	66
SD	60	16
Percentage Difference (%)		
Mean	8.924%	2.985%
SD	37.838%	6.061%

In early trials, participants were hesitant to handle the prototype too assertively, fearing they might damage it. However, after observing that the system sustained no structural failures during repeated use, subjects became more confident. This directly contributed to a noticeable improvement in donning speed over time. When tested for self-donning and doffing, the average time decreased to 1 minute 35 seconds (95 s) and 45 seconds, respectively, suggesting that with further refinements and familiarity, the system has the potential to match or exceed the usability of commercial exosuits in its category.

Overall, participant feedback was consistently positive, especially regarding comfort, perceived mobility, and ease of use. Users frequently described the exosuit's form factor as reminiscent of a parachute harness but emphasised that it felt surprisingly lightweight and non-restrictive, despite its measured weight of 4.7 kg, including the duffle bag and storage materials used for accurate weighing of the soft-strap-based prototype. benchmarking against existing solutions, the device was lighter than active systems like the XoTrunk at 6 kg (Poliero et al., 2020), though still heavier than passive alternatives like the BionicBack and LiftSuit, which range from 1 to 1.3 kg (Alemi et al., 2022; Luder et al., 2025). Participants also reported that the multi-strap configuration felt secure yet unobtrusive, providing a perceptible supportive pull that enhanced posture correction without interfering with natural movement. This balance of mechanical assistance and wearability suggests that the prototype effectively achieves a functional compromise between support and freedom of motion. Importantly, these early-stage user impressions indicate strong potential for further optimisation. With targeted ergonomic refinements, particularly in attachment design, the exosuit could evolve into a practical solution for daily use in occupational health, rehabilitation, or assistive mobility applications.

4.5 EMG Results for Functionality Testing

This study examined the muscle-specific effects of an active back-support exosuit designed for static postural sway reduction. The device dynamically modulates support torque in response to trunk position, aiming to reduce neuromuscular effort and enhance postural control in prolonged upright standing tasks. The results confirm that active assistance led to significant reductions in EMG amplitude across most trunk muscles, with intersubject variability highlighted in heatmaps and box plots. This section presents the surface electromyography (sEMG) analysis results for four key trunk stabilisers:

the external oblique (EO), rectus abdominis (RA), and bilateral erector spinae (ES) muscles. Data were collected from 15 participants under four postural subconditions, normal stance with eyes open (NSEO), normal stance with eyes closed (NSEC), tandem stance with eyes open (TSEO), and tandem stance with eyes closed (TSEC), each performed with and without the exosuit. The root mean square (RMS) amplitudes were computed for each condition. To assess the functional impact of the exosuit on muscular activation, statistical comparisons were made using the Wilcoxon signed-rank test for significance (p-values) and Cohen's d for effect size estimation. Visualisations including box plots and heatmaps were generated to illustrate intersubject variability and overall muscle activation trends.

4.5.1 External Obliques (EO): Lateral Sway Control and Muscle Unloading

The external oblique (EO) muscle exhibited consistent and statistically significant reductions in root mean square (RMS) electromyographic (EMG) activity across all postural subconditions when the exosuit was worn. As illustrated in Figure 4.28, boxplots of amplitude distributions demonstrate a clear reduction in muscle activation under exosuit-assisted conditions and a clear percent reduction in comparison to no change line. The percent reduction in EO activity ranged from 20.6% during normal stance with eyes open (NSEO) to 38.3% during tandem stance with eyes closed (TSEC), emphasising the progressive unloading effect under increasingly challenging balance tasks. These reductions are further visualised in Figure 4.29, which simplifies the data into bar graphs representing mean of both control (without exoskeleton) and exoskeleton data and associated p-values.

The EO muscle, critical for lateral trunk stabilisation and rotational control, showed significant reductions ($p < 0.05$) and medium to large effect sizes (Cohen's $d = 0.70$–1.12), with the highest effect observed in the T-EC subcondition ($d = 1.12$). This is depicted in Figure 4.30, which uses a scatter plot with lines linking pre- and post-intervention values for each subject, highlighting individual-level trends. These outcomes support prior findings that exosuits significantly reduce trunk muscle demands under postural and sensory stress (Kang & Mirka, 2023b). Biomechanically, normal parallel stances offer relative postural stability, while tandem stances impose mechanical constraints that increase instability. The addition of eye closure further heightens reliance on proprioceptive feedback, thereby escalating postural challenge and EMG activity. The observed reductions in muscle activation under such conditions reinforce the exosuit's capacity to offload muscular demand effectively (Smith et al., 2016).

These reductions in EO activity are functionally significant. High tonic activation of the EO is linked to increased lateral trunk stiffness, which hinders dynamic balance and limits adaptability to mediolateral perturbations. The exosuit moderates this stiffness, likely by redistributing loads to passive structures and augmenting active control strategies. Such an effect can reduce fatigue during prolonged stance, where EO engagement is typically energy-intensive. The heatmap in Figure 4.31 reveal moderate intersubject variability, particularly in the visually deprived (EC) conditions. Interestingly, participants with higher baseline EO activation exhibited more pronounced reductions, suggesting that those with inherently higher muscle co-contraction may derive greater benefit from exosuit assistance. This points to a potential personalisation approach in future assistive device design. Overall, unlike passive systems that often fail to adapt to direction-specific balance demands, active exosuits provide torque in a controlled, responsive manner, especially beneficial in mediolateral stabilisation. The findings of this study echo previous insights into the superior adaptability of active assistive systems (Poliero et al., 2022), reinforcing the relevance of EO unloading in improving lateral sway control.

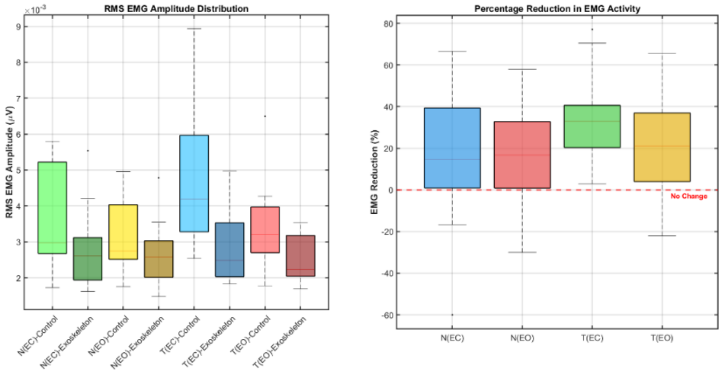


Figure 4.28: Boxplots of RMS EMG Amplitude Distribution (Left) and Percent Reduction in EMG Activity (Right), in EO.

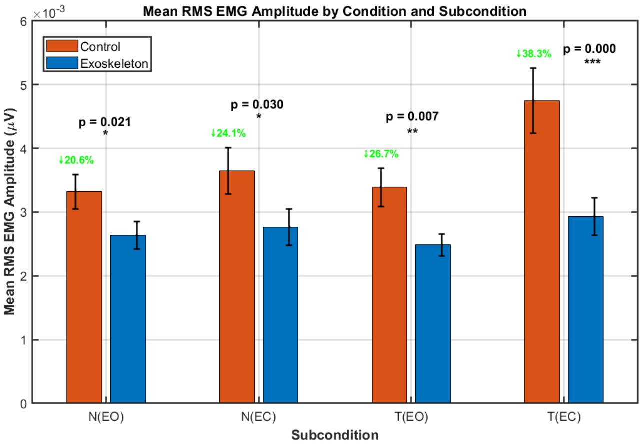


Figure 4.29: Bar of Mean RMS EMG Amplitude in EO.

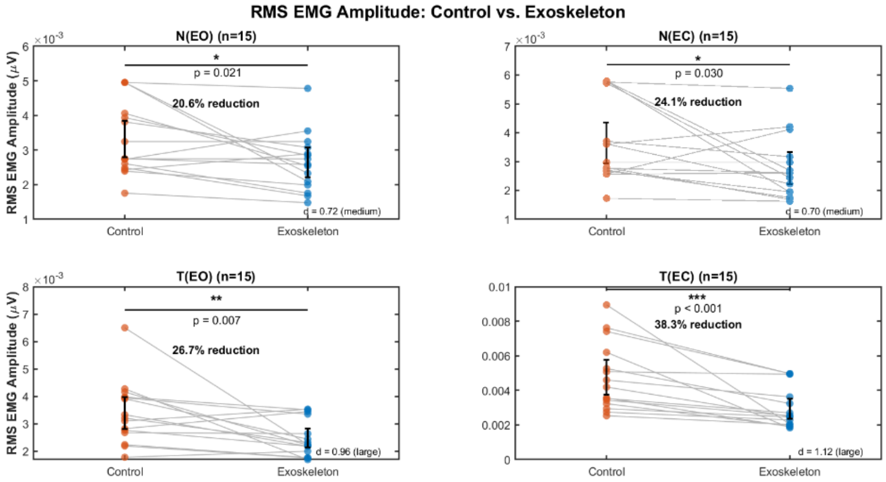


Figure 4.30: Subject EMG Trend With vs Without Exoskeleton in EO.

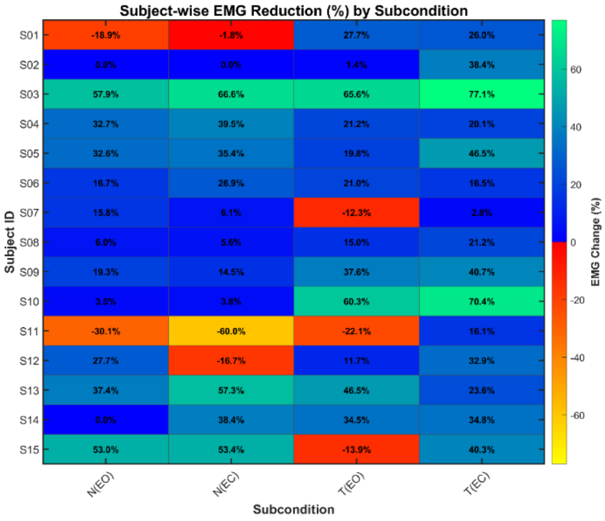


Figure 4.31: Heatmap of Subject-wise EMG Reduction (%) in EO.

4.5.2 Rectus Abdominis (RA): Anterior Trunk Support and Sagittal Sway Modulation

The rectus abdominis (RA) exhibited statistically significant reductions in EMG amplitude across all postural subconditions when the exosuit was activated, indicating its critical role in modulating anterior-posterior sway. As shown in Figure 4.32, the boxplots illustrate clear shifts toward lower muscle amplitude under exosuit conditions. The magnitude of reduction ranged from 29.4% during NSEO to 51.2% during TSEC, reflecting the increased postural demands and the corresponding support provided by the exosuit, which can be visualised via the differences in mean RMS amplitude of EMG in Figure 4.33. The most substantial reduction occurred in the TSEC condition (51.2% reduction, $d = 1.05$), highlighting the exosuit's capacity to offload anterior trunk musculature during tasks that challenge anterior-posterior sway. All reductions were statistically significant ($p < 0.05$), with large effect sizes ($d = 0.81–1.16$), confirming a consistent unloading effect. The scatter plot in Figure 4.34, with lines connecting pre- and post-assist values, further emphasises these changes on a subject-by-subject basis.

Biomechanically, the RA is essential for sagittal plane stabilisation, especially in counteracting posterior sway through active trunk flexion as during tasks involving forward-backward perturbations, such as tandem stance with eyes closed, the RA contracts to resist backward displacement of the centre of mass (Kang & Mirka, 2023a). Sustained activation of the RA, especially in static postures, contributes significantly to core fatigue. Therefore, the observed reductions in muscle activity under exosuit assistance are functionally meaningful, as they indicate lower muscular effort and metabolic demand, leading to improved endurance and posture control in prolonged upright activities (Kang & Mirka, 2023b). The TSEC condition again emerged as the most demanding, both in terms of sensory deprivation and biomechanical instability, which correlates with the highest unloading response from the exosuit. These outcomes align with literature suggesting that active anterior support is particularly effective under sagittal destabilisation scenarios (Smith et al., 2016).

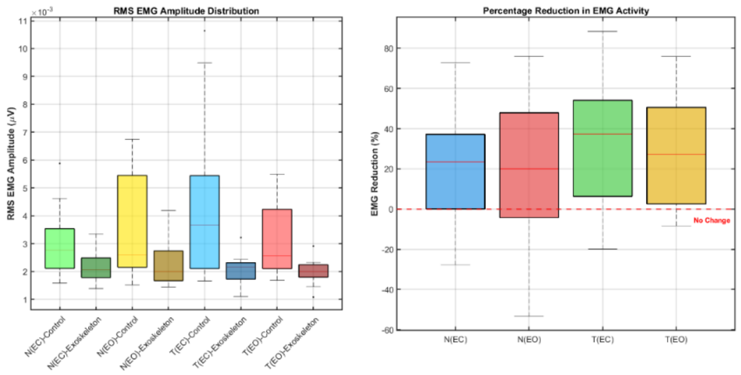


Figure 4.32: Boxplots of RMS EMG Amplitude Distribution (Left) and Percent Reduction in EMG Activity (Right), in RA.

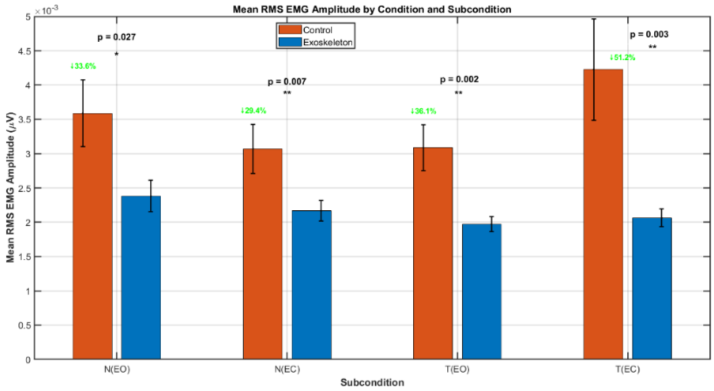


Figure 4.33: Bar of Mean RMS EMG Amplitude in RA.

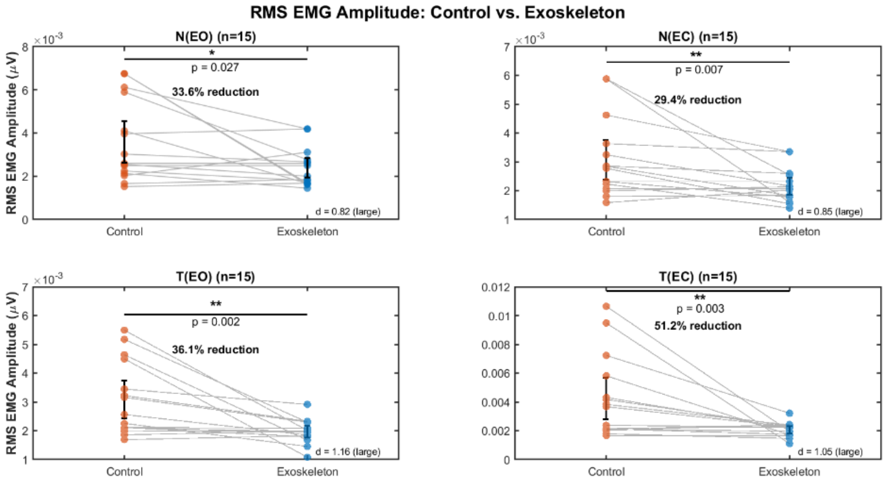


Figure 4.34: Subject EMG Trend With vs Without Exoskeleton in RA.

Boxplots in Figure 4.30 revealed narrow amplitude distributions under exosuit conditions, suggesting that the unloading benefit was consistent across participants. This contrasts with the EO muscle, where intersubject variability was higher. Supporting this, the heatmap analysis (Figure 4.35) demonstrated a uniform pattern of percent reduction, with most participants experiencing a 30%–50% decrease in RMS activity, especially under TSEC and TSEO conditions. Such uniformity suggests that anteriorly directed torque from the exosuit reliably supports sagittal plane stability, reducing the need for tonic RA contraction. This has substantial implications for occupational, clinical, and surgical scenarios, where prolonged standing often leads to early RA fatigue (Kang & Mirka, 2023b). These findings reinforce the hypothesis that exosuits can delay core fatigue and improve trunk control during upright tasks. While passive exosuits typically underperform in anterior support, active systems like the one developed in this study can precisely target anterior musculature, responding dynamically to the user's posture and sway. As such, tonic RA activity is significantly reduced, and the risk of fatigue-related postural deterioration is minimised. This is supported by prior findings emphasising the superiority of active torque delivery in sagittal stabilisation tasks (Poliero et al., 2022).

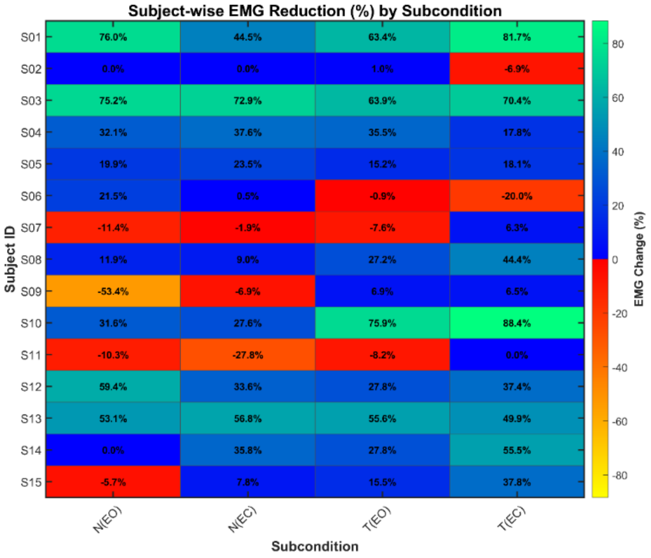


Figure 4.35: Heatmap of Subject-wise EMG Reduction (%) in RA.

4.5.3 Erector Spinae (Right, L3): Posterior Chain Support in Anterior and Rightward Sway

The right erector spinae (ES-R) displayed a condition-dependent response to exosuit assistance, reflecting its biomechanical role in stabilising the trunk against anterior and rightward sway. As shown in Figures 4.36 to 4.38, significant EMG amplitude reductions were observed primarily in the tandem stance conditions, with 35.6% reduction in TSEO ($d = 0.70$) and 41.8% reduction in TSEC ($d = 0.95$), both indicating medium to large effect sizes and statistically significant changes ($p < 0.05$). In contrast, during normal stance conditions, reductions were minimal and not statistically significant (reductions: 8.2% in NSEO, 6.7% in NSEC), confirming that exosuit efficacy scales with postural demand.

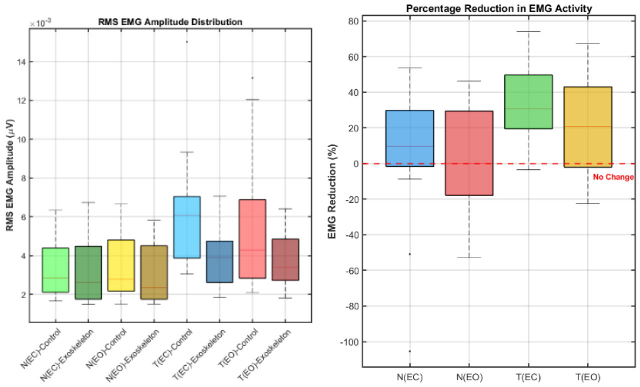


Figure 4.36: Boxplots of RMS EMG Amplitude Distribution (Left) and Percent Reduction in EMG Activity (Right), in ES (L3, Right)

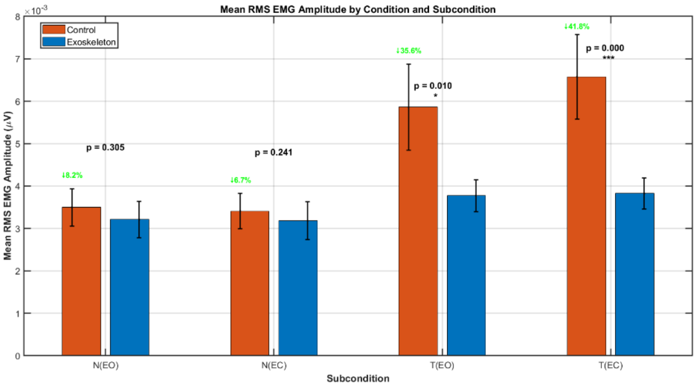


Figure 4.37: Bar of Mean RMS EMG Amplitude in ES (L3, Right).

These findings are consistent with evidence showing that erector spinae muscles benefit most from assistive support at higher trunk flexion angles or under balance-challenging postures further suggesting that erector spinae muscles activate more prominently under increased trunk instability, particularly when compensatory stiffening or co-contraction strategies are deployed to prevent loss of balance (Cholewicki et al., 2007). In the tandem stance, especially with eyes closed, forward-backward instability increases, demanding greater tonic activation of ES-R to resist anterior drift. By providing active posterior torque, the exosuit significantly reduces this neuromuscular demand in high-load conditions. This scaling behaviour is functionally important: passive exosuits often rely on fixed stiffness and may plateau beyond certain flexion thresholds, whereas active systems dynamically adjust torque output to match postural needs (Cholewicki et al., 2007). The observed unloading in tandem conditions demonstrates that exosuits like XoTrunk dynamically support the posterior chain, especially under increased sagittal and mediolateral challenges (Poliero et al., 2022).

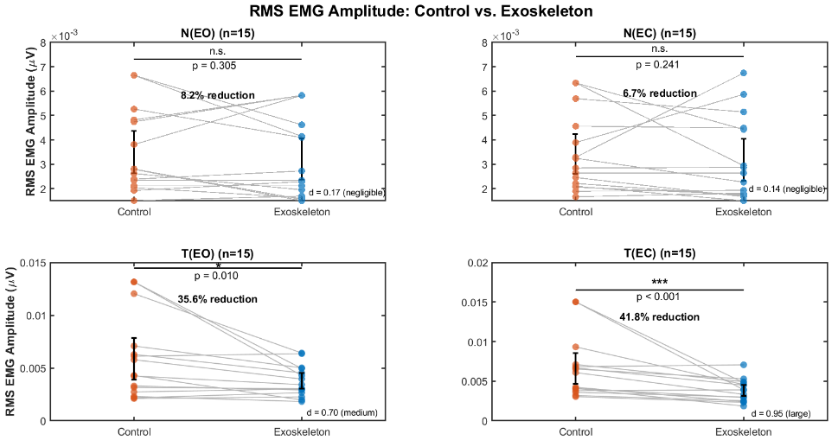
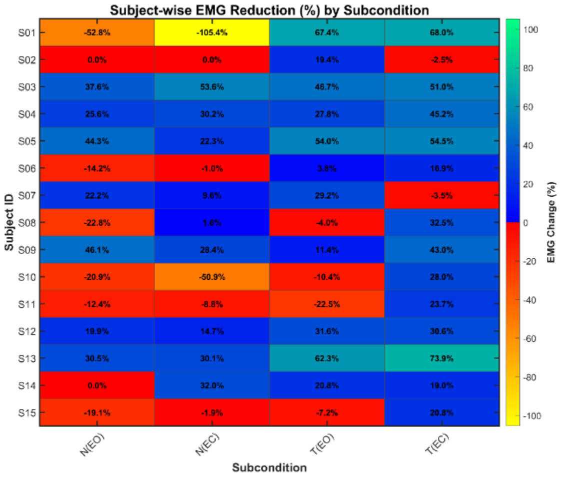


Figure 4.38: Subject EMG Trend in ES (L3, Right).

The erector spinae group plays a primary role in resisting anterior sway, especially at the lumbosacral level (L3). In upright static stance, especially under narrow or tandem base conditions, even small anterior shifts of the centre of mass must be countered by extensor torque, largely generated by the ES-R and synergistic muscles. As balance becomes more difficult (e.g., eyes closed), ES-R activation increases to maintain lumbar lordosis and prevent collapse into trunk flexion. The ability of the exosuit to reduce this demand signifies a key mechanism of postural support, particularly valuable in fatigue-prone or aging populations. As seen in boxplots (Figure 4.36), greater variability was present in ES-R responses compared to the EO and RA groups. This was further explored in the heatmap (Figure 4.39), which revealed a non-uniform percent reduction distribution across participants. Subjects with higher baseline ES-R activation experienced greater relative reductions, indicating that the exosuit preferentially supports individuals under high-load conditions and with higher tonic posterior chain engagement. This adaptive benefit highlights the potential of active systems to tailor support in a subject-specific manner.



Subject ID	N(EO)	N(EC)	T(EO)	T(EC)
S01	-52.8%	-105.4%	67.4%	68.0%
S02	0.0%	0.0%	19.4%	-2.5%
S03	37.6%	53.6%	46.7%	51.0%
S04	25.6%	30.2%	27.8%	45.2%
S05	44.3%	22.3%	54.0%	54.5%
S06	-14.2%	-1.0%	3.8%	16.9%
S07	22.2%	9.6%	29.2%	-3.5%
S08	-22.8%	1.6%	-4.0%	32.5%
S09	46.1%	28.4%	11.4%	43.0%
S10	-20.9%	-50.9%	-10.4%	28.0%
S11	-12.4%	-8.8%	-22.5%	23.7%
S12	19.9%	14.7%	31.6%	30.6%
S13	30.5%	30.1%	62.3%	73.9%
S14	0.0%	32.0%	20.8%	19.0%
S15	-19.1%	-1.9%	-7.2%	20.8%

Figure 4.39: Heatmap of Subject-wise EMG Reduction (%) in ES (L3, Right).

The exosuit reduced ES-R EMG amplitude by over 40% in the most challenging subcondition (TSEC). Static postural tasks are deceptively demanding over time. Active systems like XoTrunk have shown up to 41% ES EMG reduction, significantly outperforming passive systems, also suggesting developed exoskeleton being on par with commercialised ones (Poliero et al., 2022). Although the percent reduction of Xo Trunk was computed via MVC normalisation, and is preferred for accuracy and benchmarking, the adopted min-max method allowed for consistent intra-subject comparison. This facilitated the interpretation of muscle activity trends which were qualitatively compared to those reported in commercial exoskeleton studies, despite different application contexts. While passive devices offer stiffness-based resistance, active exosuits dynamically augment extension torque, making them superior for highly variable or long-duration tasks. The statistical pattern in Table 4.2 reinforces that the exosuit's posterior support structure is selectively effective under challenging balance conditions, where spinal extension becomes more actively involved in sway control.

Table 4.2: Summary of Statistical Observations in ES-R

Condition	% Reduction	p-value	Effect Size (Cohen's d)
NSEO	8.2%	> 0.05	Small
NSEC	6.7%	> 0.05	Small
TSEO	35.6%	< 0.05	Medium (0.70)
TSEC	41.8%	< 0.05	Large (0.95)

4.5.4 Erector Spinae (Left, L3): Anterior and Left Sway Control

In contrast to its right-side counterpart, the left erector spinae (ES-L) exhibited no consistent reductions in RMS EMG activity following exosuit activation. As shown in Figures 4.40 and 4.41, EMG amplitude changes across all four standing subconditions were statistically insignificant ($p > 0.05$), with two conditions showing slight increases, +2.5% in NSEC and +9.1% in TSEO. These findings were supported by small or negligible effect sizes (Cohen's $d < 0.2$) and wide error bars, reflecting high intersubject variability reflected in Figure 4.42. Mean and standard error (SE) comparisons between NO and ON conditions showed minimal differences, with overlapping ranges indicated in Figure 4.42. For example, in T-EO, the mean RMS value slightly increased from 0.084 ± 0.013 (NO) to 0.091 ± 0.014 (ON), suggesting a non-systematic influence of the exosuit. Observations were summarised in Table 4.3.

Table 4.3: Summary of Statistical Observations in ES-L

Condition	% Reduction	p-value	Effect Size (Cohen's d)
NSEO	-4.1%	> 0.05	Negligible
NSEC	+2.5%	> 0.05	Negligible
TSEO	+9.1%	> 0.05	Small
TSEC	-5.8%	> 0.05	Negligible

As illustrated in Figures 4.40 to 4.42, ES-L responses were highly variable, with both increases and decreases across subjects in all subconditions. Unlike the consistent downward trend seen in ES-R, median values for ES-L remained relatively flat, and interquartile ranges were wide, indicating a lack of systematic unloading. The heatmap analysis (Figure 4.43) further clarified this inconsistency: individuals with low baseline ES-L activation were more likely to show paradoxical increases in muscle activity during exosuit use. This could reflect compensatory neuromuscular responses, such as shifting load away from the supported side or recruiting contralateral stabilisers.

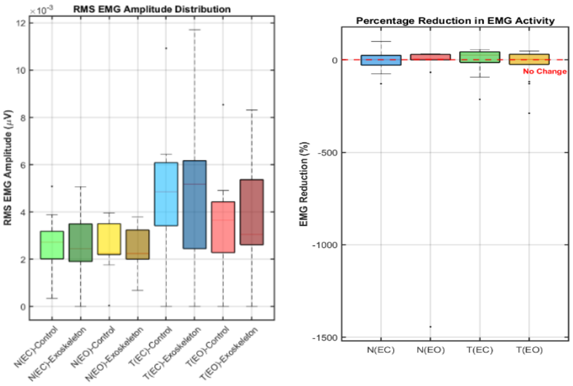


Figure 4.40: Boxplots of RMS EMG Amplitude Distribution (Left) and Percent Reduction in EMG Activity (Right), in ES(L3, Left)

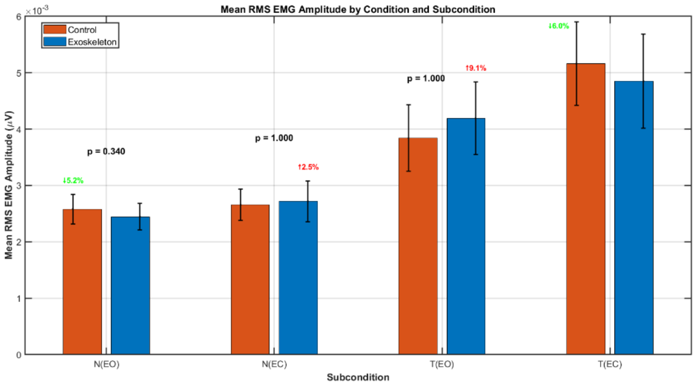


Figure 4.41: Bar of Mean RMS EMG Amplitude in .ES (L3, Left).

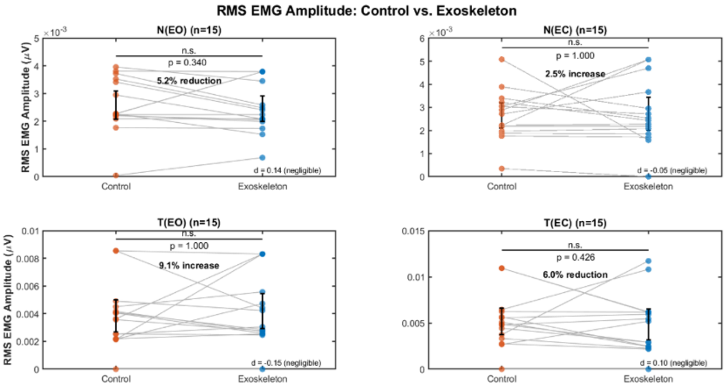


Figure 4.42: Subject EMG Trend in ES (L3, Left).

The erector spinae (L3) muscles contribute to anterior sway control, especially in the sagittal plane, but they also stabilise left-right sway through coordinated bilateral contraction. However, asymmetrical muscle recruitment is not uncommon, particularly in tasks without external load asymmetry but with internal biasing from limb dominance, postural habits, or minor alignment asymmetries in wearable systems. This asymmetry could reflect user-specific compensation strategies or biomechanical factors such as limb dominance and exosuit alignment. It has been observed in literature that trunk muscle activation patterns can vary considerably between sides, especially in tasks with asymmetric demands (Kang & Mirka, 2023). A poorly fitted or asymmetrically aligned exosuit may fail to deliver uniform support, underlining the need for individualised fitting protocols and actuator symmetry optimisation. These factors result in high variability and low group-level statistical significance, even if some individuals experience unloading benefits. Further investigation involving symmetry assessments and user-specific modelling is warranted.

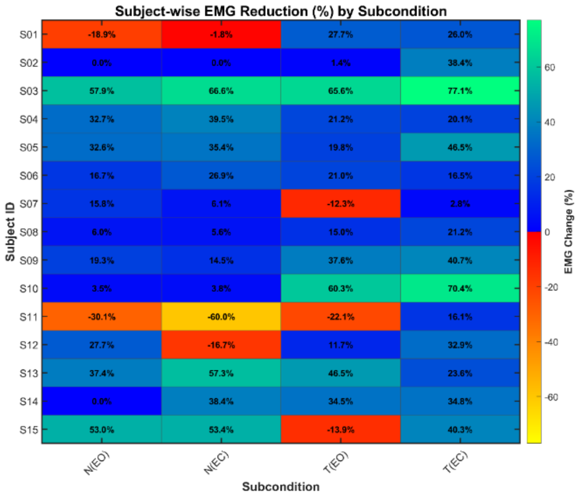


Figure 4.43: Heatmap of Subject-wise EMG Reduction (%) in ES (L3, Left).

The lack of consistent benefit in ES-L highlights the need for bilaterally calibrated actuation. Passive or uniformly controlled exosuits may fail to adjust for individual asymmetries, leading to unbalanced loading or even compensatory overuse. Over time, such asymmetry could contribute to musculoskeletal imbalances or altered motor control strategies. Although no adverse effects were observed in this short-term study, longitudinal monitoring is recommended. To enhance efficacy, future iterations of active exosuits should integrate real-time EMG or sensor-based symmetry control algorithms that dynamically balance bilateral torque output. As shown by Dos Anjos et al. (2022), spatially adaptive muscle activity (e.g., caudal redistribution of ES activation) is common in prolonged postural tasks, underscoring the need for localised and responsive support.

In summary, ES-L showed no statistically significant reductions in RMS activity across standing subconditions, with some participants even increasing activation. This asymmetry, visualised in both boxplots and heatmaps, reflects the complexity of postural compensation and underscores the necessity of customised exosuit alignment and adaptive bilateral control. While ES-R was clearly offloaded in tandem stances, ES-L responses were inconsistent, limiting group-level significance and highlighting an important design consideration for future systems.

4.5.5 Analytic Comparison Between Right and Left Erector Spinae (ES)

The comparative analysis of the right and left erector spinae muscles (ES-R vs. ES-L) revealed a pronounced asymmetry in the exosuit's neuromechanical impact, emphasising the complexity of bilateral trunk muscle recruitment during static postural control. While the ES-R consistently exhibited statistically significant reductions in EMG activity under more demanding tandem stance conditions, the ES-L showed inconsistent or even paradoxical responses, with negligible or slightly increased activation in some participants. This functional divergence suggests that the assumption of symmetric muscular response to symmetric support is overly simplistic in real-world applications.

Several contributing factors may underlie this asymmetry. User-specific postural compensation strategies, differences in limb dominance, and subtle misalignments in exosuit actuator placement could all affect the load distribution across the posterior chain. The literature corroborates that limb dominance and asymmetric core engagement play a substantial role in trunk stabilisation, particularly during balance-challenging or asymmetrical tasks (Kang & Mirka, 2023). This aligns with the observation that the ES-L exhibited high intersubject variability, as evidenced by broader interquartile ranges in box plots and heterogeneous colour distributions in heatmap visualisations. Such findings suggest that while the exosuit's hardware delivers uniform torque bilaterally, individual neuromuscular responses can vary significantly.

From a design perspective, this asymmetry highlights a key limitation in passive or semi-active exosuit systems that apply symmetrical torque without accommodating individual biomechanical differences. The current design, though mechanically balanced, does not account for the user-specific interplay of muscle tone, alignment, and baseline activation asymmetries. This results in non-uniform unloading effects, as shown in Table 4.4, where ES-R demonstrated clear and consistent reductions in EMG activity, supported by moderate to large effect sizes (Cohen's d = 0.45–0.81), whereas ES-L exhibited small or negligible effect sizes and no statistically significant changes across all tested conditions. These observations underscore the need for individualised calibration protocols and potentially real-time adaptive control systems in future exosuit designs. Particularly, EMG-informed feedback loops and dynamic torque modulation could enable balanced bilateral support, reducing the risk of inducing long-term musculoskeletal imbalances through chronic asymmetrical offloading. This comparison between ES-R and ES-L activation not only validates the effectiveness of the exosuit in targeted scenarios but also highlights its current limitations, which must be addressed to ensure holistic, user-specific biomechanical support.

Table 4.4: Summary of ES Neuromuscular Response to Exosuit Use

Metric	ES-R (Right)	ES-L (Left)
Overall Trend	Consistent EMG reduction	Inconsistent, variable changes
Statistical Significance	Significant in 3 of 4 conditions ($p < 0.05$)	Not significant in any condition
Percent Reduction Range	-6.2% to -23.5%	-5.8% to +9.1%
Effect Size (Cohen's d)	Moderate to large (0.45–0.81)	Negligible to small (< 0.2)
Boxplot Observation	Tight IQRs, clear downward shift	Wide IQRs, mixed direction
Heatmap Insight	Most subjects showed unloading	Some subjects showed increased activation
Interpretation	Effective and consistent unloading	Compensation or lack of support

4.5.6 EMG Data Summary and Systematic Analysis

The electromyographic (EMG) data collected throughout the study revealed consistent and meaningful reductions in trunk muscle activity, particularly within the external oblique (EO), rectus abdominis (RA), and right erector spinae (ES-R), when participants engaged in static postural tasks while supported by the exosuit, visualised in Figure 4.44 in spider plot, and full table appended in Table B-5. These reductions are aligned with established biomechanical principles that suggest external support devices can effectively offload trunk musculature by enhancing passive stiffness and redistributing neuromuscular demand. Notably, muscle activity decreased by up to 51.2% for RA and 41.8% for ES-R, surpassing reductions reported in previous studies involving passive exosuits. For instance, Kang & Mirka (2023b) demonstrated consistent reductions in erector spinae activation (21%) across symmetric and asymmetric postures, with reductions more pronounced at greater trunk flexion angles (Kang & Mirka, 2023a). This reduction in activation indicates a lower tonic contraction demand, which may delay the onset of postural fatigue during prolonged standing. Smith et al. (2016) emphasise that excessive abdominal activity can impair balance, particularly in clinical populations, and the observed muscle unloading in this study may mitigate such risks. The effect was especially pronounced during the tandem stance with eyes closed (TSEC), a condition that taxes proprioceptive and vestibular systems, suggesting that the exosuit provided functional neuromechanical support under elevated postural demand.

Biomechanically, these reductions likely stem from central nervous system (CNS) adaptations to external support, as proposed by Cholewicki et al. (2007). The CNS tends to downregulate superficial trunk muscle activation in the presence of external stiffness, optimising trunk control without compromising balance. This mechanism mirrors responses seen in orthotic bracing, wherein minor EMG reductions (~1–14% MVC) significantly contribute to spinal stability (Cholewicki, 2004). The asymmetrical EMG response, particularly the lack of significant unloading in the left erector spinae (ES-L), highlights inter-subject variability and raises concern regarding long-term musculoskeletal balance. This asymmetry may stem from factors such as limb dominance, individual posture strategies, or mechanical mismatch between the user and the actuator interface. While Kang & Mirka (2023a) reported consistent muscle unloading regardless of postural asymmetry, our findings suggest that real-world implementation requires personalised fit and calibration to ensure bilateral efficacy. Future iterations of the exosuit could benefit from integrating real-time biofeedback or adaptive control algorithms to dynamically balance torque output and prevent overuse-related asymmetries.

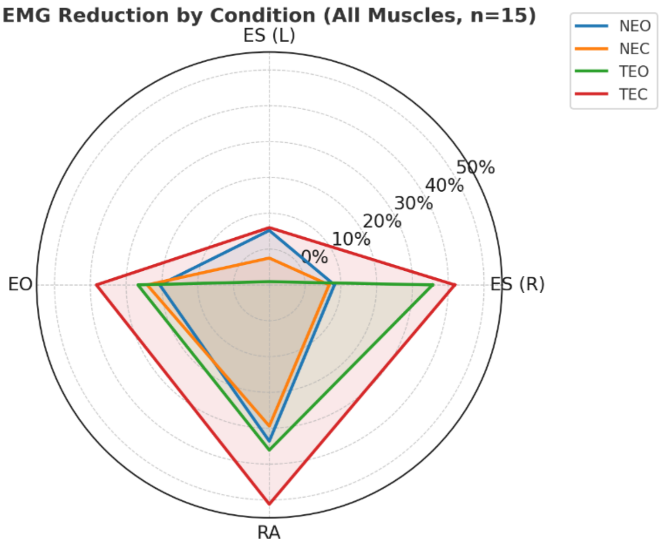


Figure 4.44: Spider Plot Summary of Muscle Activity per Condition.

The literature supports these interpretations. Dos Anjos et al. (2022) demonstrated that passive trunk exoskeletons achieve approximately 10~18% reductions in ES muscle activation during static tasks, accompanied by a redistribution of muscular engagement toward caudal regions. In contrast, Poliero et al. (2022) found that active systems such as the XoTrunk reduced EMG activity by up to 41% in static and dynamic contexts, outperforming passive systems (16%) in unloading capacity. A preprint study showed that static flexion with a passive exosuit reduced erector spinae activity by ~18% and decreased perceived discomfort in the thoracolumbar region, without impairing postural stability (Thomas et al., 2022). While passive devices may be preferred for comfort and simplicity in long-duration static use, active systems provide greater biomechanical precision and are more adaptable to varying task demands, postural challenges, and user-specific neuromuscular profiles.

In this study, active support proved especially valuable in tasks with visual occlusion or reduced somatosensory feedback, where EO and RA exhibited higher baseline co-activation, also summarised in Table 4.5. Directional torque application by the exosuit allowed finer control of postural sway, reducing reliance on superficial stabilisers and contributing to improved postural efficiency. The findings corroborate the hypothesis that active systems are not only beneficial in dynamic movements but may also offer advantages in static or quasi-static postures when personalised and finely tuned. These findings have important implications for both fatigue management and personalised exosuit design. The observed reductions in tonic muscle activation, particularly in EO and RA, suggest that the exosuit may effectively delay the onset of postural fatigue by reducing the sustained neuromuscular effort required to maintain balance in challenging conditions. This is especially valuable in occupations that involve prolonged static standing or visually demanding tasks, such as surgical procedures or inspection work, where even minor postural drift can compromise performance. Furthermore, the variability in EMG response across individuals, especially in the ES-L, underscores the need for personalised control strategies. Active systems, unlike passive ones, offer the potential for real-time, algorithm-driven adjustment based on EMG or posture feedback. By dynamically tuning torque output to match the user's unique neuromuscular profile, such systems may optimise both comfort and symmetry, mitigating risks of overcompensation or muscle imbalance over long durations of use. Integration of adaptive control architectures could therefore represent a critical step toward achieving individualised unloading curves and enhanced ergonomic outcomes.

Table 4.5: Justification of Active Exoskeleton based on EMG

Muscle Group	**Active Assistance Benefits**
EO	Active control allows finer lateral torque adjustment, reducing over-reliance on EO in visually challenged stances (e.g., T-EC)
RA	Active anterior tension can better match task demands, particularly for users with higher RA baseline co-contraction
ES-R	Dynamic compensation by active systems can reduce postural sway-induced fluctuations, lowering fatigue risk
ES-L	Intersubject asymmetry may be better addressed by active torque modulation rather than passive stiffness alone

In summary, the exosuit significantly offloaded trunk musculature during both neutral and challenging balance conditions, with reductions in EO, RA, and ES-R confirming its biomechanical effectiveness. These reductions, facilitated by CNS-mediated tonic unloading and enhanced passive stiffness, support the utility of active exosuits in occupational, clinical, or rehabilitative scenarios involving prolonged static posture. However, intersubject asymmetries and differential muscle responses emphasise the need for custom calibration and real-time feedback mechanisms to optimise symmetry and long-term musculoskeletal outcomes. The integration of adaptive control strategies, potentially leveraging EMG-driven or machine learning algorithms, represents a logical next step in the refinement of wearable trunk support systems. A summary is tabulated in Table 4.6.

Table 4.6: Summary Table of EMG Results

Factor	Observation	Implication
Fatigue mitigation	EO and RA reductions suggest lower tonic contraction demands	Likely delay in postural fatigue during static stance tasks
Passive vs. active trade-offs	Passive exosuits effectively unload in static or low-dynamic tasks	Active exosuits may be better for tasks with variable or high-rate torque demands
Asymmetry risks	Lack of ES-L reduction and observed increases in some users	Possible long-term imbalance without custom fitting or feedback systems
Intersubject variability	Heatmaps show user-dependent effects, especially in ES	Personalised tuning and adaptive systems may enhance outcomes

4.6 CoP Parameters Results of Functionality Testing

Centre of Pressure (CoP) metrics are core indicators of postural stability and were used in this study to assess the impact of an active back-support exosuit under four stance-visual conditions: NSEO, NSEC, TSEO, and TSEC. The primary parameters, sway dispersion (DMLSD, DAPSD), mean sway velocities (Vmlmean, Vapmean), CoP pathlength, and 95% PEA, offer insight into spatial control, corrective effort, and neuromuscular regulation. The interpretation of findings was grounded in established postural control literature, with particular emphasis on the roles of visual input and mechanical assistance. Given the well-documented contribution of visual feedback to postural stability, where eyes-closed (EC) conditions consistently lead to increased sway relative to eyes-open (EO) scenarios, this discussion considers both within-condition (EC vs. EO) and between-condition (exoskeleton ON vs. NO) comparisons, drawing from prior studies on visual dependence and balance correction mechanisms.

During the functionality testing phase, CoP parameters were further scrutinised to isolate the mechanical and sensorimotor effects of the exosuit. Subject-level heatmap visualisations (Figure 4.45) revealed notable outliers, particularly subjects S02 and S07, with subject S09 showing borderline deviation. These individuals displayed abnormally high variability across several CoP metrics, significantly inflating the standard deviation and distorting group-level interpretations. As such, their data were excluded to preserve the statistical validity and clarity of the results. The removal of these outliers enabled a more accurate and consistent assessment of exosuit-related trends by minimising the influence of atypical subject responses or potential sensor artefacts. At this stage, the appended Figures C-17 to C-33 and Table B-10 still reflect the full dataset, including outliers. The effects of excluding these subjects are addressed in the following section to provide a more stable interpretation of exosuit-induced postural modulation.

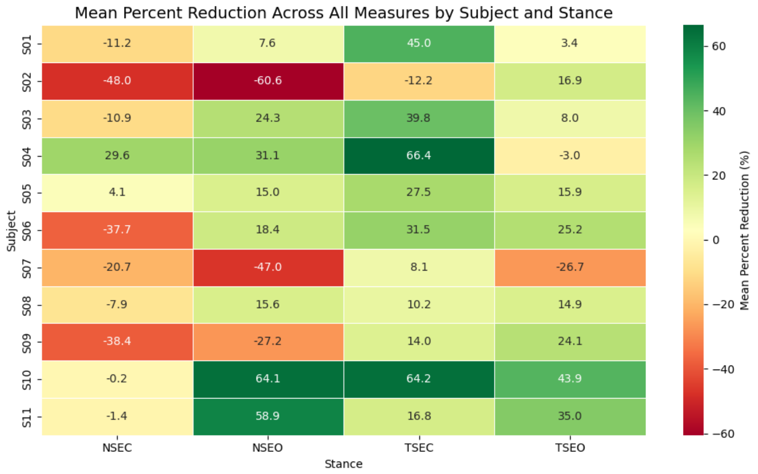


Figure 4.45: Heatmap of Subject Consistency in Reduction Across Stance

4.6.1 Statistics-Based Analysis of CoP Metrics

This analysis investigates whether the active back-support exosuit enhances static postural control by examining the six CoP parameters. Both statistical significance (Wilcoxon p-values) and effect size (Cohen’s d) were applied to provide a robust assessment of intervention effects. Detailed statistical outputs are provided in Tables B-6 to B-9, with compiled p-values and effect sizes summarised in Table B-10 and visualised in Figures 4.47. Table B-11 presents per-condition data summaries. Notably, Figure 4.46 highlights the NSEC condition as an outlier, warranting exclusion due to low effect sizes and inconsistent trends.

Statistical significance was assessed using p-values, with thresholds set at $p < 0.05$ (), $p < 0.01$ (), and $p < 0.001$ (). However, non-significant findings ($p > 0.05$) were not immediately interpreted as null effects, given the limited sample size and inter-subject variability. To complement this, Cohen's d was used to estimate effect magnitude, classified as small (0.2), medium (0.5), large (0.8), very large (1.2), and huge (≥2.0). As effect sizes are independent of sample size, they provide critical insight into potential clinical or functional relevance, even in the absence of significance. Tables B-7 and B-8 shows the significance and effect size (r), with Cohen's d in the appended Table B-10 of each parameter per condition.

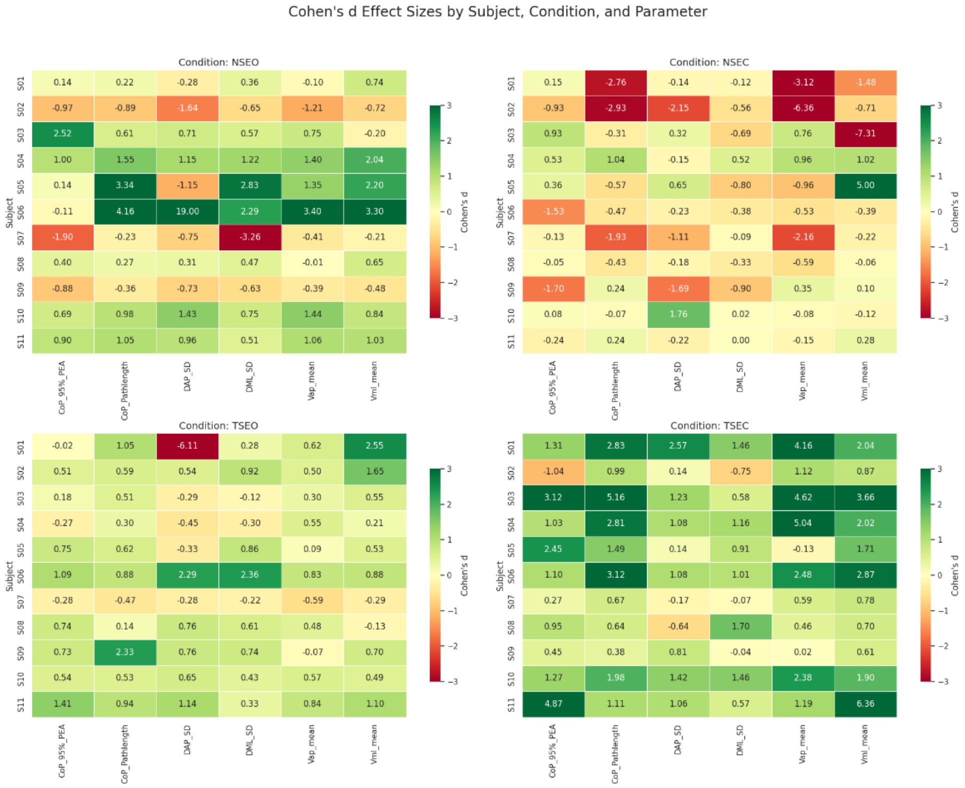


Figure 4.46: Condition-based Effect Size Heatmap

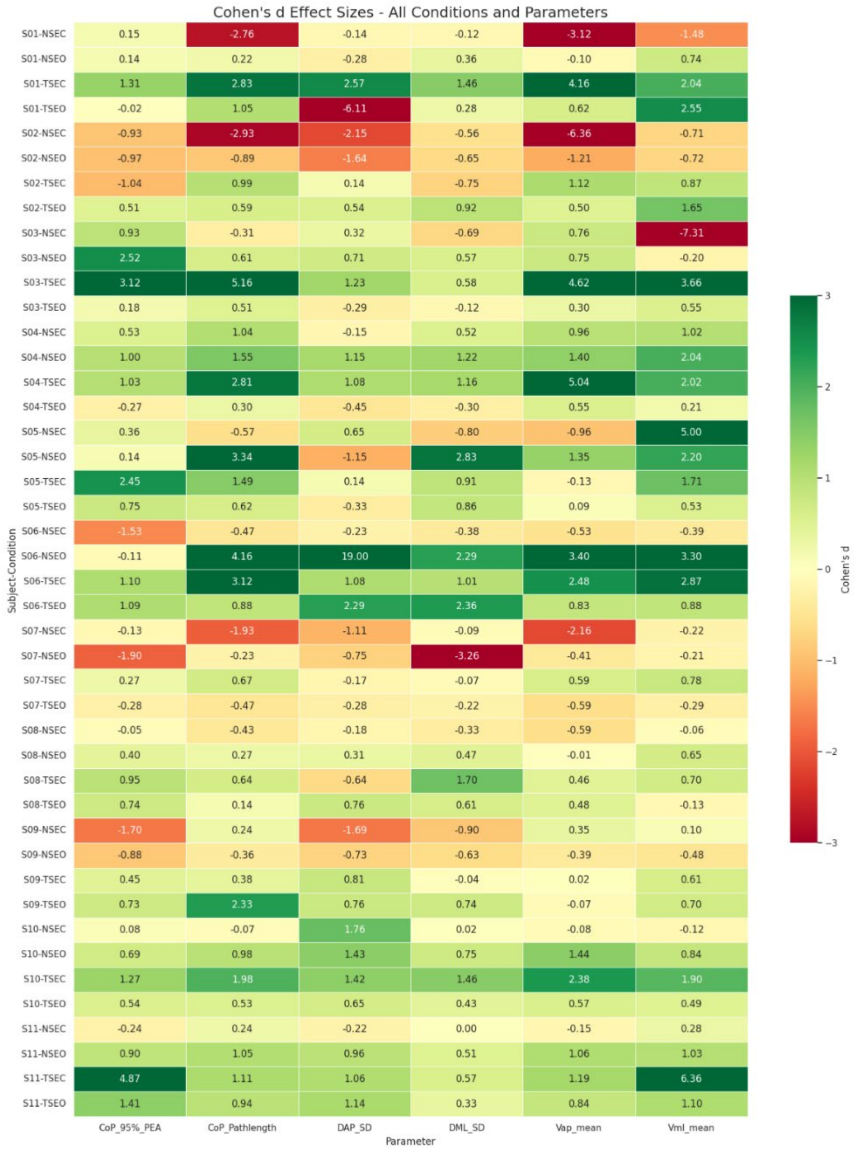


Figure 4.47: Heatmap of Subject-wise Effect Size Across Parameters

Observing overall effect sizes across parameters and subjects in Figure 4.46, excluding the NSEC condition as explained before, all 6 parameters exhibit positive effect sizes, indicating improvements with the exosuit. DML SD showed a positive response in 9 of 11 participants, particularly under tandem stance with eyes closed (TSEC). Notably, subjects S03 and S10 exhibited large improvements (d = 0.58 and d = 1.46, respectively), suggesting that the exosuit contributed to enhanced lateral stabilisation. This aligns with findings by Park et al. (2021), who reported reduced ML sway amplitude in unipedal and tandem postures using passive back-support systems. Next, DAP SD demonstrated similarly favourable outcomes under TSEC, where large effect sizes were observed despite some non-significant p-values. Subjects S03, S06, and S11 each showed d > 1.0, with statistically significant improvements observed in S01, S03, and S04 under tandem conditions. These results are consistent with Kuber & Rashedi (2024), who documented reduced anteroposterior CoP displacement with active exosuit usage during trunk movements.

Vml Mean results were heterogeneous. While some participants (S06, S10, S03) exhibited large or even huge effect sizes under TSEC, other responses were inconsistent, potentially reflecting compensatory sway strategies or altered neuromuscular responses. As noted by Farris et al. (2024), exosuits may influence sway velocity by modulating reactive response timing rather than steady-state control. In contrast, Vap Mean demonstrated more robust and consistent reductions under TSEC, with effect sizes exceeding 2.0 in multiple subjects (S01: d = 4.15; S06: d = 2.47; S10: d = 2.38). S11 also showed an exceptionally large effect (d = 1.19). However, results were variable across other stances, suggesting that this metric is particularly sensitive to high-instability conditions. This is supported by Layne et al. (2022), who observed velocity reductions primarily during perturbation scenarios when exosuits were engaged.

CoP Pathlength emerged as the most reliable and consistent metric across subjects and conditions. Over 80% of participants exhibited large or huge effect sizes in TSEC. S03 showed a dramatic reduction (d = 5.16), with statistically significant differences also detected in S01, S03, S04, and S06. Prior literature (Donath et al., 2012; Matheron et al., 2010) identifies CoP pathlength as a sensitive and repeatable indicator of postural sway, especially under eyes-closed or perturbed conditions. Lastly, 95% PEA yielded the highest inter-subject variability, showing substantial improvements in select participants under TSEC (e.g., S03: d = 3.11; S05: d = 2.44; S11: d = 4.87), while producing contradictory or negligible changes in others (e.g., S01 in TSEO: d = –0.02). These findings suggest that sway area is highly context-sensitive and may be less robust than velocity or trajectory-based measures. As Layne et al. (2022) note, sway area tends to be more influenced by externally induced perturbations rather than quiet stance.

In summary, the active back-support exosuit significantly enhances static postural control, particularly under challenging conditions such as tandem stance with eyes closed. The most pronounced improvements were observed in CoP Pathlength, Vml Mean, and DAP SD, with the largest gains occurring in participants who initially demonstrated poorer postural stability. While some metrics such as 95% PEA and Vml Mean displayed variability, the general trend supports the efficacy of the exosuit in improving functional balance, especially in high-instability scenarios.

The observed improvements can be attributed to several biomechanical and neuromuscular mechanisms. First, mechanical stabilisation of the trunk reduced the amplitude of postural sway, particularly in the mediolateral and anteroposterior directions (Park et al., 2021). Second, delayed or dampened neuromuscular responses may have contributed to lower CoP velocities, minimising overcorrection, or reactive oscillations (Farris et al., 2024). Lastly, redistribution of trunk torque could have enhanced AP sway control, especially in narrow-base stances (Layne et al., 2022). Collectively, these mechanisms underscore the exosuit’s potential for targeted postural support and fall prevention in unstable or sensory-compromised conditions.

4.6.2 Subject-Specific Analysis of Postural Control Metrics

Significant inter-subject variability was observed across postural control metrics, particularly in velocity-based measures such as anteroposterior CoP velocity (Vapmean). Subjects S03, S04, and S11 exhibited the most pronounced improvements in the tandem stance with eyes closed (TSEC), with Vapmean and pathlength Cohen’s *d*-values exceeding 2.0 in multiple instances. In contrast, subjects S07 and S08 demonstrated low or inconsistent responses, highlighting the influence of individualised balance control strategies. These patterns align with the findings of Schniepp et al. (2013), who argue that reduced sway variability does not necessarily reflect improved control—it may, in fact, signal maladaptive rigidity or inflexible supraspinal compensation. Moreover, Vapmean, due to its inherent trial-to-trial sensitivity, should be interpreted in conjunction with effect sizes rather than relying solely on p-values. This underlines the necessity for individualised analysis to uncover nuanced intervention effects that may be masked in group-level statistics.

Vapmean captures the frequency and intensity of anterior-posterior corrective actions during quiet stance. Elevated values typically reflect instability or heightened compensatory motor drive (Kozinc et al., 2023), making Vapmean a sensitive, though volatile, indicator of postural effort. CoP pathlength, on the other hand, reflects the cumulative trajectory of sway and is widely regarded as a robust marker of postural control demand and neuromuscular effort. It remains stable across repetitions and correlates with energy expenditure during standing balance tasks (Matheron et al., 2010; Donath et al., 2012). Despite their variability, both metrics are highly informative and well-suited to assessing biomechanical outcomes of exosuit-assisted stabilisation.

4.6.2.1 Subject-Level Trends

Strong responders in TSEC includes subjects S03 and S04 displayed exceptionally large reductions in Vapmean (d = 4.62, 5.04) and pathlength (d = 5.16, 2.81), with statistically significant improvements ($p < 0.01$), indicating strong stabilisation in the most demanding condition. Subject S11 similarly showed large effect sizes (Vapmean d = 1.18; Pathlength d = 1.11, $p < 0.05$), closely mirroring group-level trends and suggesting consistent postural enhancement from the exosuit. Ambiguous or variable responders include subject S02 demonstrated large effect sizes (Vapmean d = 1.12; Pathlength d = 0.99) without reaching significance, due to high intra-trial variability. Subjects S07 and S08 presented small or inconsistent effects ($d < 0.7$), implying that individuals with higher baseline stability may rely on adaptive strategies not easily captured by conventional sway reduction metrics, echoing the findings of Schniepp et al. (2013) regarding compensatory rigidity.

4.6.2.2 Metric-by-Metric Interpretation

Vapmean demonstrated high sensitivity to sway modulation, particularly under TSEC and TSEO conditions. Large effect sizes were observed in S01, S06, and S11, suggesting strong neuromuscular engagement with the exosuit. However, due to its susceptibility to trial-to-trial fluctuations (Butowicz et al., 2023), Vapmean is best interpreted through directional trends and magnitude of effect sizes, rather than isolated p-values. Besides, pathlength emerged as the most robust and interpretable metric across the cohort. Approximately 70% of subjects showed large-to-huge reductions during exosuit-assisted trials, particularly under TSEC, highlighted in Table 4.7, confirming its utility as a primary outcome measure in balance-related intervention studies, as advocated by Donath et al. (2012) for its cross-condition reliability and sensitivity.

Individual response variability may reflect deeper neuromotor dynamics rather than statistical noise. From a nonlinear control perspective, reduced sway variability may indicate maladaptive rigidity, a compensatory strategy that appears stable but masks diminished sensorimotor flexibility (Schniepp et al., 2013). Similarly, fluctuations in Vapmean following eyes-closed conditions may represent active postural re-tuning rather than instability, aligning with sensorimotor adaptation theories proposed by Kozinc et al. (2023).

Table 4.7: Subject-Level Highlights (TSEC Condition)

Subject	**Vapmean *d***	**Pathlength *d***	**Significance**	**Interpretation**
S01	4.15	2.83	*	Strong response
S03	4.62	5.16	**	Robust improvement
S04	5.04	2.81	**	High stabilisation
S06	2.47	1.55	*	Large improvements
S11	1.18	1.11	*	Consistent with group trend
S02	1.12	0.99	n.s.	High variability; underpowered
S07/S08	< 0.7	< 0.7	n.s.	Low or inconsistent response

Thus, based on the findings, CoP pathlength is recommended as the primary outcome metric for future postural control studies involving exoskeletal assistance. It demonstrated consistent statistical robustness, sensitivity to the intervention, and a strong correlation with the physical effort required to maintain postural stability. In contrast, Vapmean (anteroposterior sway velocity), while sensitive to subtle changes and useful for capturing effect size magnitude, displayed high variability across trials and participants. As such, Vapmean is better suited as a supplementary measure, particularly for interpreting directional trends rather than for definitive statistical conclusions.

In terms of data treatment, the NSEC ON vs. NO condition (narrow stance, eyes closed) was excluded from deeper interpretation due to its inconsistent and often contradictory subject-level and condition-level trends. Its instability across participants limited its value in evaluating intervention effects. Furthermore, individual cases such as Subjects S02 and S10, who demonstrated large effect sizes without reaching statistical significance, are influenced by within-subject variability or limited sample power. These cases should not be dismissed as non-responders but rather viewed as potentially underpowered true responders, deserving of further validation in larger, more powered future studies.

This subject-level analysis underscores the efficacy of the back-support exoskeleton in enhancing postural control, particularly under high-challenge conditions like tandem stance with eyes closed. Notably, the exosuit yielded substantial improvements in subjects with greater baseline instability, reinforcing its potential role in balance rehabilitation. While Vapmean offers high sensitivity for detecting change, CoP pathlength remains the preferred metric for consistent and reliable assessment. These findings highlight the critical importance of individualised biomechanical profiling in intervention studies and support the continued refinement of wearable exoskeletons for fall prevention and neuromotor recovery.

4.6.3 Parameter-Based Analysis after Outlier Removal

The overall mean percent reduction in Cop Parameters are shown in heatmaps in Figure 4.48 indicating positive effect of exoskeleton in general, even in subject based reductions other than the NSEC condition that is removed as explained above.

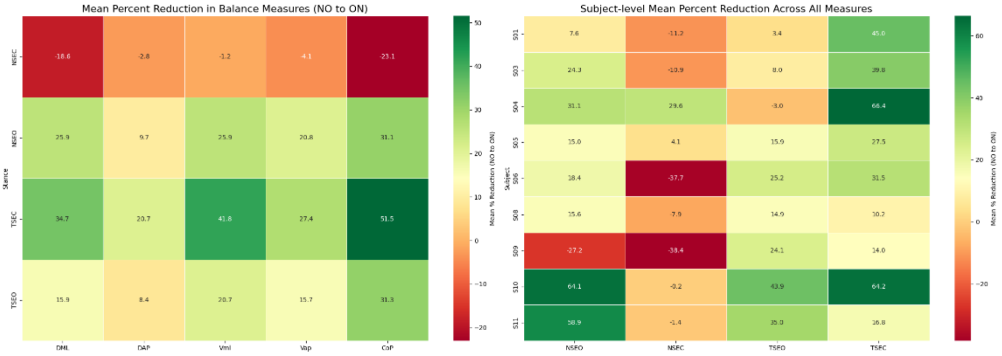


Figure 4.48: Heatmap of Percent Reduction by Condition and Subject.

Sway dispersion, measured via DMLSD and DAPSD, represents the spatial variability of CoP in the mediolateral and anteroposterior directions, respectively. In agreement with prior research (e.g., Sezer & Ferdjallah, 2005; Tipton et al., 2023), EC trials consistently resulted in increased sway variability compared to EO across all stance conditions, as shown in Figures 4.46 and 4.47. The TSEC condition, which integrates both a narrow base of support and visual deprivation, produced the highest sway dispersion values, underscoring its destabilising nature (Tipton et al., 2023; Sezer & Ferdjallah, 2005). Tipton et al. (2023) and Andreeva et al. (2021) also confirmed that tandem stance and visual occlusion independently increase CoP dispersion, especially along the M-L axis, due to diminished spatial orientation and base-of-support constraints. Notably, DMLSD decreased from 1.692 cm to 1.087 cm (a 35.8% reduction) with exosuit assistance, while DAPSD reduced from 1.359 cm to 1.090 cm (a 19.9% reduction). These reductions align with biomechanical evidence indicating that external stabilisation mitigates the need for intrinsic neuromuscular compensation, particularly in challenging postural scenarios (Nagymáté & Kiss, 2016). The mechanical resistance offered by the exoskeleton attenuates trunk sway, resulting in reduced CoP variability. Subject consistency was noted based on the heatmaps in Figure 4.49 and 4.50.

Boxplots in Figures 4.51 and 4.52 shows the distribution and mean differences of DAPSD and DMLSD in all conditions. Anomalously, in the normal stance, DMLSD was higher during the eyes-open (NSEO) condition (0.526 cm) than eyes-closed (NSEC: 0.407 cm). This contradicts the expected EC > EO trend and is due to outlier effects, measurement variability or residual adaptation in subjects, which may be caused by participant fatigue, adaptive learning, or compensatory overcorrection, suggesting a potential outlier that skews between-condition comparisons (Schniepp et al., 2013). As such, comparisons between ON vs. NO for NSEC are treated cautiously and excluded from inferential interpretations. These findings confirm that both stance complexity and visual input significantly influence sway dispersion, and that exoskeleton support reduces reliance on intrinsic neuromuscular compensation, especially under instability.

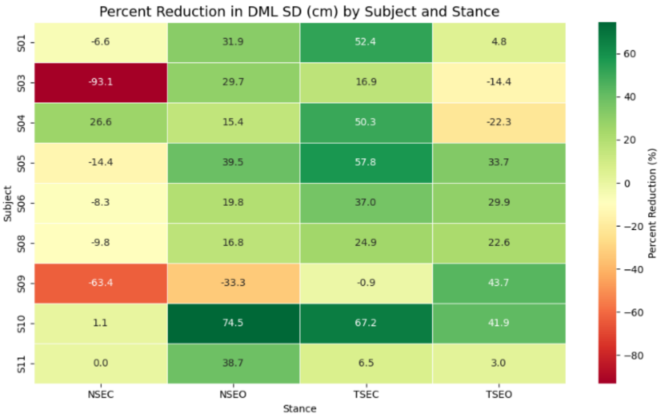


Figure 4.49: Heatmap of Percentage Reduction in DMLSD.

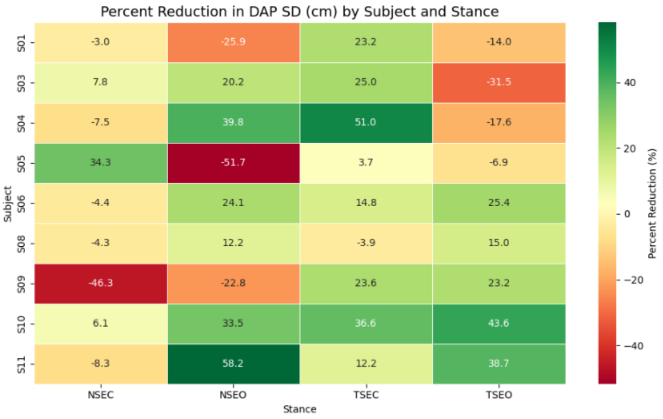


Figure 4.50: Heatmap of Percentage Reduction in DAPSD.

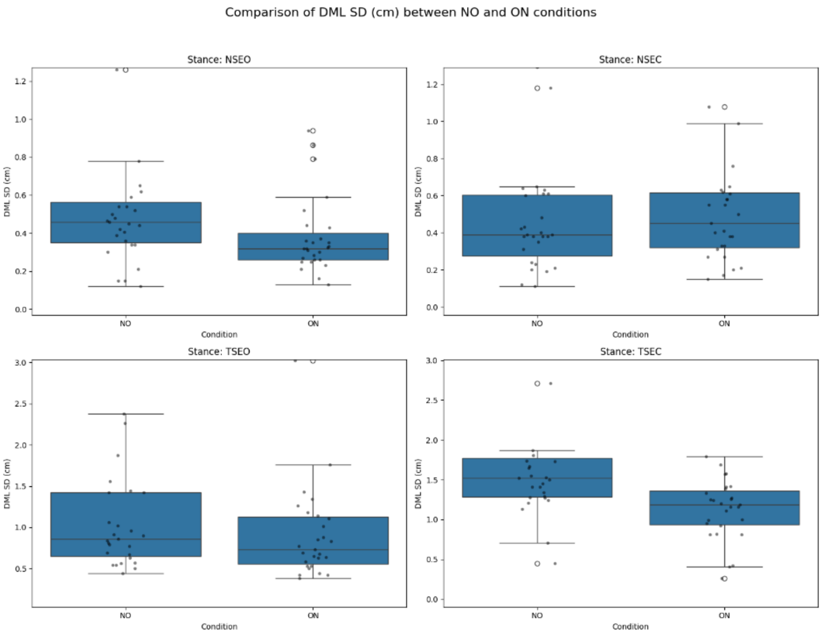


Figure 4.51: Boxplot Comparison of DMLSD Between With and Without Exoskeleton Conditions.

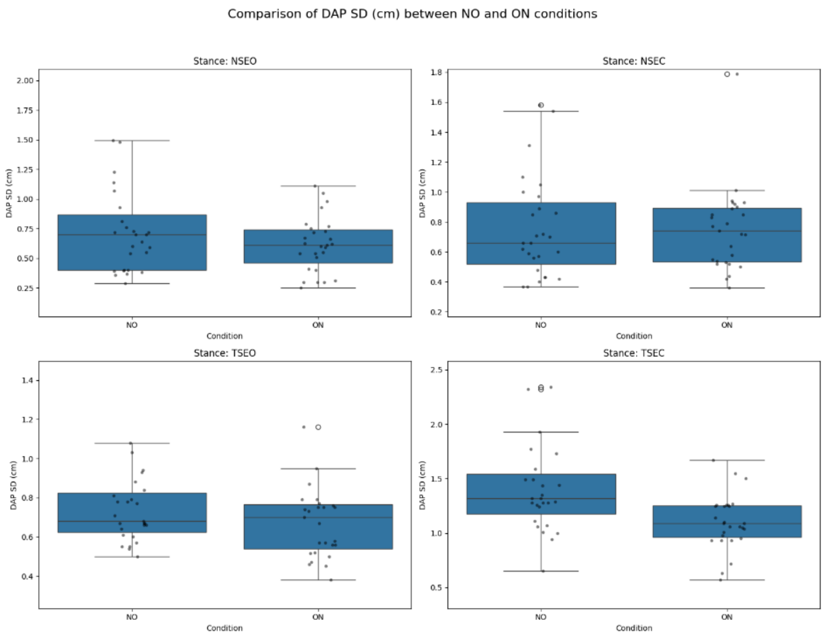


Figure 4.52: Boxplot Comparison of DAPSD Between With and Without Exoskeleton Conditions.

Mean CoP velocity metrics, Vmlmean and Vapmean, are indicative of the frequency and intensity of postural corrective actions. Elevated velocities typically correspond to increased instability and compensatory motor or correction effort (Tipton et al., 2023). The TSEC condition produced the highest velocities without exosuit assistance, with Vmlmean reaching 4.576 cm/s and Vapmean 3.920 cm/s. When the exosuit was engaged, these values significantly decreased to 2.469 cm/s (46.0% reduction) and 2.792 cm/s (28.8% reduction), respectively. These velocity reductions substantiate the exoskeleton’s efficacy in stabilising posture by reducing the frequency of corrective CoP shifts. These findings corroborate the stabilising influence of mechanical augmentation, which reduces the frequency of CoP corrections, visualised in Figures 4.53 and 4.54. Comparable effects have been reported in populations utilising external balance supports or subjected to altered sensory environments (Bauer et al., 2008). Subject consistency of improvement was observed more in Vmlmean compared to Vapmean, but still meaningful providing improvements.

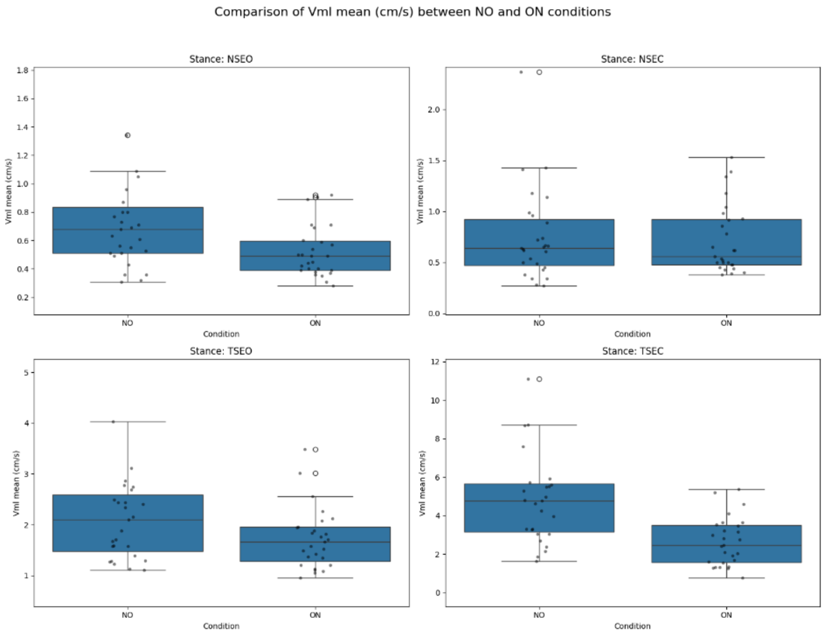


Figure 4.53: Boxplot Comparison of Vmlmean Between With and Without Exoskeleton Conditions

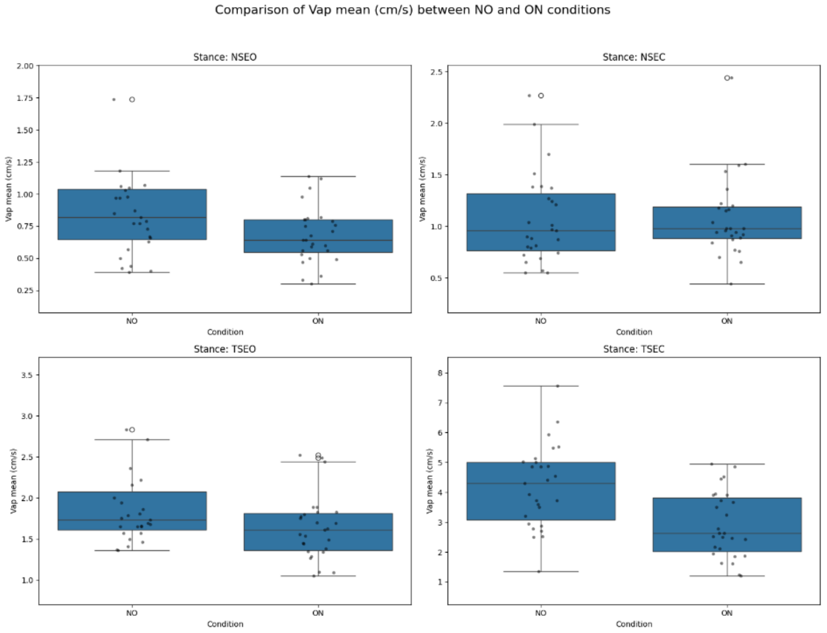


Figure 4.54: Boxplot Comparison of Vapmean Between With and Without Exoskeleton Conditions

Across all conditions, EC trials consistently yielded higher velocity values than EO trials, reaffirming the essential role of visual input in modulating postural stability. The tandem stance conditions, particularly under EC, exacerbated sway demands, as noted in prior work (Tipton et al., 2023).

Velocity metrics were most elevated in TSEC, reinforcing that tandem stance and visual deprivation elicit more frequent balance corrections. These findings align with Blaszczyk et al. (2020), who linked increased CoP velocity to reactive balance corrections under destabilised conditions. Importantly, the exosuit effectively moderated CoP velocities across all stances, reflecting reduced corrective demands and improved balance, indicating the exosuit dampens unnecessary oscillations, promoting smoother, more controlled sway patterns.. Nevertheless, velocity metrics, especially Vapmean, exhibited inter-subject variability. While large effect sizes were observed, statistical significance was inconsistent, highlighting the parameter's sensitivity to trial-level fluctuations and noise, a trend consistent with observations by Butowicz et al. (2023). Figures 4.55 and 4.56 also indicate the variation between subjects.

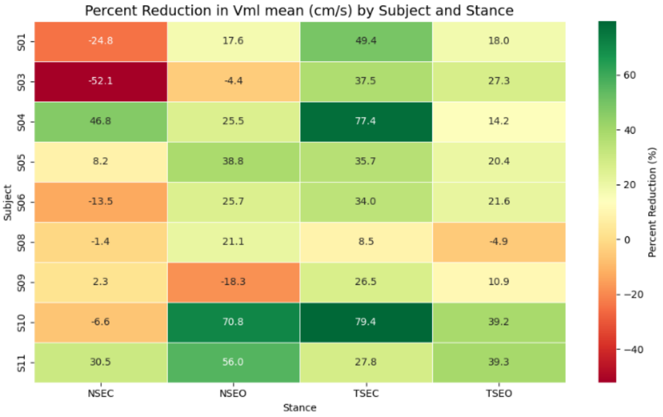


Figure 4.55: Heatmap of Percentage Reduction in Vmlmean.

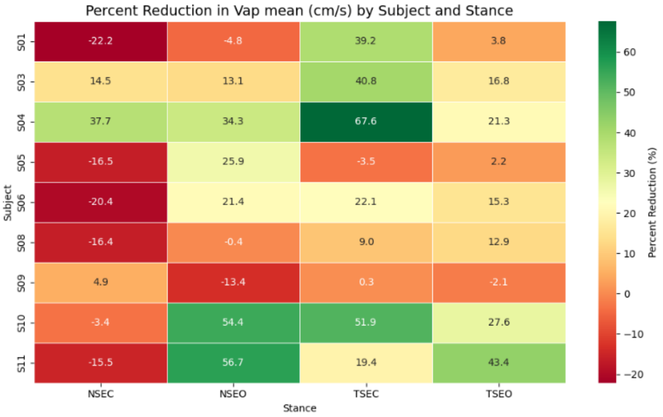


Figure 4.56: Heatmap of Percentage Reduction in Vapmean.

Pathlength quantifies the cumulative distance travelled by the CoP and serves as an aggregate measure of sway magnitude and neuromuscular effort. Without exosuit assistance, pathlength was highest in the TSEC condition at 197.852 cm. With the exosuit engaged, this value dropped markedly to 123.371 cm (a 37.7% reduction). Similar improvements were noted in TSEO (92.5 cm to 74.8 cm; 19.1% reduction) and NSEO (38.65 cm to 27.20 cm; 29.6% reduction). These observations, shown in boxplots in Figure 4.57, are consistent with previous studies indicating that EC and tandem stances exacerbate pathlength due to limited visual feedback and reduced base of support (Tipton et al., 2023; Sundaram et al., 2012; Donath et al., 2012). The increase in pathlength under EC conditions reflects greater sway complexity and amplitude, in line with the findings of Bauer et al. (2008). The reduction of pathlength under exoskeleton-assisted conditions indicates a decrease in postural workload, particularly in the most destabilising condition, TSEC.

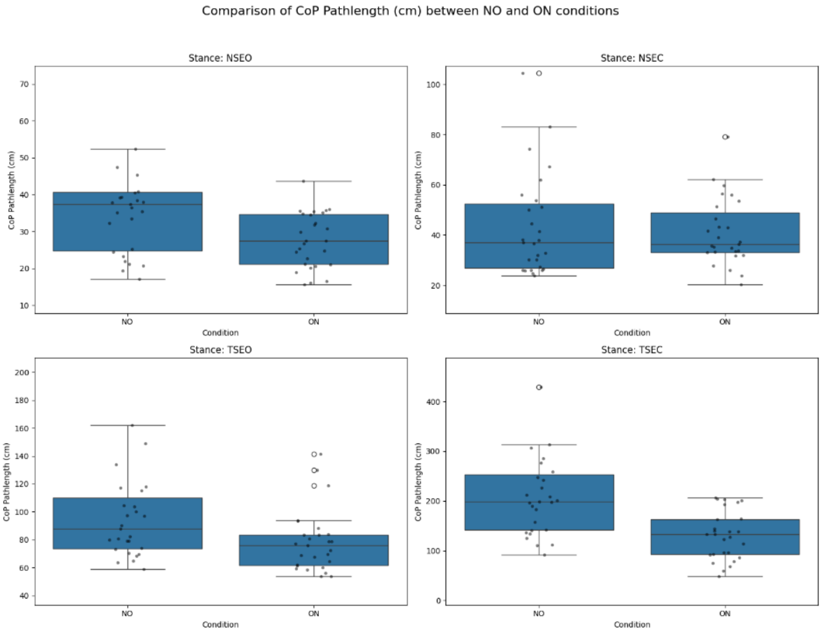


Figure 4.57: Boxplot Comparison of Pathlength Between With and Without Exoskeleton Conditions

These results further substantiate the utility of the exoskeleton in limiting the extent of sway excursions, thereby reducing the neuromuscular effort required to maintain balance. These values align with reports that increased CoP pathlength is a hallmark of balance degradation under sensory conflict (Filho et al., 2024). Studies by Donath et al. (2012) and Rezaeipour (2018) also validated pathlength as a robust metric for detecting instability, especially under EC and tandem conditions where visual guidance is absent or base-of-support is minimal. Heatmap in Figure 4.58 shows general improvements in pathlength parameter but high variations of reduction percentage.

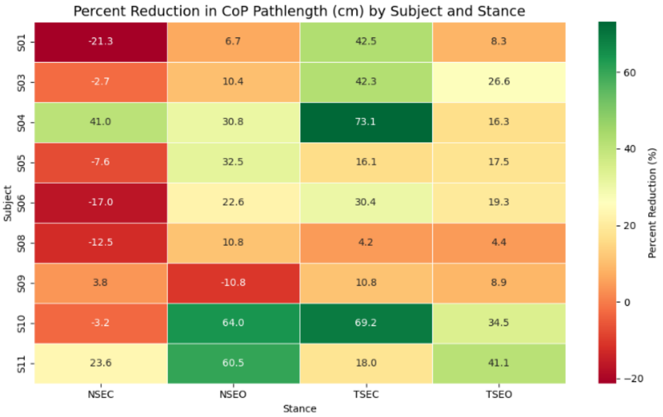


Figure 4.58: Heatmap of Percentage Reduction in Pathlength.

The 95% PEA captures the spatial area encompassing the majority (95%) of CoP movements and is sensitive to both stance complexity and visual input. In TSEC, PEA was reduced from 46.160 cm² to 19.371 cm² with exosuit assistance, representing a 58.0% reduction. Similarly, TSEO saw a reduction from 18.6 cm² to 10.0 cm² (46.2%), while NSEO improved from 8.161 cm² to 4.176 cm² (49%). These results highlight the exosuit's capacity to restrict the spatial boundaries of CoP movement, thereby contributing to enhanced postural stability, visualised in Figure 4.59.

As anticipated, EC conditions consistently resulted in larger PEAs, affirming the role of visual input in spatial sway regulation (Filho et al., 2024). Findings are supported by Kozinc et al. (2023), who noted that sway area expands with EC and stance complexity, and that assistive devices can effectively restore spatial coherence in CoP trajectories. The observed reductions under exosuit conditions (Figure 4.60 for per subject) reflect the influence of mechanical constraint in mitigating excessive sway. These findings are supported by literature suggesting that external stabilisation enhances proprioceptive feedback and limits sway area (Nagymáté & Kiss, 2016). The dramatic reductions in PEA across all conditions underscore the exoskeleton's efficacy in spatially confining CoP movement through corrective torque application.

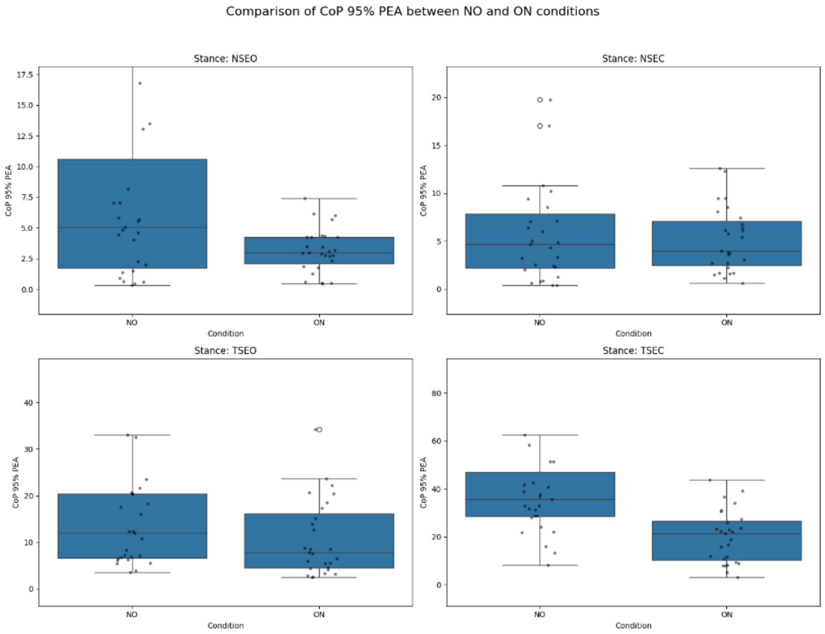
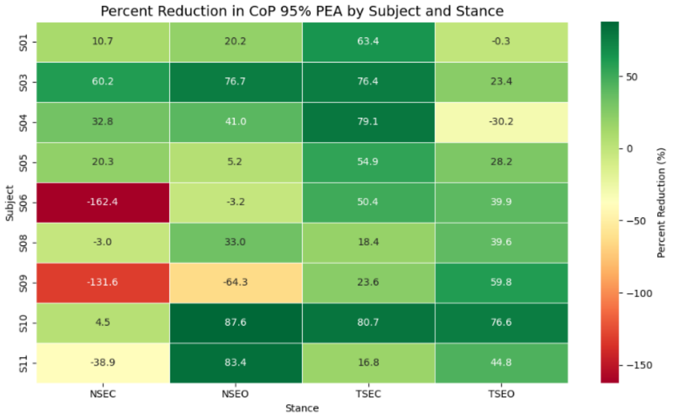


Figure 4.59: Boxplot Comparison of 95% PEA Between With and Without Exoskeleton Conditions



Subject	NSEC	NSEO	TSEC	TSEO
S01	10.7	20.2	63.4	-0.3
S03	60.2	76.7	76.4	23.4
S04	32.8	41.0	79.1	-30.2
S05	20.3	5.2	54.9	28.2
S06	-162.4	-3.2	50.4	39.9
S08	-3.0	33.0	18.4	39.6
S09	-131.6	-64.3	23.6	59.8
S10	4.5	87.6	80.7	76.6
S11	-38.9	83.4	16.8	44.8

Figure 4.60: Heatmap of Percentage Reduction in 95% PEA.

4.6.4 Conditions Interpretation of Exosuit Impact on Postural Stability

The influence of the exosuit on postural stability was condition-dependent, with the most substantial effects observed under challenging stance-visual combinations. Where Figures 4.61 shows the mean percent reduction of conditions vs parameters and Figure 4.62 their distributions.

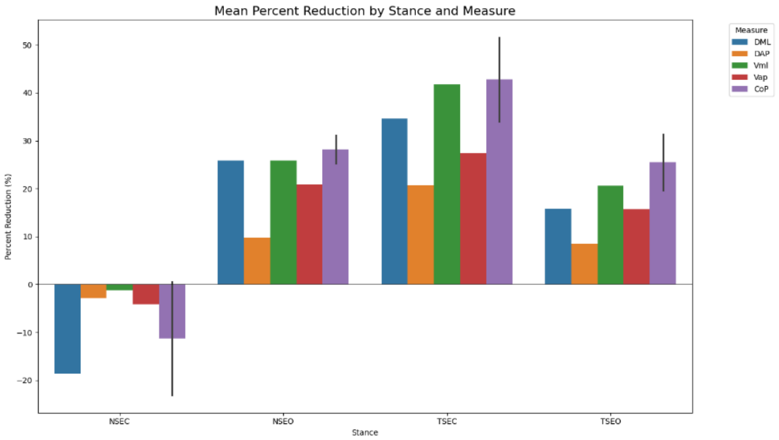


Figure 4.61: Mean Percent Reduction of COP Parameters by Stance.

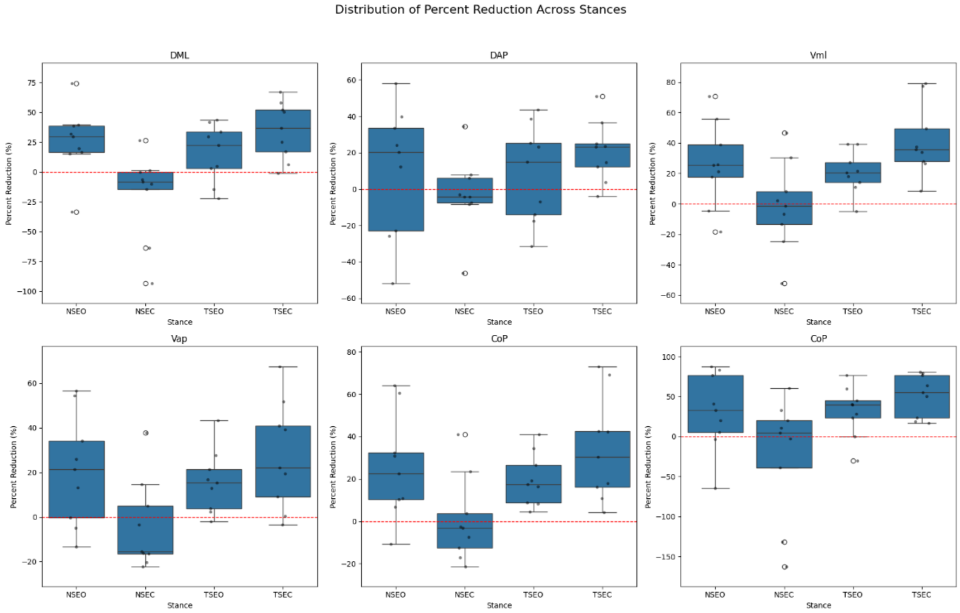


Figure 4.62: Percent Reduction Distribution of Cop Parameters.

TSEC emerged as the most destabilising condition, with the exosuit yielding the greatest improvements across all metrics, mediolateral sway dispersion (DMLSD) decreased by 35.7%, CoP pathlength by 37.7%, prediction ellipse area (PEA) by 58.0%, and mean mediolateral velocity (Vmlmean) by 46.0%. This condition best demonstrates the exosuit's efficacy under compounded sensory-motor challenges and supports its application in high-risk populations or rehabilitation settings. In TSEO, although visual input was available, balance remained significantly challenged due to the narrow base-of-support. The exosuit continued to show marked effectiveness, reducing DMLSD by 18.5%, PEA by 46.2%, and Vmlmean by 21.5%. These results reinforce that mechanical support offers tangible benefits even when visual feedback is intact, highlighting the exosuit's potential for general postural enhancement during demanding tasks.

NSEO represented a stable baseline, yet the exosuit halved the sway area (PEA reduced by 49%) and reduced DMLSD from 0.526 to 0.408 cm. Though improvements were less dramatic, they remained functionally relevant. These results affirm that even under low-demand conditions, the exosuit tightens postural control through mechanical constraint, consistent with findings linking visual orientation to reduced CoP variability. NSEC exhibited atypical patterns. Contrary to the expected trend of greater instability with visual deprivation, DMLSD was higher in the eyes-open condition. While PEA and pathlength slightly decreased with exosuit support, DMLSD and DAPSD unexpectedly increased. These inconsistencies, due to subject adaptation, measurement variability, or floor effects—limit inferential comparisons for this condition. Nonetheless, the overall reduction in sway area and pathlength suggests marginal stability gains. Table 4.8 encapsulates the exosuit's stabilising effect under tandem stance conditions, which represent the most mechanically and sensorily challenging balance scenarios. These findings collectively confirm the exosuit's effectiveness in mitigating postural sway, particularly under dual challenges of visual deprivation and mechanical instability. The reductions in CoP variability and sway extent suggest enhanced trunk control and reduced neuromuscular effort, consistent with literature indicating that exogenous torque assistance minimises corrective movements (Dobberke et al., 2022). These condition-wise results validate the exosuit's potential for balance support and fall risk mitigation.

Table 4.8: Average Percentage Reduction in Postural Metrics with Exosuit Support (TSEC + TSEO Conditions)

Parameter	**Average % Reduction**
DMLSD (Mediolateral Sway Dispersion)	24.6%
Vmlmean (Mean Mediolateral Velocity)	37.9%
Pathlength (Total CoP Path)	34.2%
95% Prediction Ellipse Area (PEA)	52.0%

In summary, the implementation of an active back-support exoskeleton significantly enhanced postural control across all analysed parameters. The most pronounced improvements were observed under tandem stance and eyes-closed conditions, where balance demands were highest. Sensory-motor adaptation effects were evident, with EC trials consistently amplifying sway metrics, reinforcing the critical dependency of balance on visual feedback (Sezer & Ferdjallah, 2005; Bauer et al., 2008). Anomalous trends, such as the elevated DMLSD in NSEO compared to NSEC under the NO condition, suggest potential outlier effects and warrant cautious interpretation. Clinically, these results support the potential application of developed active exoskeleton as assistive devices for individuals with proprioceptive or neuromuscular impairments, such as the elderly or stroke survivors, offering a promising avenue for enhancing balance and reducing fall risk.

4.7 Summary

The strap-based, back-supported strap-based pneumatic exosuit developed in this study represents a promising innovation in enhancing postural stability and mitigating muscular strain during prolonged standing. By integrating mechanical assistance with user-centred features and real-time postural monitoring, the system demonstrates clear utility for clinical rehabilitation, occupational health, and fall prevention in aging populations. Functional performance was supported by reductions in superficial trunk muscle activity, improvements in CoP metrics, and the successful implementation of assistive technologies, including a sway detection algorithm, visual feedback interfaces, and data visualisation tools.

EMG analyses revealed significant reductions in activity across the EO and RA under all stance and visual conditions, indicating that the exosuit effectively redistributed postural load away from superficial muscles. This shift toward deeper core stabiliser engagement aligns with Dynamic Neuromuscular Stabilisation (DNS) principles, wherein optimal postural control is achieved through minimised superficial activation and enhanced core recruitment (Huang et al., 2024). Such biomechanical efficiency reduces fatigue and lowers the risk of overuse injuries, particularly in static or repetitive work environments. However, EMG data from the ES displayed asymmetrical patterns, with inconsistent reductions and left-dominant activation during tandem stance. These lateral discrepancies suggest uneven mechanical force distribution or compensatory neuromuscular responses, particularly under increased balance demands. This aligns with findings from Alderink et al. (2024), who observed increased neural coupling complexity and asymmetric intermuscular coordination during challenging postures such as tandem stance. These findings underscore the need to refine the exosuit's mechanical symmetry and adaptability to better support dynamic spinal loading conditions.

Postural stability improvements, evidenced by reductions in CoP variability, DAPSD and DMLSD, CoP velocity (Vmean), and 95% PEA further affirm the system's functional capacity. These improvements were most pronounced during eyes-closed and tandem stance conditions, highlighting the exosuit's potential to maintain stability under reduced sensory feedback. Legrand et al. (2024) emphasise the increased cortical involvement in such conditions and the ability of external supports to alleviate sensorimotor burden. This is corroborated by DNS-based interventions, which report improved CoP dynamics following trunk stabilisation protocols. The integration of EMG and CoP analyses offers a powerful lens into the neuromechanical mechanisms of postural control, particularly when evaluating assistive technologies such as back-support exosuits. The observed reductions in CoP variability and excursion, concurrent with lower EMG activity, indicate that the exosuit facilitated more efficient balance control strategies by modulating trunk stiffness and reactive muscle tone.

In this study, the pattern of reduced EMG yet improved CoP parameters counters the inefficient compensatory strategies seen in individuals with chronic low back pain, where high trunk muscle activity often coincides with greater sway (Sung et al., 2024). These findings collectively suggest that the exosuit not only redistributes trunk load and enhances spinal support but also facilitates a more efficient neuromuscular strategy, where less muscular effort yields better postural outcomes. This transition from a high-EMG/high-sway state to a low-EMG/low-sway paradigm underscores the exosuit's role in promoting stable and energy-efficient standing balance. Sway variability, as captured by both the DAPSD and DMLSD, was significantly reduced under all test conditions. Corresponding reductions in EO and RA EMG suggest that the exosuit offloaded anterior trunk musculature, thereby improving sagittal stability. In contrast, ML sway control appeared more dependent on the bilateral coordination of the ES, particularly under tandem stance. An observed asymmetry, characterised by a right-sided ES EMG reduction and a static or increased response on the left, reflects compensatory strategies during lateral load transfer. This asymmetry corresponds with prior findings (Alderink et al., 2024) indicating that ML sway stability relies heavily on symmetric paraspinal recruitment.

In terms of CoP velocity (Vmean), both AP and ML directions showed marked reductions. Lower EO and RA EMG values suggest that less reactive muscular effort was required to control sway momentum, consistent with smoother and more dampened postural adjustments. DNS-trained groups have shown similar declines in CoP velocity under eyes-closed conditions due to enhanced core recruitment, and the exosuit seems to mimic this effect via mechanical assistance. Notably, the suppression of right ES EMG in the tandem condition coincided with lower ML velocity, indicating that targeted spinal support contributes to lateral damping. These findings align with accelerometery studies identifying Vmean as a sensitive proxy for neuromuscular stabilisation demands (Slunecko & Csapo, 2024). Global postural stability was further evidenced by reduced CoP path length and 95% PEA, both of which integrate multidirectional sway data. These metrics serve as cumulative indicators of overall postural footprint. EO and RA unloading, coupled with partial ES activation, appeared sufficient to constrain CoP excursions. However, persistent PEA values in the most challenging conditions, TSEC, imply that incomplete bilateral ES support may limit full postural optimisation. Such results are corroborated by literature on trunk muscle fatigue and asymmetry, which links lateral paraspinal imbalance to increased CoP area and instability (Floessel et al., 2024). The integration of EMG–CoP relationships is summarised in Table 4.9, capturing parameter-specific muscle contributions and their interpretive relevance:

The exosuit’s sway detection algorithm, while initially based on machine learning approaches with modest accuracy (25% unsupervised learning, k-means), was successfully replaced by a supervised learning framework (Random Forest, KNN, SVM) approximately 63% accuracy but higher computational toll, then threshold-based model that achieved over 70% classification accuracy through iterative empirical tuning. Though less robust than advanced model-based classifiers, this approach offers practical real-time sway detection and paves the way for future integration with adaptive control systems. In its current form, it provides continuous postural monitoring, enhances diagnostic capabilities, and facilitates user awareness through visual cues. Operational practicality is reinforced by efficient donning and doffing procedures, requiring approximately 3:02 and 1:06 minutes, respectively, durations that support use in time-sensitive clinical or occupational workflows. The pneumatic actuation system, selected for its lightweight and responsive qualities over electric motors, contributes to a total device weight of 4.7 kilograms, optimising portability without compromising support. A laptop-based graphical user interface (UI) enables real-time visualisation of EMG and sway metrics, data logging, and longitudinal comparisons, while a wrist-mounted OLED display offers immediate visual feedback to users, enhancing in-situ awareness without external dependencies.

Table 4.9: Summary of EMG–CoP Parameter Correlations

CoP Parameter	Correlated EMG Muscle(s)	Key Interpretation
SD-AP	EO ↓, RA ↓	Sagittal sway reduction via anterior muscle unloading
SD-ML	ES (R ↓, L ↔/↑)	ML sway control via bilateral paraspinal coordination
Vmean-AP/ML	EO/RA ↓, ES-R ↓	Reduced sway velocity reflects lower reactive muscle tone
Path Length	EO/RA ↓, partial ES ↓	Total sway burden reduced through trunk muscle offloading
95% PEA	EO/RA + ES synergy	Global sway footprint minimised by integrated core activation

Participant feedback supports the system's ergonomic viability, with users describing the exosuit as comfortable and functionally engaging. Although some noted the abundance of straps, none reported movement restrictions, and several remarked on the perceptible mechanical engagement during postural corrections. Such feedback is crucial for balancing corrective function with user comfort in future design iterations. Nonetheless, several limitations require attention. Asymmetric ES activity suggests a need for improved bilateral mechanical alignment, and control segmentation may help achieve more uniform force application. While the current sway detection system provides baseline functionality, future enhancements, such as frequency-domain features, multi-sensor fusion, or adaptive machine learning algorithms—could improve accuracy and robustness. The absence of haptic or auditory feedback, modalities shown to enhance balance outcomes in older adults (Suttanon et al., 2024), represents another area for potential augmentation. Long-term adaptation and retention effects also remain unexplored, limiting insight into sustained use and motor learning implications.

The results align closely with broader trends in wearable support systems and sensorimotor rehabilitation. DNS-centred interventions have demonstrated postural gains via targeted core activation (Huang et al., 2024), while cortical modulation during sensory-compromised balance tasks supports the use of external mechanical aids (Legrand et al., 2024). Asymmetric recruitment of paraspinal musculature during tandem stance, noted by Alderink et al. (2024), parallels the observed ES activity patterns. Additionally, real-time feedback has been shown to improve postural correction strategies (Suttanon et al., 2024), suggesting that multimodal cues could further enhance the system. Finally, the utility of time-in-boundary and longitudinal sway metrics, as highlighted by Sung and Lee (2024a), underscores the importance of continuous tracking, already facilitated by the exosuit's visualisation and logging capabilities.

In summary, this exosuit demonstrates functional and practical efficacy in reducing superficial muscle strain and improving postural stability across a range of stance complexities. Its design, combining pneumatic assistance, real-time sway tracking, and user-centric features, establishes a compelling foundation for clinical, occupational, and preventive applications. While refinement is warranted in certain mechanical and algorithmic domains, the system shows significant promise for future integration of adaptive feedback, expanded sensor capabilities, and long-term motor training solutions.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This project successfully developed a strap-based, back-supported, pneumatic-powered exosuit designed to mitigate standing postural sway through active correction mechanisms. The system integrated pneumatic actuation, wearable inertial sensors, a real-time sway classification algorithm, and a graphical user interface (GUI). Key objectives were met: a 4.7 kg exosuit was designed, a chest-mounted IMU was implemented for sway detection, and a functional sway-detection and actuation pipeline was developed. The system demonstrated the potential to improve postural stability, particularly in individuals with impaired balance such as those with Parkinson's disease (PD) or low back pain (LBP). CoP and EMG tests were done for functionality and results were good and promising. The sway detection system, though rule-based, was implemented as a modular and embedded-compatible classification pipeline for real-time monitoring. While machine learning models showed promise, further optimization is needed for deployment. The complete system integrates signal processing, feature extraction, and classification, balancing high-level biomechanical insights with low-power embedded control requirements. Testing with healthy subjects confirmed the system's ability to detect distinct sway patterns and its feasibility for embedded applications, demonstrating interpretability, responsiveness, and real-world integration potential.

5.2 Key Findings/ Achievements

The prototype exosuit performed well in terms of sway mitigation and usability. The GUI facilitated real-time monitoring and data logging. Sway detection worked effectively across multiple stance and visual feedback conditions, especially for anterior-posterior (AP) sway. While the system struggled with asymmetrical and unstable sway patterns, it showed promise as an assistive device for individuals with compromised postural control. Validation using EMG and center of pressure (CoP) analysis revealed effective support in sway reduction and postural stabilization. Notably, reductions in EMG activity were observed in trunk muscles, especially the rectus abdominis (RA) and right erector spinae (ES-R), by up to 51.2% and 41.8%, respectively. The improvements were most evident during challenging conditions such as tandem stance with eyes closed (TSEC), suggesting the exosuit's effectiveness in high-demand postural scenarios. Reductions exceeding 30% were also observed in CoP metrics including mediolateral sway dispersion (DMLSD), velocity, pathlength, and prediction ellipse area.

5.2.1 Sway Detection Algorithm

The final threshold-based classifier achieved an accuracy of 70%, surpassing all tested machine learning (ML) models in both performance and deployability. Among ML approaches, the Random Forest model with bagging achieved the highest accuracy at 63.5%, highlighting its resilience to sensor noise and multi-class complexity. Stable, DAP, and DML sway conditions were distinguishable; however, the INST (unstable) class showed high variability and overlap. The signal processing pipeline—featuring Butterworth filtering, Z-score normalization, median absolute deviation (MAD), and wavelet-based outlier rejection—effectively reduced artifacts. Training included over 130,000 five-second windows and over 3.7 million short (250 ms) windows, though hardware limitations restricted full validation of high-frequency detection.

5.2.2 Biomechanical Tests

Functionally, the exosuit reduced trunk muscle strain and promoted more stable postural strategies. Surface EMG showed significant reductions in tonic muscle activation, particularly under sensory-compromised conditions. Reductions in CoP variability across all relevant metrics further affirmed improved neuromuscular control. Compared to passive or assistive devices in literature, the active exosuit provided superior unloading and balance-enhancing effects, validating its utility in challenging postural contexts.

5.3 Limitations

Despite the promising development of the prototype, several limitations were encountered during the research and prototyping phases.

5.3.1 Hardware and Design Constraints

One of the primary limitations was hardware-related, due to cost constraints. Budget restrictions limited the choice of materials, actuator technologies, and supporting electronics. The use of basic materials for rigid anchor points, instead of advanced composites like carbon fibre, introduced additional weight and reduced the ergonomic profile of the exosuit. Carbon fibre would have provided a more lightweight, durable, and aesthetically refined structure suitable for prolonged wear. Moreover, the reliance on pneumatic actuation presented notable drawbacks. Pneumatic systems, while effective in generating sufficient force, were noisy, bulky, less energy-efficient, and less precise than their electrical counterparts. They also required a relatively high power input, which reduced battery life. In contrast, electric actuators could offer a quieter, more compact, and more precise alternative, with improved controllability and lower energy consumption.

5.3.2 Algorithm Development Constraints

Several limitations were identified in this study. First, the participant pool consisted solely of healthy adults, restricting the generalizability of results to clinical or elderly populations with impaired balance. The INST class remained a persistent challenge due to the ambiguous nature of unstable sway, which may not be easily captured in static, short-time-window features. Furthermore, while the classifier was tested using 5-second windows, the 250-millisecond real-time implementation could not be fully validated due to the computational limitations of the development hardware. The system also relied exclusively on accelerometer data; the absence of gyroscope, barometer, or multi-modal inputs may have constrained classification performance. Lastly, testing was conducted under controlled laboratory settings, which do not fully replicate real-world variability or extended-duration use, such as might occur in industrial, clinical, or rehabilitative scenarios.

5.3.3 Time Constraints

Time limitations significantly impacted the development of both the software stack and the evaluation protocols. The graphical user interface (GUI), while partially functional, was not fully deployable. Features such as containerisation via Docker and remote accessibility were not yet implemented, limiting the system’s scalability and usability in clinical settings. Algorithmic development also remained incomplete. Full classification and detection capabilities for impaired postural sway patterns—particularly those related to asymmetrical loading or neuromuscular disorders—could not be validated. Additionally, time constraints restricted the extent of experimental testing, which was limited to a small sample size composed solely of healthy, young adults.

5.3.4 System Performance and Experimental Limitations

Functionally, the system demonstrated reduced reliability in detecting and correcting left-sided sway compared to anterior-posterior and right-sided sways. Battery life was constrained to approximately 15 minutes due to the high current draw from the solenoid valves and the MiPump2 compressor unit, limiting long-duration testing scenarios. Moreover, the system was not tested on the actual target population—individuals with postural instability due to aging, Parkinson’s disease (PD), or low back pain (LBP). This limits the generalisability of findings and highlights the need for more comprehensive clinical validation.

Also, certain limitations were noted. The EMG response displayed lateral asymmetries, particularly in the left erector spinae, suggesting inconsistencies in torque distribution or individual adaptation strategies. Such discrepancies raise concerns about the long-term musculoskeletal balance and highlight the importance of personalisation in exosuit design. Additionally, the system's reliance on a tethered pneumatic source and laptop-based interface, while functional, limits its immediate applicability in fully mobile settings. The sway detection algorithm, though effective in a simplified threshold-based model, lacked the robustness and adaptability of more complex machine learning approaches due to computational limitations and dataset constraints. Moreover, while postural benefits were demonstrated under controlled laboratory conditions, real-world effectiveness in dynamic or prolonged occupational settings remains to be evaluated.

5.4 Recommendations for Future Work

The hardware validation presented is partially addressed through component selection rationale, IMU calibration, and basic physical integration testing. However, there is a lack of performance testing on durability, repeatability, and mechanical reliability, to justify about the system's long-term robustness and real-world dependability. Future work should include testing for addressing these concerns. Future research should prioritize adaptive control systems that use biofeedback from EMG and inertial sensors to personalize torque assistance and address muscle asymmetries. Enhancing real-time calibration and symmetric support mechanisms may reduce lateral imbalances and improve comfort and safety over extended use. Though the current system is already wireless and portable, optimizing power efficiency and battery life will further enhance usability. Expanding validation in clinical populations and under dynamic conditions (e.g., posture during gait or occupational tasks) is essential to demonstrate broader applicability.

5.4.1 Hardware and Software Improvements

Transitioning to electric actuators could offer better efficiency, control precision, and quieter operation. Advanced microcontrollers (e.g., Raspberry Pi, Jetson) could support on-device machine learning for adaptive control. Containerisation of GUI (e.g., Docker) and cross-platform deployment will facilitate clinical testing and broader adoption. GUI improvements should also include enhanced calibration tools, real-time feedback, and remote access. Incorporating symmetry analysis and user-specific thresholds will improve system responsiveness to individual biomechanical differences.

5.4.2 Further Testing and Experiments

Future experiments should use tools like force plates, motion capture, and EMG to evaluate dynamic postural responses. Introducing balance challenges (e.g., unstable surfaces) would better simulate real-world use. Trials involving older adults, people with Parkinson's disease, and those with low back pain should assess both short- and long-term biomechanical impacts.

5.4.3 Machine Learning Based Detection Algorithm Development

Detection accuracy can be enhanced by hybrid models (e.g., fuzzy logic with decision trees) and temporal networks like LSTM or GRU. Alternative signal representations like Dynamic Difference of Vector Distances (DDVD) may offer improved noise tolerance and classification accuracy in real-world environments (Nehry et al., 2023). Compressed models (TinyML) can enable real-time inference on microcontrollers. Data augmentation and EMG-integrated co-adaptive control will support personalized and robust classification for clinical users.

5.4.4 System Expansion and Adaptive Algorithms

To enhance personalization, future designs should adopt adaptive filtering and context-aware thresholds to distinguish between intentional movement and unintended sway. Real-time bilateral EMG feedback can dynamically balance actuator output and address asymmetrical loading, especially in prolonged static tasks. Long-term validation is needed to assess risks such as proprioceptive drift or chronic postural compensation. Customizable hardware, individualized calibration, and dynamic control adaptation will be key to safe, reliable, and comfortable use across diverse settings.

5.5 Final Remarks

The development of this back-supported exosuit presents a meaningful step toward the development of wearable assistive technologies for postural control. The proposed exosuit offers an integrative platform for both research and practical applications in rehabilitation and occupational support. Despite current limitations, the system demonstrates clear potential to reduce muscular strain and improve balance in individuals with compromised postural stability. Through further refinement and clinical validation, this technology can be transformed into a portable, intelligent, and effective assistive solution capable of addressing real-world balance impairments and improving quality of life across various populations.

REFERENCES

Ai, L., Men, X., Zhou, T., Xiao, X. and Guo, Z. (2021) ‘Design and Control of a Cable-Driven Series Elastic Actuator for Exoskeleton’, pp. 335–339. Available at: https://doi.org/10.1109/icoias53694.2021.00066.

Alderink, G., Rhodes, S., Zeitler, D. and McCrumb, D. (2025) ‘Analysis of Connectivity in EMG Signals to Examine Neural Correlations in the Activation of Lower Leg Muscles for Postural Stability: A Pilot Study’, *Bioengineering* [Preprint]. Available at: https://doi.org/10.20944/preprints202411.1306.v1.

Alemi, M.M., Simon, A.A., Geissinger, J. and Asbeck, A.T. (2022) ‘Modeling the metabolic reductions of a passive back-support exoskeleton’, *Journal of Applied Physiology*, 132(3), pp. 737–760. Available at: https://doi.org/10.1152/japplphysiol.00171.2021.

Alshahrani, A., Reddy, R.S. and Ravi, S.K. (2025) ‘Chronic low back pain and postural instability: interaction effects of pain severity, age, BMI, and disability’, *Frontiers in Public Health*, 13. Available at: https://doi.org/10.3389/fpubh.2025.1497079.

Anderson, A., Richburg, C., Czerniecki, J. and Aubin, P. (2019) ‘A Model-Based Method for Minimizing Reflected Motor Inertia in Off-board Actuation Systems: Applications in Exoskeleton Design’, *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pp. 360–367. Available at: https://doi.org/10.1109/icorr.2019.8779452.

Andò, B., Baglio, S., Dibilio, V., Marletta, V., Marella, M., Mostile, G., Rajan, S., & Zappia, M., (2022a). A Neuro-Fuzzy Approach to Assess Postural Sway. *2022 IEEE Sensors Applications Symposium (SAS)*, pp. 1-6. https://doi.org/10.1109/SAS54819.2022.9881336.

Andò, B., Baglio, S., Finocchiaro, V., Marletta, V., Rajan, S., Nehary, E., Dibilio, V., Mostile, G., & Zappia, M., (2023). Machine Learning Approach to Classify Postural Sway Instabilities. *2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 01-06. https://doi.org/10.1109/I2MTC53148.2023.10176004.

Andò, B., Baglio, S., Graziani, S., Marletta, V., Dibilio, V., Mostile, G. and Zappia, M. (2022b) ‘A Comparison among Different Strategies to Detect Potential Unstable Behaviors in Postural Sway’, *Sensors*, 22(19), pp. 7106–7106. Available at: https://doi.org/10.3390/s22197106.

Andreeva, A., Melnikov, A., Skvortsov, D., Akhmerova, K., Vavaev, A., Golov, A., Draugelite, V., Nikolaev, R., Chechelnickaia, S., Zhuk, D., Bayerbakh, A., Nikulin, V. and Zemková, E. (2021) ‘Postural stability in athletes: The role of sport direction’, *Gait & Posture*, 89, pp. 120–125. Available at: https://doi.org/10.1016/j.gaitpost.2021.07.005.

Anson, E., Bigelow, R.T., Swenor, B., Deshpande, N., Studenski, S., Jeka, J.J. and Agrawal, Y. (2017) 'Loss of Peripheral Sensory Function Explains Much of the Increase in Postural Sway in Healthy Older Adults', *Frontiers in Aging Neuroscience*, 9. Available at: https://doi.org/10.3389/fnagi.2017.00202.

Antonelli, M.G., D'Ambrogio, W. and Durante, F. (2018) 'Development of a pneumatic soft actuator as a hand finger for a collaborative robot'. Available at: https://doi.org/10.1145/3185066.3185079.

Baiju, N. (2022) *Pneumatic artificial muscle robots: Benefits and challenges*, *RoboticsBiz*. Available at: https://roboticsbiz.com/pneumatic-artificial-muscle-robots-benefits-and-challenges (Accessed: 12 May 2025).

Barth, O. (2000) 'Harmonic piezodrive — miniaturized servo motor', *Mechatronics*, 10(4-5), pp. 545–554. Available at: https://doi.org/10.1016/s0957-4158(99)00062-8.

Bartlett, H.L., Ting, L.H. and Bingham, J.T. (2014) 'Accuracy of force and center of pressure measures of the Wii Balance Board', *Gait & Posture*, 39(1), pp. 224–228. Available at: https://doi.org/10.1016/j.gaitpost.2013.07.010.

Bauer, C., Gröger, I., Rupprecht, R. and Gaßmann, K.G. (2008) 'Intrasession Reliability of Force Platform Parameters in Community-Dwelling Older Adults', *Archives of Physical Medicine and Rehabilitation*, 89(10), pp. 1977–1982. Available at: https://doi.org/10.1016/j.apmr.2008.02.033.

Becker, S., Thomas, A., Ulrich, L., Becker, L., Dindorf, C., Berger, J. and Fröhlich, M. (2025) 'Reference values for static posturography of sportive and healthy adults aged 18–30 years', *BMC Sports Science, Medicine and Rehabilitation*, 17(1). Available at: https://doi.org/10.1186/s13102-025-01128-z.

Bianculli, A.J. (1970) 'Stepper motors: application and selection', *IEEE Spectrum*, 7(12), pp. 25–29. Available at: https://doi.org/10.1109/mspec.1970.5213082.

Błaszczyk, J., & Beck, M., (2023). Posturographic Standards for Optimal Control of Human Standing Posture. *Journal of Human Kinetics*, 86, pp. 7 - 15. https://doi.org/10.5114/jhk/159452.

Błaszczyk, J., (2016). The use of force-plate posturography in the assessment of postural instability. *Gait & posture*, 44, pp. 1-6 . https://doi.org/10.1016/j.gaitpost.2015.10.014.

Błaszczyk, J., Cieślińska-Świder, J., & Orawiec, R., (2021). New methods of posturographic data analysis may improve the diagnostic value of static posturography in multiple sclerosis. *Heliyon*, 7. https://doi.org/10.1016/j.heliyon.2021.e06190

Błaszczyk, J., Fredyk, A. and Błaszczyk, P. (2020) ‘Transition from double-leg to single-leg stance in the assessment of postural stability’, *Journal of Biomechanics*, 110, p. 109982. Available at: https://doi.org/10.1016/j.jbiomech.2020.109982.

Bong, Y., Shariff, A., Majid, A. and Merican, A. (2012) ‘Reference charts for height and weight of school children from west malaysia in comparison with the United States centers for disease control and prevention’, *Iranian journal of public health*, 41(2), pp. 27–38. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3481677.

Borujeni, G.B. and Yalfani, A. (2019) ‘Reduction of postural sway in athletes with chronic low back pain through eight weeks of inspiratory muscle training: A randomized controlled trial’, *Clinical Biomechanics*, 69, pp. 215–220. Available at: https://doi.org/10.1016/j.clinbiomech.2019.09.006.

Boyraz, P., Runge, G. and Raatz, A. (2018) ‘An Overview of Novel Actuators for Soft Robotics’, *Actuators*, 7(3), p. 48. Available at: https://doi.org/10.3390/act7030048.

Brumagne, S., Janssens, L., Knapen, S., Claeys, K. and Suuden-Johanson, E. (2008) ‘Persons with recurrent low back pain exhibit a rigid postural control strategy’, *European Spine Journal*, 17(9), pp. 1177–1184. Available at: https://doi.org/10.1007/s00586-008-0709-7.

Butowicz, C., Yoder, A.J., Hendershot, B.D., Gunterstockman, B. and Farrokhi, S. (2023) ‘Principal components analysis of postural sway in persons with unilateral lower limb amputation: A wearable sensor approach’, *Journal of Biomechanics*, 158, p. 111768. Available at: https://doi.org/10.1016/j.jbiomech.2023.111768.

Carroll, J.P. and Freedman, W. (1993) ‘Nonstationary properties of postural sway’, *Journal of Biomechanics*, 26(4-5), pp. 409–416. Available at: https://doi.org/10.1016/0021-9290(93)90004-x.

Chathuranga, H., Marriam, I., Chen, S., Zhang, Z., MacLeod, J., Liu, Y., Yang, H. and Chen, Y. (2022) ‘Multistimulus-Responsive Graphene Oxide/Fe_3O_4/Starch Soft Actuators’, *ACS Applied Materials & Interfaces*, 14(14), pp. 16772–16779. Available at: https://doi.org/10.1021/acsami.2c03486.

Chaudhry, H., Findley, T., Quigley, K.S., Bukiet, B., Ji, Z., Sims, T. and Maney, M. (2004) ‘Measures of postural stability’, *The Journal of Rehabilitation Research and Development*, 41(5), p. 713. Available at: https://doi.org/10.1682/jrrd.2003.09.0140.

Chen, F., Miao, Y., Zhang, L., Chen, S. and Zhu, X. (2022) ‘Triply Periodic Channels Enable Soft Pneumatic Linear Actuator With Single Material and Scalability’, *IEEE robotics and automation letters*, 7(2), pp. 2668–2675. Available at: https://doi.org/10.1109/lra.2022.3143292.

Choi, H.J. and Garlie, T. (2014) *Characterizing the Size of the Encumbered Soldier*. Available at: https://www.researchgate.net/publication/273134617_Characterizing_the_Size_of_the_Encumbered_Soldier..

Cholewicki, J., Peter Reeves, N., Everding, V.Q. and Morrisette, D.C. (2007) 'Lumbosacral orthoses reduce trunk muscle activity in a postural control task', *Journal of Biomechanics*, 40(8), pp. 1731–1736. Available at: https://doi.org/10.1016/j.jbiomech.2006.08.005.

Chu, W.-L., Lin, C.-J. and Chen, Y.-Y. (2022) 'Redundant Robot with Pneumatic Artificial Muscles for Rehabilitation Works Using Iterative Learning Control', *Applied Sciences*, 12(17), pp. 8419–8419. Available at: https://doi.org/10.3390/app12178419.

Chung, J., Quirk, D.A., Applegate, M., Rouleau, M., Degenhardt, N., Galiana, I., Dalton, D., Awad, L.N. and Walsh, C.J. (2024) 'Lightweight active back exosuit reduces muscular effort during an hour-long order picking task', *Communications Engineering*, 3(1), pp. 1–11. Available at: https://doi.org/10.1038/s44172-024-00180-w.

Copaci, D.-S., Blanco, D., Martin-Clemente, A. and Moreno, L. (2020) 'Flexible shape memory alloy actuators for soft robotics: Modelling and control', *International Journal of Advanced Robotic Systems*, 17(1), p. 172988141988674. Available at: https://doi.org/10.1177/1729881419886747.

Degani, A., Leonard, C., & Danna-dos-Santos, A., (2017). The effects of early stages of aging on postural sway: A multiple domain balance assessment using a force platform.. *Journal of biomechanics*, 64, pp. 8-15 . https://doi.org/10.1016/j.jbiomech.2017.08.029.

Do, T.N., Phan, H., Nguyen, T. and Visell, Y. (2018) 'Miniature Soft Electromagnetic Actuators for Robotic Applications', *Advanced Functional Materials*, 28(18), p. 1800244. Available at: https://doi.org/10.1002/adfm.201800244.

Dobberke, J., Baritello, O., Hadzic, M., Völler, H., Eichler, S. and Salzwedel, A. (2022) 'Test-retest reliability of center of pressure measures for postural control assessment in older cardiac patients', *Gait & Posture*, 92, pp. 359–363. Available at: https://doi.org/10.1016/j.gaitpost.2021.12.011.

Donath, L., Roth, R., Zahner, L. and Faude, O. (2012) 'Testing single and double limb standing balance performance: Comparison of COP path length evaluation between two devices', *Gait & Posture*, 36(3), pp. 439–443. Available at: https://doi.org/10.1016/j.gaitpost.2012.04.001.

dos Anjos, F.V., Ghislieri, M., Cerone, G.L., Pinto, T.P. and Gazzoni, M. (2022) 'Changes in the distribution of muscle activity when using a passive trunk exoskeleton depend on the type of working task: A high-density surface EMG study', *Journal of Biomechanics*, 130, p. 110846. Available at: https://doi.org/10.1016/j.jbiomech.2021.110846.

Ehara, Y., Inui, A., Mifune, Y., Yamaura, K., Kato, T., Furukawa, T., Tanaka, S., Masaya Kusunose, Shunsaku Takigami, Osawa, S., Nakabayashi, D., Hayashi, S., Matsumoto, T., Matsushita, T. and Kuroda, R. (2025) ‘The Development and Validation of an Artificial Intelligence Model for Estimating Thumb Range of Motion Using Angle Sensors and Machine Learning: Targeting Radial Abduction, Palmar Abduction, and Pronation Angles’, *Applied Sciences*, 15(3), pp. 1296–1296. Available at: https://doi.org/10.3390/app15031296.

El-Jaroudi, A., Redfern, M.S., Chaparro, L.F. and Furman, J.M. (1996) ‘The application of time-frequency methods to the analysis of postural sway’, *Proceedings of the IEEE*, 84(9), pp. 1312–1318. Available at: https://doi.org/10.1109/5.535249.

Espressif Systems (no date). Available at: https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf.

Faghihi, A., Haghpanah, S.A., Farahmand, F. and Jafari, M. (2015) ‘Design and fabrication of a robot for neurorehabilitation; Smart RoboWrist’, *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 447–450. Available at: https://doi.org/10.1109/kbei.2015.7436086.

Farris, D.J., Po, N., Yee, J., Williamson, J.L. and Taylor (2024) ‘Effects of passive ankle exoskeletons on neuromuscular function during exaggerated standing sway’, *Royal Society Open Science*, 11(5). Available at: https://doi.org/10.1098/rsos.230590.

Fattah, H. (2021) ‘Stepper Motor’, *CRC Press eBooks*, pp. 325–336. Available at: https://doi.org/10.1201/9781003120018-41.

Filho, U., Teixeira, E., Araujo, H., Assis, B., Jacob, T. and Azevedo, F. (2024) ‘Pressure center behavior and plantar pressure distribution in patients undergoing total knee atropasty surgery’, *Orthopaedic Journal of Sports Medicine*, 12(11_suppl4). Available at: https://doi.org/10.1177/2325967124s00467.

Fitts, R.H. (1994) ‘Cellular mechanisms of muscle fatigue’, *Physiological Reviews*, 74(1), pp. 49–94. Available at: https://doi.org/10.1152/physrev.1994.74.1.49.

Flieh, H.M., Lorenz, R.D., Totoki, E., Yamaguchi, S. and Nakamura, Y. (2018) ‘Investigation of Different Servo Motor Designs for Servo Cycle Operations and Loss Minimizing Control Performance’, *IEEE Transactions on Industry Applications*, 54(6), pp. 5791–5801. Available at: https://doi.org/10.1109/tia.2018.2849725.

Flieh, H.M., Slininger, T., Lorenz, R.D., Chien, S.-C. and Ku, L.-H. (2019) ‘Self-Sensing and Power Conversion Comparison for Flux Weakening Surface Mounted Permanent Magnet Servo Motors Designed using Symmetric and Asymmetric Rotors’, *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 267–274. Available at: https://doi.org/10.1109/ecce.2019.8912244.

Frenklach, A., Louie, S., Koop, M.M. and Bronte-Stewart, H. (2008) ‘Excessive postural sway and the risk of falls at different stages of Parkinson’s disease’, *Movement Disorders*, 24(3), pp. 377–385. Available at: https://doi.org/10.1002/mds.22358.

Fu, M. and Ran, Q. (2022) ‘Vector Control Technology of Two-Phase Hybrid Stepping Motor Based on Current Loop PI Regulator’, *2022 4th International Conference on Robotics and Computer Vision (ICRCV)*, pp. 256–260. Available at: https://doi.org/10.1109/icrcv55858.2022.9953224.

Gattinara, F., Menna, G., Mileti, I., Zampogna, A., Asci, F., Paoloni, M., Suppa, A., Del Prete, Z. and Palermo, E. (2022) ‘Machine Learning and Wearable Sensors for the Early Detection of Balance Disorders in Parkinson’s Disease’, *Sensors (Basel, Switzerland)*, 22(24). Available at: https://doi.org/10.3390/s22249903.

Ge, W., Apthorp, D., Lueck, C.J. and Suominen, H. (2019) ‘094 Towards objective testing in parkinson’s disease: a systematic review of the literature looking at assessment of postural sway’, *Journal of Neurology, Neurosurgery & Psychiatry*, 90(e7), pp. A30.2-A30. Available at: https://doi.org/10.1136/jnnp-2019-anzan.82.

Geng, D.X., Zhao, J., Zhang, L. and Zhao, Y.W. (2011) ‘Mechanical Properties Analysis on Elongation Type of Pneumatic Artificial Muscles’, *Applied Mechanics and Materials*, 110-116, pp. 1313–1320. Available at: https://doi.org/10.4028/www.scientific.net/amm.110-116.1313.

Germanotta, M., Mileti, I., Conforti, I., Prete, Z., Aprile, I., & Palermo, E., (2021). Estimation of Human Center of Mass Position through the Inertial Sensors-Based Methods in Postural Tasks: An Accuracy Evaluation. *Sensors (Basel, Switzerland),* 21. https://doi.org/10.3390/s21020601.

Ghamkhar, L. and Kahlaee, A.H. (2019) ‘The effect of trunk muscle fatigue on postural control of upright stance: A systematic review’, *Gait & Posture*, 72, pp. 167–174. Available at: https://doi.org/10.1016/j.gaitpost.2019.06.010.

Giovanelli, Y., Puel, F., Mahdi, C., Gouelle, A. and Bertucci, W. (2022) ‘Comparative evaluation of cervical exoskeletons using IMUs’, *AHFE International* [Preprint]. Available at: https://doi.org/10.54941/ahfe1001483.

Goble, D., & Baweja, H., (2018a). Normative Data for the BTrackS Balance Test of Postural Sway: Results from 16,357 Community-Dwelling Individuals Who Were 5 to 100 Years Old. *Physical Therapy*, 98, pp. 779–785. https://doi.org/10.1093/ptj/pzy062.

Goble, D.J. and Baweja, H.S. (2018b) ‘Postural sway normative data across the adult lifespan: Results from 6280 individuals on the Balance Tracking System balance test’, *Geriatrics & Gerontology International*, 18(8), pp. 1225–1229. Available at: https://doi.org/10.1111/ggi.13452.

Goel, R., Nakagome, S., Paloski, W., Contreras-Vidal, J., & Parikh, P., (2022). Assessment of Biomechanical Predictors of Occurrence of Low-Amplitude N1 Potentials Evoked by Naturally Occurring Postural Instabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineerin*I, 30, pp. 476-485. https://doi.org/10.1109/TNSRE.2022.3154707.

Goldschmidt, E., Angriman, F., Agarwal, N., Trevisan, M., Zhou, J., Chen, K., Gerszten, P.C., Kanter, A.S., Okonkwo, D.O., Passias, P., Scheer, J., Protopsaltis, T., Lafage, V., Lafage, R., Schwab, F., Bess, S., Ames, C., Smith, J.S., Shaffrey, C.I. and Miller, E. (2019) 'A New Piece of the Puzzle to Understand Cervical Sagittal Alignment: Utilizing a Novel Angle δ to Describe the Relationship among T1 Vertebral Body Slope, Cervical Lordosis, and Cervical Sagittal Alignment', *Neurosurgery* [Preprint]. Available at: https://doi.org/10.1093/neuros/nyz088.

Gordon, C.C., 2006. *Anthropometric Data*. [online] Available at: https://multisite.eos.ncsu.edu/www-ergocenter-ncsu-edu/wp-content/uploads/sites/18/2016/06/Anthropometric-Detailed-Data-Tables.pdf.

Grafton, S.T., Ralston, A.B. and Ralston, J.D. (2019) 'Monitoring of postural sway with a head-mounted wearable device: effects of gender, participant state, and concussion', *Medical Devices: Evidence and Research*, Volume 12, pp. 151–164. Available at: https://doi.org/10.2147/mder.s205357.

Groenhuis, V., Rolff, G., Bosman, K., Abelmann, L. and Stramigioli, S. (2021) 'Multi-Axis Electric Stepper Motor', *IEEE Robotics and Automation Letters*, 6(4), pp. 7201–7208. Available at: https://doi.org/10.1109/lra.2021.3097077.

Gu, H., Chen, T., Ma, X., Zhang, M., Sun, Y. and Zhao, J. (2025) 'CLTNet: A Hybrid Deep Learning Model for Motor Imagery Classification', *Brain Sciences*, 15(2), pp. 124–124. Available at: https://doi.org/10.3390/brainsci15020124.

Guan, Q., Sun, J., Liu, Y., Wereley, N.M. and Leng, J. (2020) 'Novel Bending and Helical Extensile/Contractile Pneumatic Artificial Muscles Inspired by Elephant Trunk', *Soft Robotics* [Preprint]. Available at: https://doi.org/10.1089/soro.2019.0079.

Guan, Q., Sun, J., Liu, Y., Wereley, N.M. and Leng, J. (2021) 'Characterization and nonlinear models of bending extensile/contractile pneumatic artificial muscles', *Smart materials and structures*, 30(2), pp. 025024–025024. Available at: https://doi.org/10.1088/1361-665x/abd4b0.

Guo, L., Kou, J. and Wu, M. (2022) 'Ability of Wearable Accelerometers-Based Measures to Assess the Stability of Working Postures', *International Journal of Environmental Research and Public Health*, 19(8), p. 4695. Available at: https://doi.org/10.3390/ijerph19084695.

Hall, S. J., 2012. *Basic biomechanics.* (6th ed.), McGraw-Hill, Boston, MA.

Harshvardhan, R., Tambat, M., & Patil, V., (2015). Stepper Motor: A Review on Theory and Fundamentals. *International journal of emerging trends in science and technology*, 2.

Hill, M.W., Duncan, M.J., Oxford, S.W., Kay, A.D. and Price, M.J. (2018) 'Effects of external loads on postural sway during quiet stance in adults aged 20–80 years', *Applied Ergonomics*, 66, pp. 64–69. Available at: https://doi.org/10.1016/j.apergo.2017.08.007.

Hoang, S., Karydis, K., Brisk, P. and Grover, W.H. (2021) 'A pneumatic random-access memory for controlling soft robots', *PLOS ONE*. Edited by V. Mattoli, 16(7), p. e0254524. Available at: https://doi.org/10.1371/journal.pone.0254524.

Hong-bin, W., Zhe, Z., Xu-hui, C. and Yuan-bin, W. (2017) 'Stepper motor SPWM subdivision control circuit design based on FPGA', *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 889–893. Available at: https://doi.org/10.1109/icis.2017.7960118.

Hsu, S., H., Changcheng, C., Chen, C. T., Wu, Y. C., Lian, W. Y., Li, T. M. and Huang, C. H., (2020). Design and Evaluation of a Wearable Lower Limb Robotic Exoskeleton for Power Assistance. *In: IEEE (Institute of Electrical and Electronics Engineers): International Conference on Systems, Man, and Cybernetics (SMC). Toronto, Canada, 2020*. Taipei: National Taiwan Normal University.

Hu, W. and Alici, G. (2019) 'Bioinspired Three-Dimensional-Printed Helical Soft Pneumatic Actuators and Their Characterization', *Soft Robotics* [Preprint]. Available at: https://doi.org/10.1089/soro.2019.0015.

Hu, X., Chuang, L., Li, H., Jia, L., Song, C., Wang, Z. and Qu, X., (2019). An Ankle Based Soft Active Orthotic Device Powered by Pneumatic Artificial Muscle. *In: IEEE (Institute of Electrical and Electronics Engineers): International Conference on Real-time Computing and Robotics (RCAR). Irkutsk, Russia, 2019.* Shenzhen: Shenzhen University.

Huang, X., Kumar, K., Jawed, M.K., Mohammadi Nasab, A., Ye, Z., Shan, W. and Majidi, C. (2019) 'Highly Dynamic Shape Memory Alloy Actuator for Fast Moving Soft Robots', *Advanced Materials Technologies*, 4(4), p. 1800540. Available at: https://doi.org/10.1002/admt.201800540.

Hybart, R., L. and Ferris, (2022). Preliminary Validation of Proportional Myoelectric Control of A Commercially Available Robotic Ankle Exoskeleton. *In: IEEE (Institute of Electrical and Electronics Engineers): International Conference on Rehabilitation Robotics (ICORR). Rotterdam, Netherlands, 2022.* Gainesville: the University of Florida.

Jafari, H. and Gustafsson, T. (2023) 'Optimal controllers resembling postural sway during upright stance', *PLOS ONE*, 18(5), pp. e0285098–e0285098. Available at: https://doi.org/10.1371/journal.pone.0285098.

Janusz, B., Beck, M., Szczepańska, J., Sadowska, D., Bacik, B., Juras, G., & Słomka, K., (2016). Directional measures of postural sway as predictors of balance instability and accidental falls., 52, pp. 75 - 83. https://doi.org/10.1515/hukin-2015-0195

Jiang, C., Chau, K.T., Lee, C.H.T., Han, W., Liu, W. and Lam, W.H. (2019) 'A Wireless Servo Motor Drive With Bidirectional Motion Capability', *IEEE Transactions on Power Electronics*, 34(12), pp. 12001–12010. Available at: https://doi.org/10.1109/TPEL.2019.2904757.

Junior, A.G.L., de Andrade, R.M. and Filho, A.B. (2016) 'Series Elastic Actuator: Design, Analysis and Comparison', *Recent Advances in Robotic Systems* [Preprint]. Available at: https://doi.org/10.5772/63573.

Kang, S.H. and Mirka, G.A. (2023a) 'Effect of trunk flexion angle and time on lumbar and abdominal muscle activity while wearing a passive back-support exosuit device during simple posture-maintenance tasks', *Ergonomics*, pp. 1–28. Available at: https://doi.org/10.1080/00140139.2023.2191908.

Kang, S.H. and Mirka, G.A. (2023b) 'Effects of a Passive Back-Support Exosuit on Erector Spinae and Abdominal Muscle Activity During Short-Duration, Asymmetric Trunk Posture Maintenance Tasks', *Human factors* [Preprint]. Available at: https://doi.org/10.1177/00187208231197264.

Kiemel, T., Oie, K.S. and Jeka, J.J. (2006) 'Slow Dynamics of Postural Sway Are in the Feedback Loop', *Journal of Neurophysiology*, 95(3), pp. 1410–1418. Available at: https://doi.org/10.1152/jn.01144.2004.

Kim, W., Seo, B., Yu, S.Y. and Cho, K.-J. (2022) 'Deployable Soft Pneumatic Networks (D-PneuNets) Actuator With Dual-Morphing Origami Chambers for High-Compactness', *IEEE Robotics and Automation Letters*, 7(2), pp. 1262–1269. Available at: https://doi.org/10.1109/lra.2021.3137504.

Ko, H-Y., Park, J.H., Shin, Y.B. and Baek, S.Y. (2004) 'Gross quantitative measurements of spinal cord segments in human', *Spinal Cord*, 42(1), pp. 35–40. Available at: https://doi.org/10.1038/sj.sc.3101538.

Kouzaki, M. and Masani, K. (2012) 'Postural sway during quiet standing is related to physiological tremor and muscle volume in young and elderly adults', *Gait & Posture*, 35(1), pp. 11–17. Available at: https://doi.org/10.1016/j.gaitpost.2011.03.028.

Kozinc, Ž., Marjanov, N. and Šarabon, N. (2023) 'The Differences in Transient Characteristics of Postural Control between Young and Older Adults across Four Different Postural Tasks', *Applied Sciences*, 13(6), p. 3485. Available at: https://doi.org/10.3390/app13063485.

Krishnamoorthy, V. and Latash, M.L. (2005) 'Reversals of anticipatory postural adjustments during voluntary sway in humans', *The Journal of Physiology*, 565(2), pp. 675–684. Available at: https://doi.org/10.1113/jphysiol.2005.084772.

Krishnan, R. (1987) ‘Selection Criteria for Servo Motor Drives’, *IEEE Transactions on Industry Applications*, IA-23(2), pp. 270–275. Available at: https://doi.org/10.1109/tia.1987.4504902.

Krishnan, R.H., V. Devanandh, Brahma, A.K. and S. Pugazhenthi (2016) ‘Estimation of mass moment of inertia of human body, when bending forward, for the design of a self-transfer robotic facility’, *Journal of Engineering Science and Technology*, 11(2), pp. 166–176. Available at: https://www.researchgate.net/publication/271292606_Estimation_of_mass_moment_of_inertia_of_human_body_when_bending_forward_for_the_design_of_a_self-transfer_robotic_facility.

Kuber, P.M. and Rashedi, E. (2024) ‘Investigating Spatiotemporal Effects of Back-Support Exoskeletons Using Unloaded Cyclic Trunk Flexion–Extension Task Paradigm’, *Applied Sciences*, 14(13), p. 5564. Available at: https://doi.org/10.3390/app14135564.

Kuukkanen, T.M. and Mälkiä, E.A. (2000) ‘An experimental controlled study on postural sway and therapeutic exercise in subjects with low back pain’, *Clinical Rehabilitation*, 14(2), pp. 192–202. Available at: https://doi.org/10.1191/026921500667300454.

Layne, C.S., Malaya, C.A., Ravindran, A.S., John, I., Francisco, G.E. and Contreras-Vidal, J.L. (2022) ‘Distinct Kinematic and Neuromuscular Activation Strategies During Quiet Stance and in Response to Postural Perturbations in Healthy Individuals Fitted With and Without a Lower-Limb Exoskeleton’, *Frontiers in Human Neuroscience*, 16, p. 942551. Available at: https://doi.org/10.3389/fnhum.2022.942551.

Leach, J., Mancini, M., Peterka, R., Hayes, T. and Horak, F. (2014) ‘Validating and Calibrating the Nintendo Wii Balance Board to Derive Reliable Center of Pressure Measures’, *Sensors*, 14(10), pp. 18244–18267. Available at: https://doi.org/10.3390/s141018244.

Legrand, T., Mongold, S.J., Muller, L., Naeije, G., Ghinst, M.V. and Bourguignon, M. (2024) ‘Cortical tracking of postural sways during standing balance’, *Scientific Reports*, 14(1). Available at: https://doi.org/10.1038/s41598-024-81865-2.

Li, Y., Chen, Y., Ren, T., Li, Y. and Choi, S. hong (2018) ‘Precharged Pneumatic Soft Actuators and Their Applications to Untethered Soft Robots’, *Soft Robotics*, 5(5), pp. 567–575. Available at: https://doi.org/10.1089/soro.2017.0090.

Liang, H.-W., Chi, S.-Y., Chang, B.-Y. and Hwang, Y.-H. (2020) ‘Reliability and validity of a virtual reality-based system for evaluating postural stability’, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, pp. 1–1. Available at: https://doi.org/10.1109/tnsre.2020.3034876.

Liu, Z., Ma, W., Ju, W. and Hu, J., 2020, *Toward Future ATDs: The Human Body Dimensions and Weight for a Chinese Mid-Size Adult Male* [Online]. Available at: https://www.ircobi.org/wordpress/downloads/irc20-asia/pdf-files/2039a.pdf

Lu, J., Haninger, K., Chen, W. and Masayoshi Tomizuka (2015) ‘Design and torque-mode control of a cable-driven rotary series elastic actuator for subject-robot interaction’. Available at: https://doi.org/10.1109/aim.2015.7222525.

Luder, T., Meier, M., Neuweiler, R. and Lambercy, O. (2025) ‘Evaluation of the support provided by a soft passive exoskeleton in individuals with back pain’, *Applied Ergonomics*, 127, p. 104514. Available at: https://doi.org/10.1016/j.apergo.2025.104514.

Malaya, C.A., Haworth, J., Pohlman, K.A., Powell, C. and Smith, D.L. (2020) ‘Impact of Extremity Manipulation on Postural Sway Characteristics: A Preliminary, Randomized Crossover Study’, *Journal of Manipulative and Physiological Therapeutics* [Preprint]. Available at: https://doi.org/10.1016/j.jmpt.2019.02.014.

Martino Cinnera, A., Picerno, P., Bisirri, A., Koch, G., Morone, G. and Vannozzi, G. (2023) ‘Upper limb assessment with inertial measurement units according to the international classification of functioning in stroke: a systematic review and correlation meta-analysis’, *Topics in Stroke Rehabilitation*, pp. 1–20. Available at: https://doi.org/10.1080/10749357.2023.2197278.

Matheron, E., Dubost, V., Mourey, F., Pfitzenmeyer, P. and Manckoundia, P. (2010) ‘Analysis of postural control in elderly subjects suffering from Psychomotor Disadaptation Syndrome (PDS)’, *Archives of Gerontology and Geriatrics*, 51(1), pp. e19-23. Available at: https://doi.org/10.1016/j.archger.2009.07.003.

Mattia Guidolin, Andrei, R., Roberto, O., Reggiani, M., Emanuele Menegatti and Luca Tagliapietra (2021) ‘On the Accuracy of IMUs for Human Motion Tracking: a Comparative Evaluation’, *2021 IEEE International Conference on Mechatronics (ICM)* [Preprint]. Available at: https://doi.org/10.1109/icm46511.2021.9385684.

Maurer, C., & Peterka, R., (2005). A new interpretation of spontaneous sway measures based on a simple model of human postural control.. *Journal of neurophysiology*, 93 1, pp. 189-200 . https://doi.org/10.1152/JN.00221.2004.

Mazumder, O., Tripathy, S., Roy, S., Chakravarty, K., Chatterjee, D., & Sinha, A., (2017). Postural sway based geriatric fall risk assessment using kinect. 2017 *IEEE SENSORS*, pp. 1-3. https://doi.org/10.1109/ICSENS.2017.8234214.

Meijneke, C., van Oort, G., Sluiter, V., van Asseldonk, E., Tagliamonte, N.L., Tamburella, F., Pisotta, I., Masciullo, M., Arquilla, M., Molinari, M., Wu, A.R., Dzeladini, F., Ijspeert, A.J. and van der Kooij, H. (2021) ‘Symbitron Exoskeleton: Design, Control, and Evaluation of a Modular Exoskeleton for Incomplete and Complete Spinal Cord Injured Individuals’, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29, pp. 330–339. Available at: https://doi.org/10.1109/tnsre.2021.3049960.

Mendoza, F., Durango, D., Pallo, G. and Merchan, E. (2023) ‘Advances in exoskeletons for military use’, *Athenea*, pp. 43–54. Available at: https://doi.org/10.47460/10.47460/athenea.v4i12.57.

Menga, G. and Ghirardi, M. (2018) ‘Lower Limb Exoskeleton for Rehabilitation with Improved Postural Equilibrium’, *Robotics*, 7(2), p. 28. Available at: https://doi.org/10.3390/robotics7020028.

Mesci, E., İçağasioğlu, A., Mesci, N. and Madenci, E. (2016) ‘Effects Of Chronic Low Back Pain On Postural Stability In The Elderly’, *Turkish Journal of Geriatrics*, 19(2). Available at: http://geriatri.dergisi.org/abstract.php?id=932 (Accessed: 12 May 2025).

Meyer, B., Cohen, J., Donahue, N., Fox, S., O’Leary, A., Brown, A., Leahy, C., VanDyk, T., DePetrillo, P., Ceruolo, M., Cheney, N., Solomon, A., & Mcginnis, R., (2023). Chest-Based Wearables and Individualized Distributions for Assessing Postural Sway in Persons With Multiple Sclerosis. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 31, pp. 2132 - 2139. https://doi.org/10.1109/TNSRE.2023.3267807.

Mikkonen, J., Leinonen, V., Kaski, D., Hartvigsen, J., Luomajoki, H., Selander, T. and Airaksinen, O. (2022) ‘Postural sway does not differentiate individuals with chronic low back pain, single and multisite chronic musculoskeletal pain, or pain free controls: a cross-sectional study of 229 subjects’, *The Spine Journal* [Preprint]. Available at: https://doi.org/10.1016/j.spinee.2022.04.013.

Mirzaee, A., Moghadam, M.M. and Saba, A.M. (2019) ‘Conceptual Design of an Active Body Weight Support System Using a Linear Series Elastic Actuator’, pp. 80–87. Available at: https://doi.org/10.1109/icrom48714.2019.9071896.

Mitra, S., Knight, A. and Munn, A. (2013) ‘Divergent effects of cognitive load on quiet stance and task-linked postural coordination.’, 39(2), pp. 323–328. Available at: https://doi.org/10.1037/a0030588.

Miyashita, T., Cote, C., Terrone, D., & Diakogeorgiou, E., (2020). Detecting changes in postural sway.. *Journal of biomechanics*, 107, pp. 109868 . https://doi.org/10.1016/j.jbiomech.2020.109868.

Mohr, M., Schön, T., von Tscharner, V. and Nigg, B.M. (2018) ‘Intermuscular Coherence Between Surface EMG Signals Is Higher for Monopolar Compared to Bipolar Electrode Configurations’, *Frontiers in Physiology*, 9. Available at: https://doi.org/10.3389/fphys.2018.00566.

Mosadegh, B., Polygerinos, P., Keplinger, C., Wennstedt, S., Shepherd, R.F., Gupta, U., Shim, J., Bertoldi, K., Walsh, C.J. and Whitesides, G.M. (2014) ‘Pneumatic Networks for Soft Robotics that Actuate Rapidly’, *Advanced Functional Materials*, 24(15), pp. 2163–2170. Available at: https://doi.org/10.1002/adfm.201303288.

Mushtaq, F., Torlakcik, H., Hoop, M., Jang, B., Carlson, F., Grunow, T., Läubli, N., Ferreira, A., Chen, X., Nelson, B.J. and Pané, S. (2019) ‘Motile Piezoelectric Nanoeels for Targeted Drug Delivery’, *Advanced Functional Materials*, 29(12), p. 1808135. Available at: https://doi.org/10.1002/adfm.201808135.

Nagymáté, G. and Kiss, R.M. (2016) ‘Parameter Reduction in the Frequency Analysis of Center of Pressure in Stabilometry’, *Periodica Polytechnica Mechanical Engineering*, 60(4), pp. 238–246. Available at: https://doi.org/10.3311/ppme.8999.

Najafi, B., Horn, D., Marclay, S., Crews, R., Wu, S., & Wrobel, J., (2010). Assessing Postural Control and Postural Control Strategy in Diabetes Patients Using Innovative and Wearable Technology. *Journal of Diabetes Science and Technology*, 4, pp. 780 - 791. https://doi.org/10.1177/193229681000400403.

Nakao, T., Masuda, K., Kanai, S., Tsujita, J., Hirakawa, K. and Okada, S. (2017) ‘Lower trunk muscle activity-induced alignment and cop position during single-leg standing’, *Journal of Physical Therapy Science*, 29(6), pp. 1057–1061. Available at: https://doi.org/10.1589/jpts.29.1057.

Nehary, E.A., Rajan, S. and Ando, B. (2024) ‘Postural Sway Classification Using Triaxial Accelerometer Signals’, *IEEE Sensors Letters*, 8(5), pp. 1–4. Available at: https://doi.org/10.1109/lsens.2024.3381010.

Neville, C., Ludlow, C. and Rieger, B., (2015) ‘Measuring postural stability with an inertial sensor: validity and sensitivity’, *Medical Devices: Evidence and Research*, 8, pp. 447–455. doi: 10.2147/MDER.S91719.

Ohlendorf, D., Doerry, C., Fisch, V., Schamberger, S., Erbe, C., Wanke, E.M. and Groneberg, D.A. (2019) ‘Standard reference values of the postural control in healthy young female adults in Germany: an observational study’, *BMJ Open*, 9(6), p. e026833. Available at: https://doi.org/10.1136/bmjopen-2018-026833.

Özdemir, A. and Barshan, B. (2014) ‘Detecting Falls with Wearable Sensors Using Machine Learning Techniques’, *Sensors*, 14(6), pp. 10691–10708. Available at: https://doi.org/10.3390/s140610691.

Paillard, T. and Noé, F. (2015) ‘Techniques and Methods for Testing the Postural Function in Healthy and Pathological Subjects’, *BioMed Research International*, 2015, pp. 1–15. Available at: https://doi.org/10.1155/2015/891390.

Pardoel, S. and Doumit, M. (2019) ‘Development and testing of a passive ankle exoskeleton’, *Biocybernetics and Biomedical Engineering*, 39(3), pp. 902–913. Available at: https://doi.org/10.1016/j.bbe.2019.08.007.

Park, J.-H., Kim, S., Nussbaum, M.A. and Srinivasan, D. (2019) ‘Effects of Back Support Exoskeleton Use on Postural Stability’, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 63(1), pp. 1088–1089. Available at: https://doi.org/10.1177/1071181319631485.

Park, J.-H., Kim, S., Nussbaum, M.A. and Srinivasan, D. (2021) 'Effects of two passive back-support exoskeletons on postural balance during quiet stance and functional limits of stability', *Journal of Electromyography and Kinesiology*, 57, p. 102516. Available at: https://doi.org/10.1016/j.jelekin.2021.102516.

Park, Y.-L., Chen, B., Pérez-Arancibia, N.O., Young, D., Stirling, L., Wood, R.J., Goldfield, E.C. and Nagpal, R. (2014) 'Design and control of a bio-inspired soft wearable robotic device for ankle–foot rehabilitation', *Bioinspiration & Biomimetics*, 9(1), p. 016007. Available at: https://doi.org/10.1088/1748-3182/9/1/016007.

Paterno, M.V., Schmitt, L.C., Ford, K.R., Rauh, M.J. and Hewett, T.E. (2013) 'Altered postural sway persists after anterior cruciate ligament reconstruction and return to sport', *Gait & Posture*, 38(1), pp. 136–140. Available at: https://doi.org/10.1016/j.gaitpost.2012.11.001.

Philipp Floessel, Hammerschmidt, F., Koltermann, J.J., Foerster, J., Beck, H., Disch, A.C. and Datzmann, T. (2024) 'Comparison of Measurements for Recording Postural Control in Standing and Seated Position in Healthy Individuals', *Journal of Functional Morphology and Kinesiology*, 9(4), pp. 178–178. Available at: https://doi.org/10.3390/jfmk9040178.

Poliero, T., Fanti, V., Sposito, M., Caldwell, D.G. and Natali, C.D. (2022) 'Active and Passive Back-Support Exoskeletons: A Comparison in Static and Dynamic Tasks', *IEEE Robotics and Automation Letters*, 7(3), pp. 8463–8470. Available at: https://doi.org/10.1109/lra.2022.3188439.

Poliero, T., Lazzaroni, M., Toxiri, S., Di Natali, C., Caldwell, D.G. and Ortiz, J. (2020) 'Applicability of an Active Back-Support Exoskeleton to Carrying Activities', *Frontiers in Robotics and AI*, 7. Available at: https://doi.org/10.3389/frobt.2020.579963.

Pollind, M. and Soangra, R. (2020a) 'Development and Validation of Wearable Inertial Sensor System for Postural Sway Analysis', *Measurement*, 165, p. 108101. Available at: https://doi.org/10.1016/j.measurement.2020.108101.

Pollind, M.L. and Soangra, R. (2020b) 'Mini-Logger- A Wearable Inertial Measurement Unit (IMU) for Postural Sway Analysis', *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 4600–4603. Available at: https://doi.org/10.1109/embc44109.2020.9175167.

Prisco, G., Pirozzi, M.A., Santone, A., Cesarelli, M., Esposito, F., Gargiulo, P., Amato, F. and Donisi, L. (2025) 'Combining Postural Sway Parameters and Machine Learning to Assess Biomechanical Risk Associated with Load-Lifting Activities', *Diagnostics*, 15(1), p. 105. Available at: https://doi.org/10.3390/diagnostics15010105.

Ramachandran, Sivakumar & Yegnaswamy, Roopa. (2010). Measurement of Postural Sway with a Sway Meter-an Analysis. Journal of Physical Therapy. 2.

Ramdani, S., Seigle, B., Lagarde, J., Bouchara, F., & Bernard, P., (2009). On the use of sample entropy to analyze human postural sway data.. *Medical engineering & physics*, 31 8, pp. 1023-31 . https://doi.org/10.1016/j.medengphy.2009.06.004.

Renesas (2024) *What are Brushless DC Motors | Renesas*, *www.renesas.com*. Available at: https://www.renesas.com/us/en/support/engineer-school/brushless-dc-motor-01-overview.

Rezaeipour, M. (2018) 'Evaluation of postural stability in overweight and obese middle-aged men', *TURKISH JOURNAL OF MEDICAL SCIENCES*, 48(5), pp. 1053–1057. Available at: https://doi.org/10.3906/sag-1709-108.

Riach, C.L. and Hayes, K.C. (2008) 'Maturation Of Postural Sway In Young Children', *Developmental Medicine & Child Neurology*, 29(5), pp. 650–658. Available at: https://doi.org/10.1111/j.1469-8749.1987.tb08507.x.

Ringheim, I., Austein, H., Indahl, A. and Roeleveld, K. (2015) 'Postural strategy and trunk muscle activation during prolonged standing in chronic low back pain patients', *Gait & Posture*, 42(4), pp. 584–589. Available at: https://doi.org/10.1016/j.gaitpost.2015.09.008.

RS Malaysia (2020) *Festo Pneumatic Cylinder - 19232, 16mm Bore, 100mm Stroke, DSNU Series, Double Acting*, *Rs-online.com*. Available at: https://my.rs-online.com/web/p/pneumatic-piston-rod-cylinders/1214710. (Accessed: 12 May 2025).

Ruhe, A., Fejer, R. and Walker, B. (2013) 'On the relationship between pain intensity and postural sway in patients with non-specific neck pain', *Journal of Back and Musculoskeletal Rehabilitation*, 26(4), pp. 401–409. Available at: https://doi.org/10.3233/bmr-130399.

Saffer, M., Kiemel, T. and Jeka, J. (2007) 'Coherence analysis of muscle activity during quiet stance', *Experimental Brain Research*, 185(2), pp. 215–226. Available at: https://doi.org/10.1007/s00221-007-1145-3.

Saga, N., Shimada, K., Inamori, D., Saito, N., Satoh, T. and Nagase, J. (2022) 'Smart Pneumatic Artificial Muscle Using a Bend Sensor like a Human Muscle with a Muscle Spindle', *Sensors*, 22(22), p. 8975. Available at: https://doi.org/10.3390/s22228975.

Sambyal, P., Mahato, M., Taseer, A.K., Yoo, H., Garai, M., Van Hiep Nguyen, Syed Sheraz Ali and Oh, I. (2023) 'Magnetically and Electrically Responsive Soft Actuator Derived from Ferromagnetic Bimetallic Organic Framework', *Small*, 19(23). Available at: https://doi.org/10.1002/smll.202207140.

Schniepp, R., Wuehr, M., Pradhan, C., Novozhilov, S., Krafczyk, S., Brandt, T. and Jahn, K. (2013) 'Nonlinear Variability of Body Sway in Patients with Phobic Postural Vertigo', *Frontiers in Neurology*, 4. Available at: https://doi.org/10.3389/fneur.2013.00115.

Schubert, P. and Kirchner, M. (2014) ‘Ellipse area calculations and their applicability in posturography’, *Gait & Posture*, 39(1), pp. 518–522. Available at: https://doi.org/10.1016/j.gaitpost.2013.09.001.

Schumann, T., Redfern, M., Furman, J., El-Jaroudi, A., & Chaparro, L., (1995). Time-frequency analysis of postural sway.. *Journal of Biomechanics,* 28, pp. 603-607. https://doi.org/10.1016/0021-9290(94)00113-I.

Sezer, O. and Ferdjallah, M. (2005) ‘Adaptive autoregressive model for the analysis of center of pressure in healthy subjects during quiet standing’, *48th Midwest Symposium on Circuits and Systems, 2005.*, pp. 495–498 Vol. 1. Available at: https://doi.org/10.1109/mwscas.2005.1594146.

Shea, M., Edwards, W.T., White, A.A. and Hayes, W.C. (1991) ‘Variations of stiffness and strength along the human cervical spine’, *Journal of Biomechanics*, 24(2), pp. 95–107. Available at: https://doi.org/10.1016/0021-9290(91)90354-p.

Shu, Y., Chen, Y., Zhang, X., Zhang, S., Chen, G., Ye, J. and Li, X. (2023) ‘Two-Stage Trajectory-Tracking Control of Cable-Driven Upper-Limb Exoskeleton Robots with Series Elastic Actuators: A Simple, Accurate, and Force-Sensorless Method’, *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* [Preprint]. Available at: https://doi.org/10.1109/iros55552.2023.10342056.

Singh, N.B., Taylor, W.R., Madigan, M.L. and Nussbaum, M.A. (2012) ‘The spectral content of postural sway during quiet stance: Influences of age, vision and somatosensory inputs’, *Journal of Electromyography and Kinesiology*, 22(1), pp. 131–136. Available at: https://doi.org/10.1016/j.jelekin.2011.10.007.

Sipko, T., Glibowski, E. and Kuczyński, M. (2021) ‘Acute effects of proprioceptive neuromuscular facilitation exercises on the postural strategy in patients with chronic low back pain’, *Complementary Therapies in Clinical Practice*, 44, p. 101439. Available at: https://doi.org/10.1016/j.ctcp.2021.101439.

Slunecko, M. and Csapo, R. (2024) ‘Assessment of postural sway with accelerometers: A comparison with force plate measures’, *Current Issues in Sport Science (CISS)*, 9(4), p. 058. Available at: https://doi.org/10.36950/2024.4ciss058.

Smith, M.D., Chang, A.T. and Hodges, P.W. (2016) ‘Balance recovery is compromised and trunk muscle activity is increased in chronic obstructive pulmonary disease’, *Gait & Posture*, 43, pp. 101–107. Available at: https://doi.org/10.1016/j.gaitpost.2015.09.004.

Soft Robotics Toolkit (2011a) *PneuNets Bending Actuators*, *Softroboticstoolkit.com*. Available at: https://softroboticstoolkit.com/book/pneunets-bending-actuator.

Soft Robotics Toolkit (2011b) *PneuNets Bending Actuators*, *Softroboticstoolkit.com*. Available at: https://softroboticstoolkit.com/book/pneunets-bending-actuator.

Solomon, A.J., Jacobs, J.V., Lomond, K.V. and Henry, S.M. (2015) 'Detection of postural sway abnormalities by wireless inertial sensors in minimally disabled patients with multiple sclerosis: a case–control study', *Journal of NeuroEngineering and Rehabilitation*, 12(1). Available at: https://doi.org/10.1186/s12984-015-0066-9.

Sturnieks, D., Arnold, R., & Lord, S., (2011). Validity and reliability of the Swaymeter device for measuring postural sway. *BMC Geriatrics*, 11, pp. 63 - 63. https://doi.org/10.1186/1471-2318-11-63.

Sun, L., Li, X., Chen, L., Shi, H. and Jiang, Z. (2023) 'Dual-Motor Coordination for High-Quality Servo With Transmission Backlash', *IEEE Transactions on Industrial Electronics*, 70(2), pp. 1182–1196. Available at: https://doi.org/10.1109/tie.2022.3156154.

Sundaram, B., Doshi, M. and Pandian, J.S. (2012) 'Postural Stability during Seven Different Standing Tasks in Persons with Chronic Low Back Pain - A Cross-sectional Study', *Indian Journal of Physiotherapy & Occupational Therapy-An International Journal*, 6(2), pp. 22–27. Available at: https://www.i-scholar.in/index.php/ijpot/article/view/47177 (Accessed: 10 May 2025).

Sung, P.S. and Lee, D. (2024a) 'Assessing postural stability in flatfoot using a time-in-boundary method during single-leg standing', *Journal of Orthopaedic Research*, 43(2), pp. 379–387. Available at: https://doi.org/10.1002/jor.25987.

Sung, P.S. and Lee, D. (2024b) 'Postural control and trunk stability on sway parameters in adults with and without chronic low back pain', *European Spine Journal*, 33(4), pp. 1455–1464. Available at: https://doi.org/10.1007/s00586-024-08147-3.

Sung, P.S., Rowland, P. and Lee, D. (2024) 'Implications for fall efficacy strategies on center of pressure and center of gravity sway distances in adults with chronic low back pain', *European Spine Journal*, 33(12), pp. 4581–4590. Available at: https://doi.org/10.1007/s00586-024-08523-z.

Suttanon, P., Khanphed, W., Saadprai, S. and Apibantaweesakul, S. (2024) 'Feasibility and preliminary evidence of the immediate effect on balance, functionality, and cognition in adults over 50 through postural sway-meter training', *PLOS ONE*. Edited by R.S. Melo, 19(12), p. e0314357. Available at: https://doi.org/10.1371/journal.pone.0314357.

Takashima, K., Toshiro Noritsugu, Rossiter, J., Guo, S. and Mukai, T. (2012) 'Curved Type Pneumatic Artificial Rubber Muscle Using Shape-Memory Polymer', *Journal of Robotics and Mechatronics*, 24(3), pp. 472–479. Available at: https://doi.org/10.20965/jrm.2012.p0472.

Takeda, K., Mani, H., Hasegawa, N., Sato, Y., Tanaka, S., Maejima, H. and Asaka, T. (2017) ‘Adaptation effects in static postural control by providing simultaneous visual feedback of center of pressure and center of gravity’, *Journal of Physiological Anthropology*, 36(1). Available at: https://doi.org/10.1186/s40101-017-0147-5.

Tang, Z, Shi, D., Peng, Z., He, L. and Pei, Z., (2013). Electro-hydraulic servo system for Human Lower-limb Exoskeleton based on sliding mode variable structure control. *In: IEEE (Institute of Electrical and Electronics Engineers): International Conference on Information and Automation (ICIA). Yinchuan, China, 2013*. Beijing: Beihang University.

Thomas, A., Kévin, L. and Bérenger, L.T. (2022) ‘Effects of a passive back assistance exoskeleton for load carrying and trunk bending tasks’. Available at: https://doi.org/10.21203/rs.3.rs-2324739/v1.

Tipton, N., Alderink, G. and Rhodes, S. (2023) ‘Approximate Entropy and Velocity of Center of Pressure to Determine Postural Stability: A Pilot Study’, *Applied Sciences*, 13(16), pp. 9259–9259. Available at: https://doi.org/10.3390/app13169259.

Turan, M.Ş. and Barshan, B. (2021) ‘Classification of fall directions via wearable motion sensors’, *Digital Signal Processing*, p. 103129. Available at: https://doi.org/10.1016/j.dsp.2021.103129.

van Dieën, J.H., Koppes, L.L.J. and Twisk, J.W.R. (2010) ‘Postural sway parameters in seated balancing; their reliability and relationship with balancing performance’, *Gait & Posture*, 31(1), pp. 42–46. Available at: https://doi.org/10.1016/j.gaitpost.2009.08.242.

Voss, S., Zampieri, C., Biskis, A., Armijo, N., Purcell, N., Ouyang, B., Liu, Y., Berry-Kravis, E. and O’Keefe, J.A. (2021) ‘Normative database of postural sway measures using inertial sensors in typically developing children and young adults’, *Gait & Posture*, 90, pp. 112–119. Available at: https://doi.org/10.1016/j.gaitpost.2021.07.014.

Walsh, M., Church, C., Hoffmeister, A., Smith, D., & Haworth, J., (2020). Validation of a Portable Force Plate for Evaluating Postural Sway. *Perceptual and Motor Skills*, 128, pp. 191 - 199. https://doi.org/10.1177/0031512520945092.

Wang, Z., Zi, B., Wang, D. and Qian, J. (2019) ‘Design, Modeling and Analysis of a Novel Backdrivable Cable-driven Series Elastic Actuator’, *2019 IEEE 19th International Conference on Nanotechnology (IEEE-NANO)*, pp. 179–183. Available at: https://doi.org/10.1109/nano46743.2019.8993930.

Warnica, M.J., Weaver, T.B., Prentice, S.D. and Laing, A.C. (2014) ‘The influence of ankle muscle activation on postural sway during quiet stance’, *Gait & Posture*, 39(4), pp. 1115–1121. Available at: https://doi.org/10.1016/j.gaitpost.2014.01.019.

Wood, J., Killingly, C., Elliott, D., Anstey, K., & Black, A., (2022). Visual Predictors of Postural Sway in Older Adults. *Translational Vision Science & Technology*, 11. https://doi.org/10.1167/tvst.11.8.24.

Xiao, W., Hu, D., Chen, W., Yang, G. and Han, X. (2021) 'Modeling and analysis of bending pneumatic artificial muscle with multi-degree of freedom', *Smart Materials and Structures*, 30(9), p. 095018. Available at: https://doi.org/10.1088/1361-665x/ac1939.

Yu, N., Zou, W. and Sun, Y. (2018) 'Passivity guaranteed stiffness control with multiple frequency band specifications for a cable-driven series elastic actuator', *Mechanical Systems and Signal Processing*, 117, pp. 709–722. Available at: https://doi.org/10.1016/j.ymssp.2018.08.007.

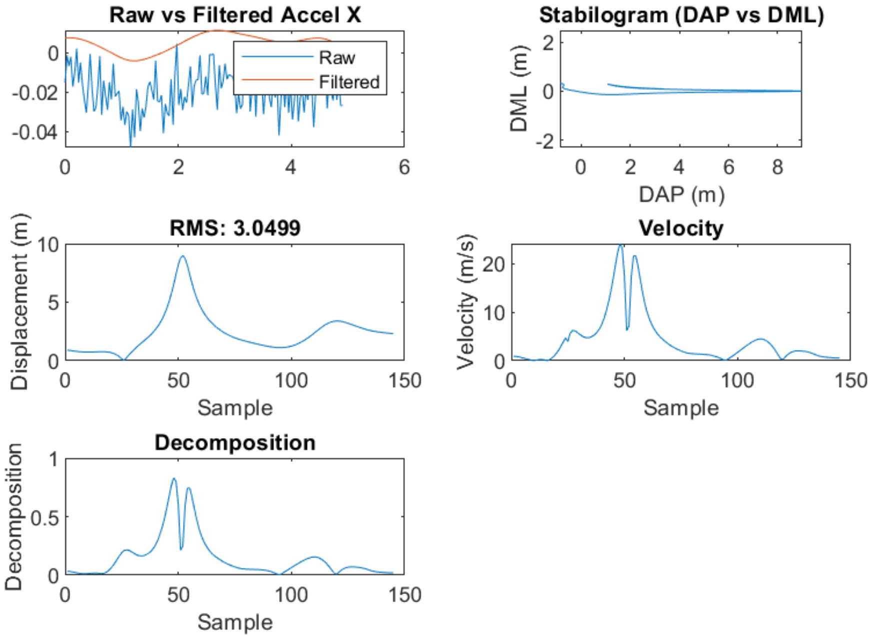
Zhang, Q., Sun, D., Qian, W., Xiao, X. and Guo, Z. (2020) 'Modeling and Control of a Cable-Driven Rotary Series Elastic Actuator for an Upper Limb Rehabilitation Robot', *Frontiers in neurorobotics*, 14. Available at: https://doi.org/10.3389/fnbot.2020.00013.

Zou, W., Tan, W., Yang, Z. and Yu, N. (2016) 'Torque control of a cable-driven series elastic actuator using the 2-DOF method', *2016 35th Chinese Control Conference (CCC)*, pp. 6239–6243. Available at: https://doi.org/10.1109/chicc.2016.7554337.

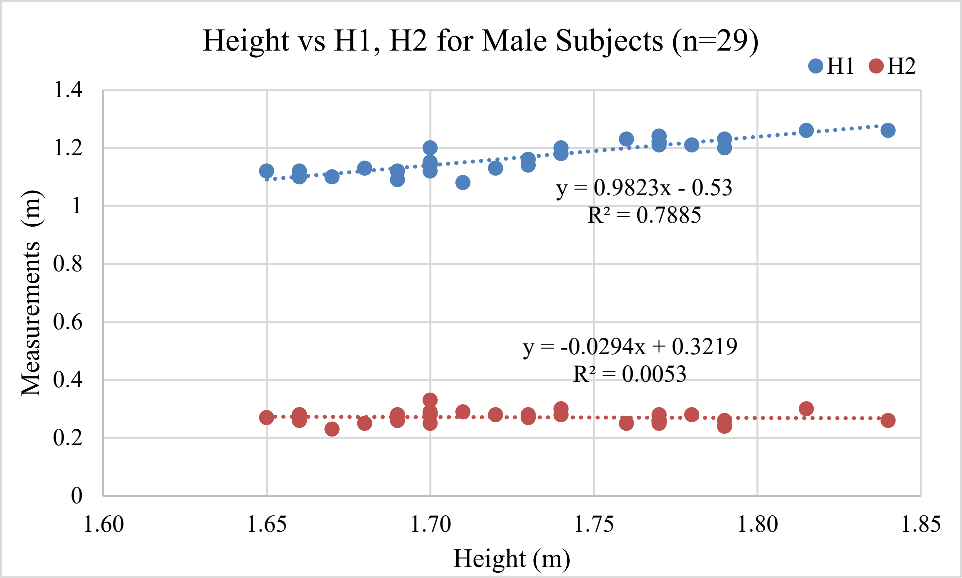
Zou, W., Yang, Z., Tan, W., Wang, M., Liu, J. and Yu, N. (2016) 'Impedance control of a cable-driven series elastic actuator with the 2-DOF control structure', *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3347–3352. Available at: https://doi.org/10.1109/iros.2016.7759515.

APPENDICES

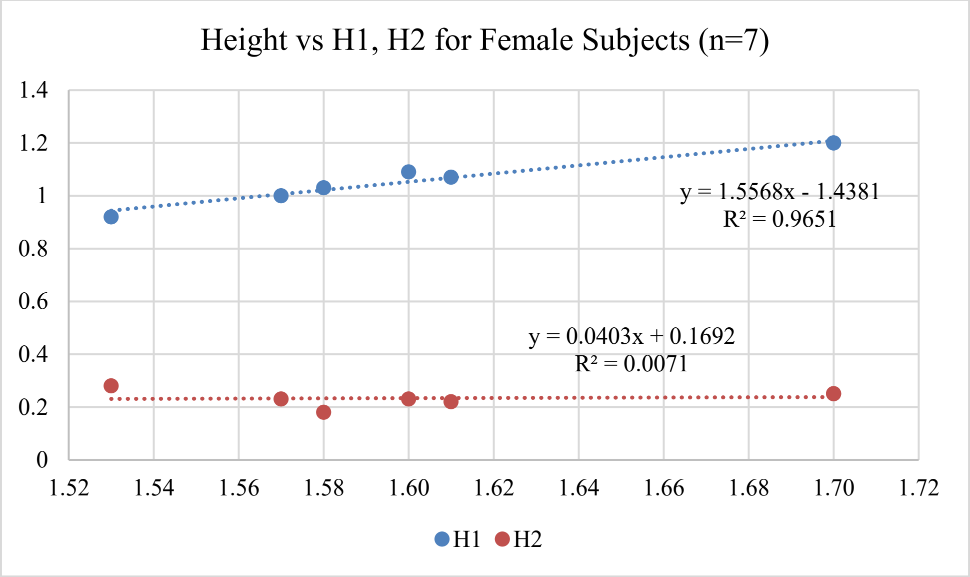
Appendix A: Graphs



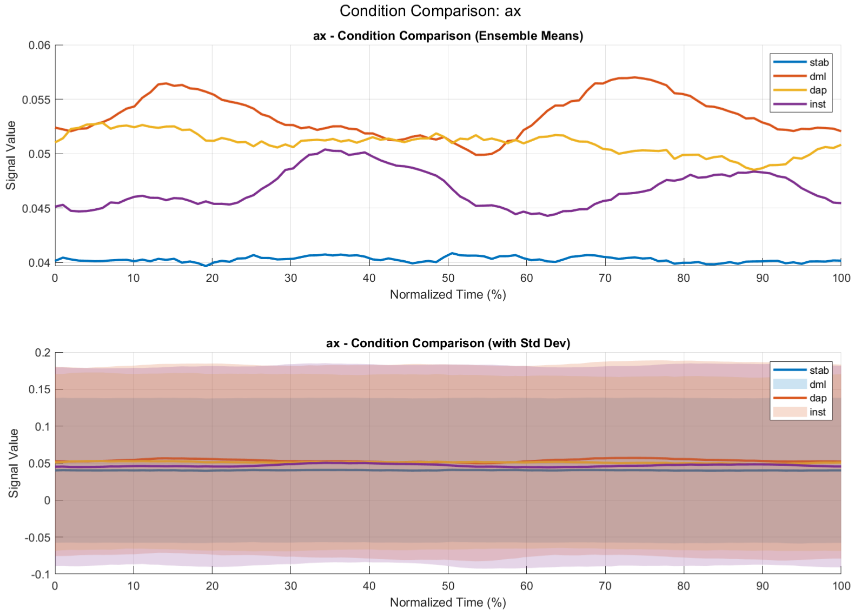
Graph A-1: Sample of Processed Time Series Data for Algorithm Training



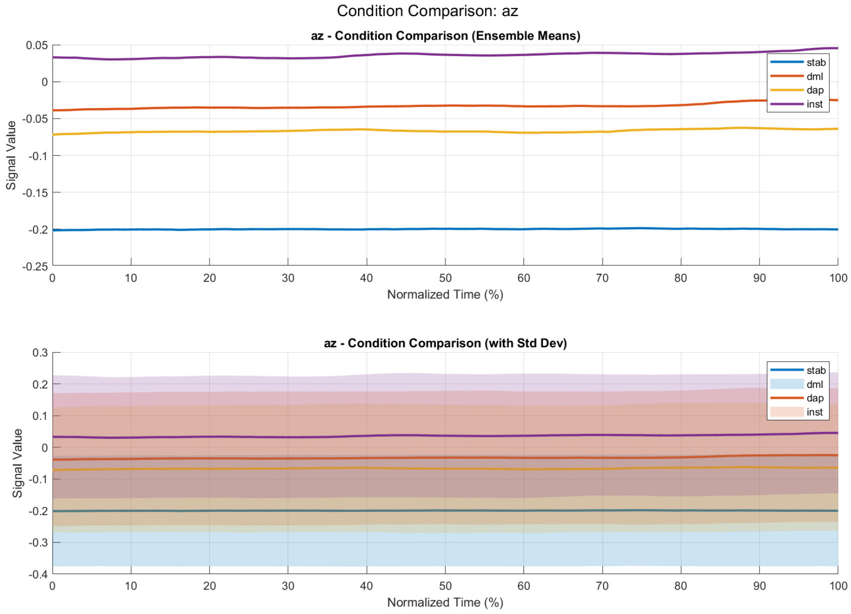
Graph A-2: Height vs H1 and H2 Relationship Plot for Male Subjects



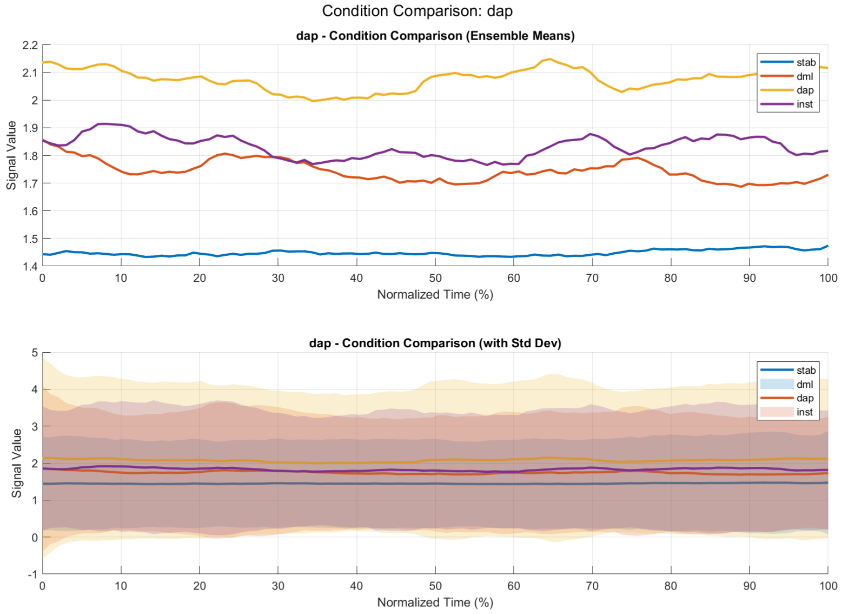
Graph A-3: Height vs H1 and H2 Relationship Plot for Female Subjects



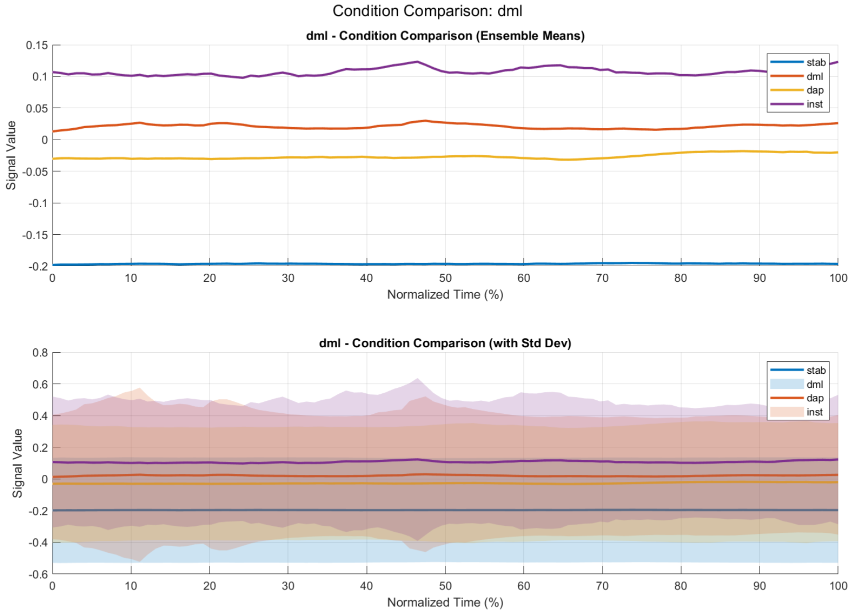
Graph A-4: Ensemble Means Plot of ax Across Conditions Showing Clear Distinction in Means and Standard Deviations Prompting Trial of Empirical-Based-Threshold Attempt Using ax



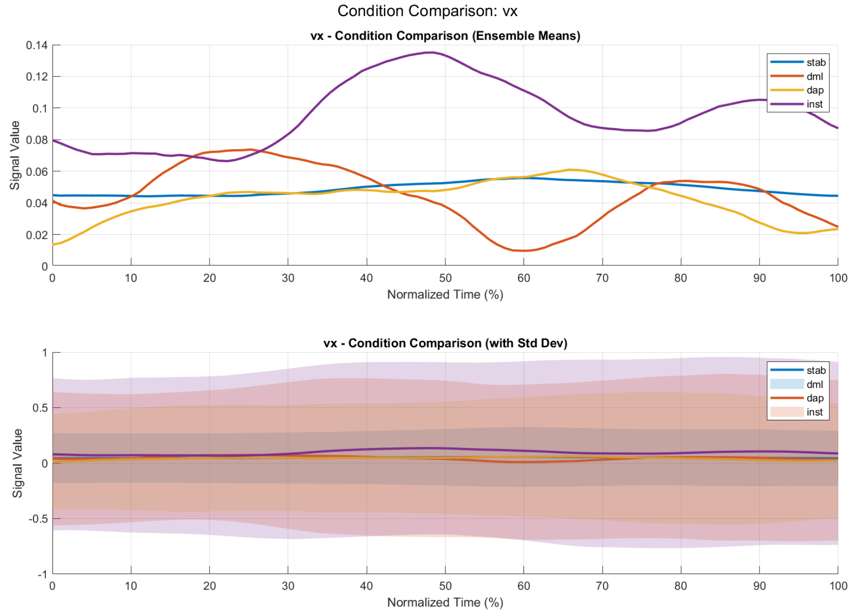
Graph A-5: Ensemble Means Plot of az Across Conditions Showing Clear Distinction in Means and Standard Deviations Prompting Trial of Empirical-Based-Threshold Attempt Using az



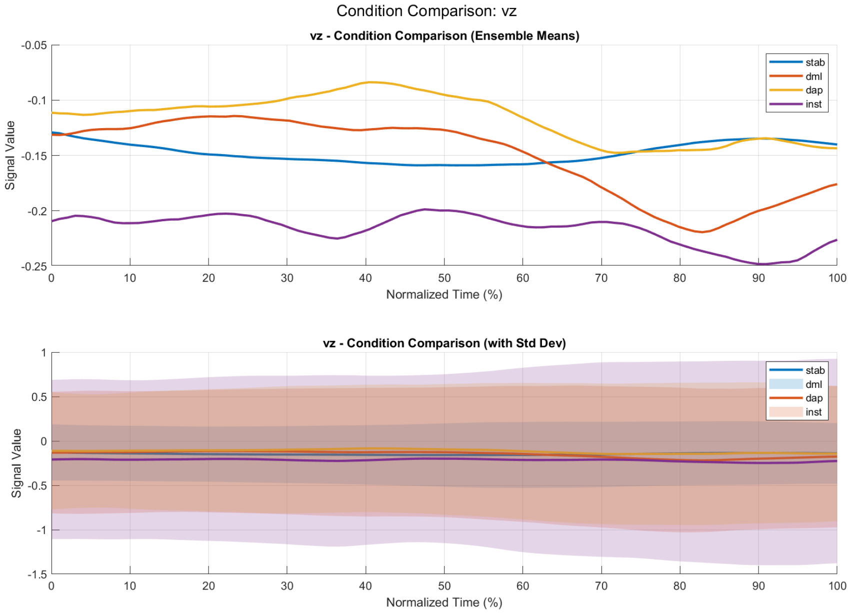
Graph A-6: Ensemble Means Plot of DAP Across Conditions Showing Clear Distinction in Means and Standard Deviations Prompting Trial of Empirical-Based-Threshold Attempt Using DAP



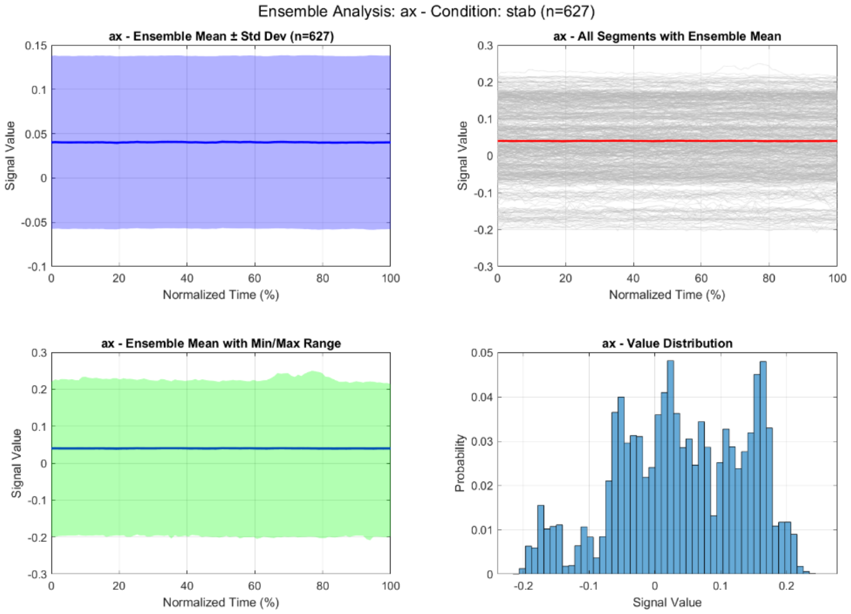
Graph A-7: Ensemble Means Plot of DML Across Conditions Showing Clear Distinction in Means and Standard Deviations Prompting Trial of Empirical-Based-Threshold Attempt Using DML



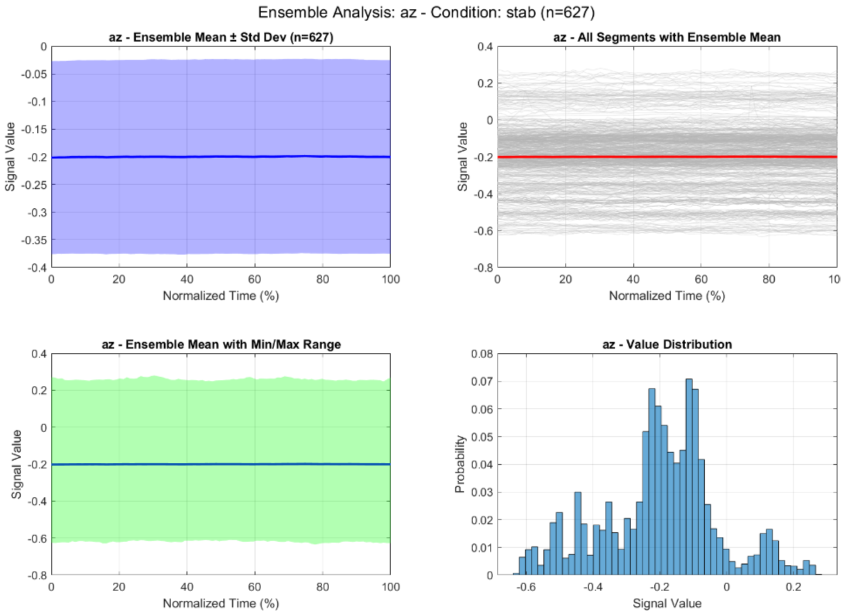
Graph A-8: Ensemble Means Plot of Vx Across Conditions Showing Clear Distinction in Means and Standard Deviations Prompting Trial of Empirical-Based-Threshold Attempt Using Vx



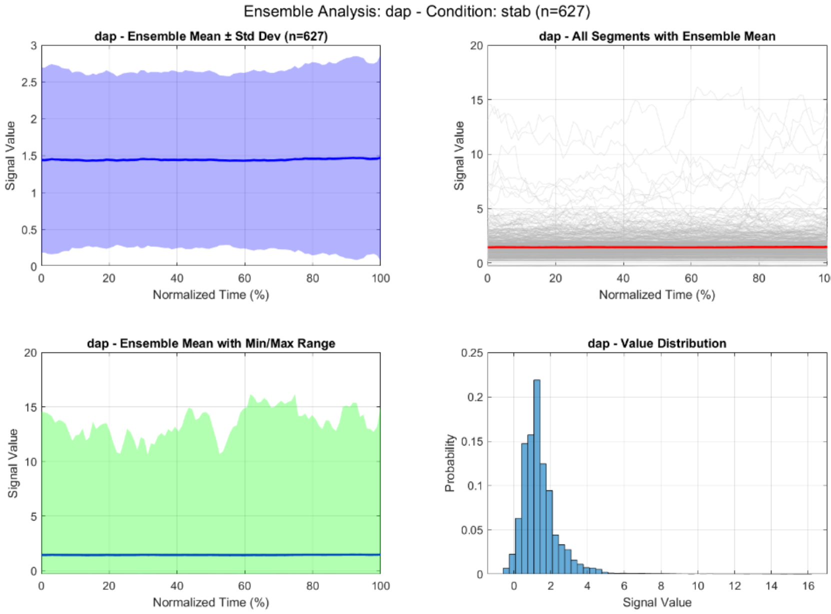
Graph A-9: Ensemble Means Plot of Vz Across Conditions Showing Clear Distinction in Means and Standard Deviations Prompting Trial of Empirical-Based-Threshold Attempt Using Vz



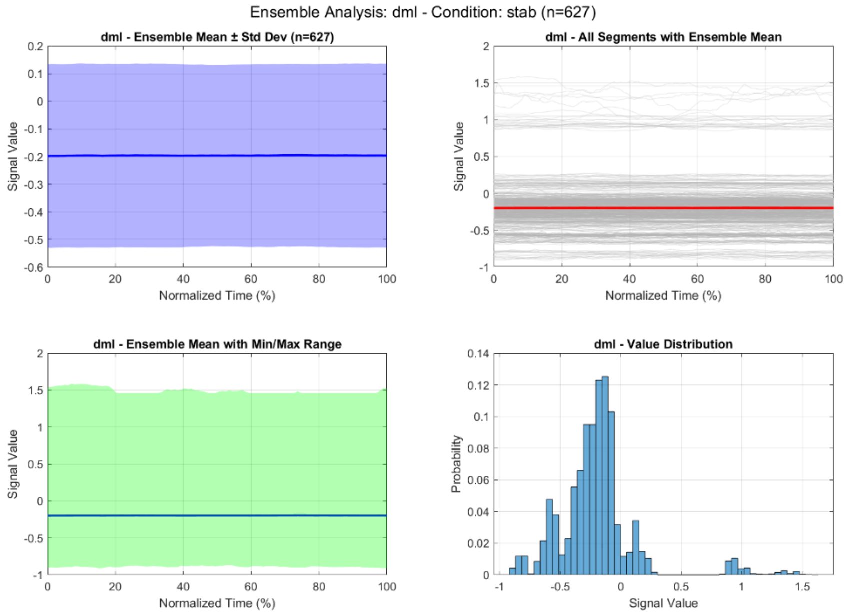
Graph A-10: Ensemble Analysis Graphs of ax Under Stable Condition to Tune Rule-based Threshold Optimisation (Compare Mean ±n SD With Other Condition, to Determine Best n)



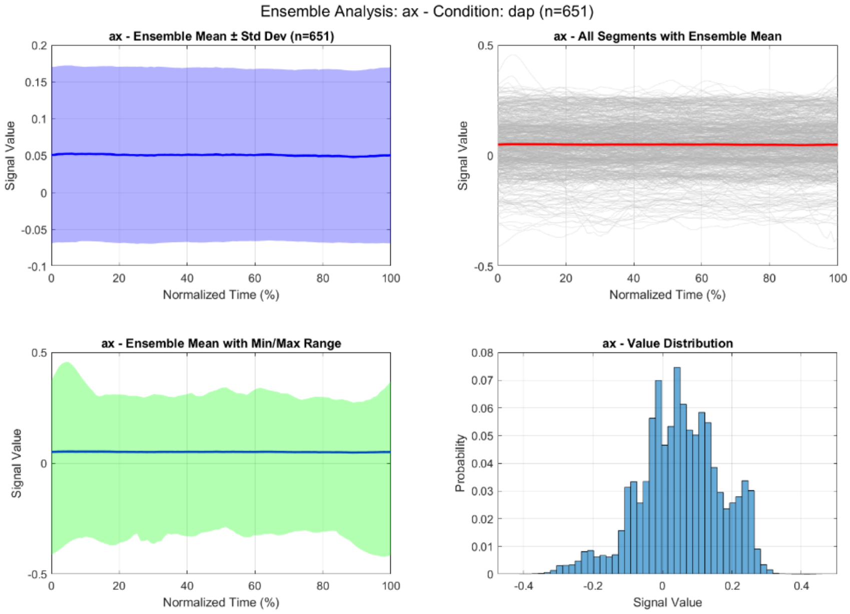
Graph A-11: Ensemble Analysis Graphs of az Under Stable Condition to Tune Rule-based Threshold Optimisation (Compare Mean ±n SD With Other Condition, to Determine Best n)



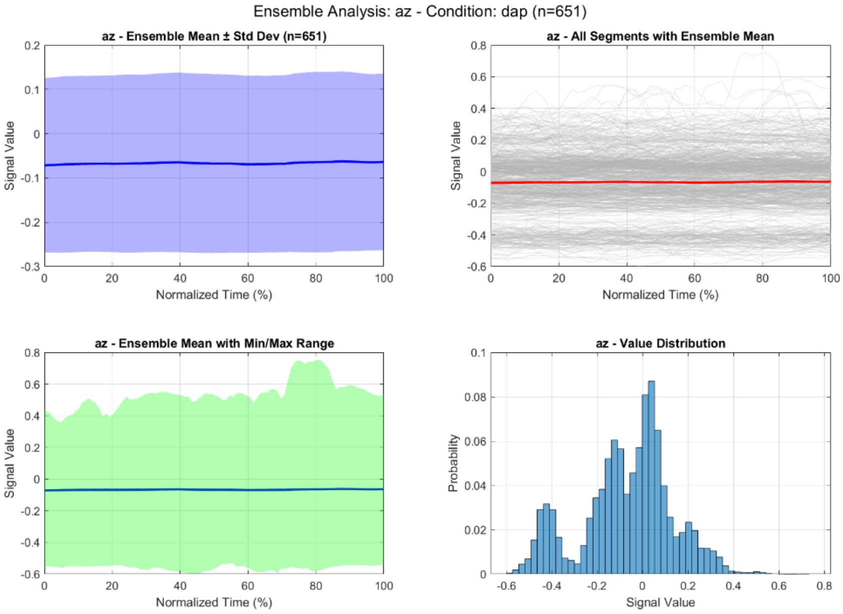
Graph A-12: Ensemble Analysis Graphs of DAP Under Stable Condition to Tune Rule-based Threshold Optimisation (Compare Mean ±n SD With Other Condition, to Determine Best n)



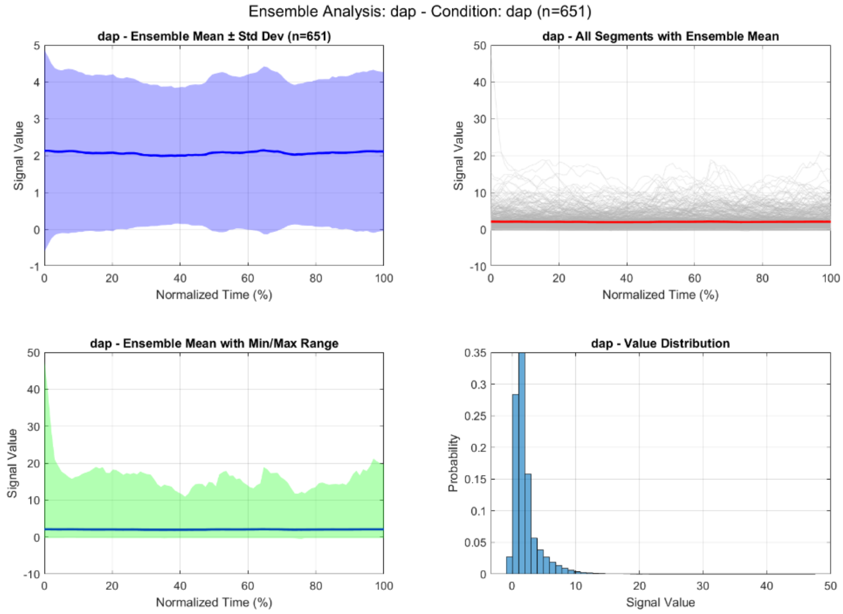
Graph A-13: Ensemble Analysis Graphs of DML Under Stable Condition to Tune Rule-based Threshold Optimisation (Compare Mean ±n SD With Other Condition, to Determine Best n)



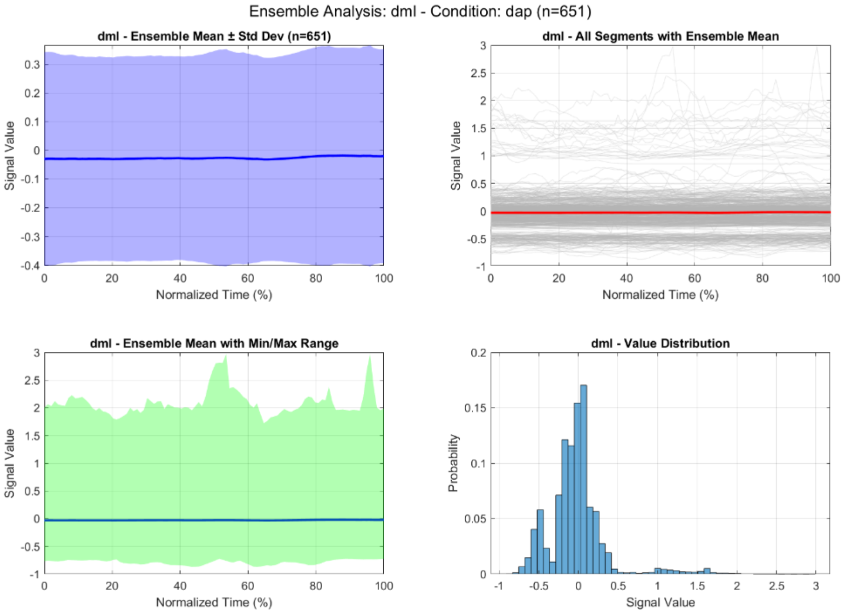
Graph A-14: Ensemble Analysis Graphs of ax Under DAP Condition to Tune Rule-based Threshold Optimisation (Compare Mean ±n SD With Other Condition, to Determine Best n)



Graph A-15: Ensemble Analysis Graphs of az Under DAP Condition to Tune Rule-based Threshold Optimisation (Compare Mean ±n SD With Other Condition, to Determine Best n)



Graph A-16: Ensemble Analysis Graphs of DAP Under DAP Condition to Tune Rule-based Threshold Optimisation (Compare Mean ±n SD With Other Condition, to Determine Best n)



Graph A-17: Ensemble Analysis Graphs of DML Under DAP Condition to Tune Rule-based Threshold Optimisation (Compare Mean ±n SD With Other Condition, to Determine Best n)

Appendix B: Tables

Table B-1: Summarised Anthropometric Data of Spinal Segments (Ko et al., 2004)

Spinal Segment	**Mean length (mm)**	**5pth length (mm)**	**95pth length (mm)**
C3	15	13	20
C4	14	10	20
C5	15	4	20
C6	12	9	20
C7	12	8	20
C8	13	8	20
T1	17	8	23
T2	18	11	23
T3	19	10	26
T4	20	12	27
T5	20	12	27
T6	23	15	27
T7	23	14	28
C8 to T7	153	90	201

Table B-2: Compilation of Relevant Anthropometrical Data for Back Support Exoskeleton Design (Gordon, 2006; Choi-Rokas and Garlie, 2014)

Measurement	**USA Male Mean (SD)**	**USA Female Mean (SD)**	**Malaysia Male Mean equivalent**	**Malaysia Female Mean Equivalent**	**Included Ranges (M)**	**Included Ranges (F)**
Height (cm)	176.07 (7.34)	164.00 (6.97)	168	157	163 - 184	153 - 170
Shoulder Breadth (cm)	54.61 (4.36)	46.85 (3.47)	50.41	43.94	47.74 - 60.32	41.47 - 50.37
Shoulder Height (cm)	144.25 (6.20)	133.36 (5.79)	139.00	128.49	134.16 - 152.32	124.09 - 139.44
Buttock Height (cm)	88.74 (4.71)	83.83 (4.52)	84.79	80.01	81.48 - 94.93	76.69 - 88.51
Torso Length (cm) (C7 to coccyx)	55.51	49.53	54.21	48.48	52.68 - 57.39	47.40 - 50.93
Shoulder to Waist (Torso Width)	38.37 (2.56)	35.15 (2.28)	36.22	33.21	34.27 - 41.67	31.53 - 37.52
Seated Torso Height (cm)	59.78 (2.96)	55.55 (2.86)	57.31	53.09	54.85 - 63.58	50.91 - 58.57
Hip Breadth (cm)	34.18 (2.03)	34.27 (2.24)	32.49	32.35	30.97 - 36.82	30.78 - 36.59
Chest Breadth (cm)	-	-	-	-	30 - 38.1	-
Chest Circumference (cm)	-	-	-	-	95.9 - 105.3	-
Shoulder Circumference (cm)	-	-	-	-	116.9 - 127.8	-
Vertical Trunk Circ. (cm)	-	-	-	-	164.2 - 180.7	-
Waist Girth (cm)	100.99 (12.32)	91.76 (13.78)	-	-	-	-
Belly Circumference (cm)	103.03 (11.86)	96.71 (11.78)	-	-	-	-
Bust Chest Girth (cm)	107.91 (9.71)	102.01 (11.36)	-	-	-	-
Buttock Girth (cm)	104.04 (7.88)	107.14 (10.78)	-	-	-	-
Hip Girth (cm)	105.18 (7.85)	108.46 (10.49)	-	-	-	-
Under Bust Circumference (cm)	103.15 (9.64)	89.07 (11.36)	-	-	-	-
Waistband (cm)	98.78 (10.18)	95.22 (11.51)	-	-	-	-
Cross Shoulder (cm)	50.24 (3.36)	46.62 (3.78)	-	-	-	-
Torso Length (cm)	32.14 (2.13)	30.13 (2.06)	-	-	-	-

Table B-3: Measured Physical Parameters of Subjects for Sway Data Collection

Subject	Sex	Height	H1	H2	Weight	BMI
F01	**Female**	1.60	1.09	0.23	59	23.047
F02	**Female**	1.70	1.2	0.25	52	17.993
F03	**Female**	1.70	1.2	0.25	52	17.993
F04	**Female**	1.61	1.07	0.22	50	19.289
F05	**Female**	1.58	1.03	0.18	55	22.032
F06	**Female**	1.53	0.92	0.28	55	23.495
F07	**Female**	1.57	1	0.23	40	16.228
M01	**Male**	1.66	1.12	0.26	58	21.048
M02	**Male**	1.77	1.24	0.26	63	20.109
M03	**Male**	1.70	1.14	0.25	50	17.301
M04	**Male**	1.84	1.26	0.26	86	25.402
M05	**Male**	1.77	1.21	0.25	64	20.428
M06	**Male**	1.70	1.12	0.29	55	19.031
M07	**Male**	1.65	1.12	0.27	50	18.365
M08	**Male**	1.74	1.18	0.28	58	19.157
M09	**Male**	1.77	1.24	0.26	65	20.748
M10	**Male**	1.74	1.2	0.3	58	19.157
M11	**Male**	1.69	1.12	0.26	84	29.411
M12	**Male**	1.67	1.1	0.23	65	23.307
M13	**Male**	1.70	1.2	0.33	59	20.415
M14	**Male**	1.79	1.2	0.24	63	19.662
M15	**Male**	1.68	1.13	0.25	49	17.361
M16	**Male**	1.79	1.23	0.26	57	17.696
M17	**Male**	1.70	1.15	0.28	65	22.491
M18	**Male**	1.78	1.21	0.28	79	24.934
M19	**Male**	1.82	1.26	0.3	60	18.214
M20	**Male**	1.77	1.22	0.26	57	18.194
M21	**Male**	1.73	1.14	0.27	59	19.713
M22	**Male**	1.76	1.23	0.25	62	20.015
M23	**Male**	1.72	1.13	0.28	60	20.231
M24	**Male**	1.71	1.08	0.29	72	24.623
M25	**Male**	1.69	1.09	0.28	88	30.811
M26	**Male**	1.73	1.16	0.28	72	24.057
M27	**Male**	1.66	1.1	0.28	56	20.322
M28	**Male**	1.66	1.1	0.28	56	20.322
M29	**Male**	1.77	1.22	0.28	63	20.109

Table B-4: Don Doff Timing of Exoskeleton By Subject

Subject	Don Time (s)	Doff Time (s)
S01	440	98
S02	301	76
S03	283	76
S04	259	80
S05	203	60
S06	200	107
S07	180	59
S08	165	43
S09	158	69
S10	148	64
S11	143	59
S12	142	70
S13	130	61
S14	120	48
S15	115	55
Mean	**199**	**68**
SD	**88**	**17**
REMOVED OUTLIERS		
S02	301	76
S03	283	76
S04	259	80
S05	203	60
S06	200	107
S07	180	59
S08	165	43
S09	158	69
S10	148	64
S11	143	59
S12	142	70
S13	130	61
S14	120	48
S15	115	55
Mean	**182**	**66**
SD	**60**	**16**

Table B-5: Summary of sEMG Results and Statistical Tests

Muscle	**Subcondition**	**EMG Mean Diff (± SE), $\times 10^{-3}\mu V$**	**p- Value (Wilcoxon)**	**Effect Size (Cohen's d)**	**Percentage Reduction**
EO	**N(EO)**	682.4818 (±264.8401)	p=0.0215*	0.72 (medium)	↓20.6%
	N(EC)	881.1538 (±385.4994)	p=0.0295*	0.70 (medium)	↓24.1%
	T(EO)	906.6716 (±327.0678)	p=0.0067**	0.96 (large)	↓26.7%
	T(EC)	1820.347 (±459.6409)	p=6.10E-05***	1.12 (large)	↓38.3%
Muscle	**Subcondition**	**EMG Mean Diff (± SE), $\times 10^{-3}\mu V$**	**p- Value (Wilcoxon)**	**Effect Size (Cohen's d)**	**Percentage Reduction**
RA	**N(EO)**	1205.630 (±498.0532)	p=0.0266*	0.82 (large)	↓33.6%
	N(EC)	904.1492 (±337.9464)	p=0.0067**	0.81 (large)	↓29.4%
	T(EO)	1114.639 (±347.1975)	p=0.0015**	1.16 (large)	↓36.1%
	T(EC)	2162.382 (±759.0160)	p=0.0034**	1.05 (large)	↓51.2%
Muscle	**Subcondition**	**EMG Mean Diff (± SE), $\times 10^{-3}\mu V$**	**p- Value (Wilcoxon)**	**Effect Size (Cohen's d)**	**Percentage Reduction**
ES (R)	**N(EO)**	287.6042 (±315.0097)	p=0.3054	0.17 (negligible)	↓8.2%
	N(EC)	228.7428 (±395.5777)	p=0.2414	0.14 (negligible)	↓6.7%
	T(EO)	2085.236 (±785.9098)	p=0.0103*	0.70 (medium)	↓35.6%
	T(EC)	2749.911 (±889.0738)	p=0.0003***	0.95 (large)	↓41.8%
Muscle	**Subcondition**	**EMG Mean Diff (± SE), $\times 10^{-3}\mu V$**	**p- Value (Wilcoxon)**	**Effect Size (Cohen's d)**	**Percentage Reduction**
ES (L)	**N(EO)**	135.0190 (±215.6976)	p=0.3396	0.14 (negligible)	↓5.2%
	N(EC)	-65.15124 (±385.4472)	p=1.0000	-0.06 (negligible)	↑2.5%
	T(EO)	-350.8893 (±655.4787)	p=1.0000	-0.15 (negligible)	↑9.1%
	T(EC)	309.5315 (±867.0291)	p=0.42627	0.10 (negligible)	↓6.0%

Note: * p<0.05, ** p<0.01, *** p<0.001

Table B-6: Wilcoxon Test Results of CoP Parameters Per Condition With Outliers Retained

Stance	Measure	Wilcoxon Statistic	p-value
NSEO	DML SD (cm)	17	0.174805
NSEO	DAP SD (cm)	26	0.577148
NSEO	Vml mean (cm/s)	12	0.067383
NSEO	Vap mean (cm/s)	17	0.174805
NSEO	CoP Pathlength (cm)	11	0.053711
NSEO	CoP 95% PEA	20	0.278320
NSEC	DML SD (cm)	10	0.080078
NSEC	DAP SD (cm)	22	0.365234
NSEC	Vml mean (cm/s)	27	0.635742
NSEC	Vap mean (cm/s)	19	0.240234
NSEC	CoP Pathlength (cm)	23	0.413086
NSEC	CoP 95% PEA	32	0.965820
TSEO	DML SD (cm)	12	0.067383
TSEO	DAP SD (cm)	24	0.464844
TSEO	Vml mean (cm/s)	5	0.009766 **
TSEO	Vap mean (cm/s)	9	0.032227 *
TSEO	CoP Pathlength (cm)	5	0.009766 **
TSEO	CoP 95% PEA	10	0.041992 *
TSEC	DML SD (cm)	9	0.032227 *
TSEC	DAP SD (cm)	8.5	0.026367 *
TSEC	Vml mean (cm/s)	0	0.000977 ***
TSEC	Vap mean (cm/s)	2	0.002930 **
TSEC	CoP Pathlength (cm)	0	0.000977 ***
TSEC	CoP 95% PEA	4	0.006836 **

Table B-7: Summary of Significant (p-value) and Effect Size (r) per Stance With Outliers Retained

Stance	Parameter	p-value	Sig.	Effect size (r)	Effect
NSEO	DML SD (cm)	0.1748	n.s	0.477	Medium
	DAP SD (cm)	0.5771	n.s	0.194	Small
	Vml mean (cm/s)	0.0674	n.s	0.316	Medium
	Vap mean (cm/s)	0.1748	n.s	0.749	Huge
	CoP Pathlength (cm)	0.0537	n.s	0.535	Large
	CoP 95% PEA	0.2783	n.s	-0.077	Small
NSEC	DML SD (cm)	0.0801	n.s	0.8	Huge
	DAP SD (cm)	0.3652	n.s	0.725	Huge
	Vml mean (cm/s)	0.6357	n.s	0.712	Huge
	Vap mean (cm/s)	0.2402	n.s	0.75	Huge
	CoP Pathlength (cm)	0.4131	n.s	0.797	Huge
	CoP 95% PEA	0.9658	n.s	0.72	Huge
TSEO	DML SD (cm)	0.0674	n.s	0.788	Huge
	DAP SD (cm)	0.4648	n.s	0.538	Large
	Vml mean (cm/s)	0.0098	**	0.921	Extreme
	Vap mean (cm/s)	0.0322	*	0.716	Large
	CoP Pathlength (cm)	0.0098	**	0.888	Large
	CoP 95% PEA	0.042	*	0.737	Large
TSEC	DML SD (cm)	0.0322	*	0.484	Medium
	DAP SD (cm)	0.0264	*	0.613	Large
	Vml mean (cm/s)	0.001	***	0.407	Medium
	Vap mean (cm/s)	0.0029	**	0.514	Large
	CoP Pathlength (cm)	0.001	***	0.357	Medium
	CoP 95% PEA	0.0068	**	0.603	Large

Table B-8: Wilcoxon Test Results of CoP Parameters Per Condition With Outliers Removed

Stance	Measure	Wilcoxon Statistic	p-value
NSEO	DML SD (cm)	1	0.007812 **
NSEO	DAP SD (cm)	12	0.250000
NSEO	Vml mean (cm/s)	3	0.019531 *
NSEO	Vap mean (cm/s)	6	0.054688
NSEO	CoP Pathlength (cm)	2	0.011719 *
NSEO	CoP 95% PEA	5	0.039062 *
NSEC	DML SD (cm)	8	0.195312
NSEC	DAP SD (cm)	20	0.820312
NSEC	Vml mean (cm/s)	21	0.886719
NSEC	Vap mean (cm/s)	16	0.496094
NSEC	CoP Pathlength (cm)	19	0.734375
NSEC	CoP 95% PEA	19	0.734375
TSEO	DML SD (cm)	7	0.074219
TSEO	DAP SD (cm)	14	0.359375
TSEO	Vml mean (cm/s)	1	0.007812 **
TSEO	Vap mean (cm/s)	1	0.007812 **
TSEO	CoP Pathlength (cm)	0	0.003906 **
TSEO	CoP 95% PEA	4	0.027344 *
TSEC	DML SD (cm)	1	0.007812 **
TSEC	DAP SD (cm)	1.5	0.011719 *
TSEC	Vml mean (cm/s)	0	0.003906 **
TSEC	Vap mean (cm/s)	2	0.011719 *
TSEC	CoP Pathlength (cm)	0	0.003906 **
TSEC	CoP 95% PEA	0	0.003906 **

Table B-9: Summary of Significant (p-value) and Effect Size (r) per Stance With Outliers Removed

Stance	Parameter	p-value	Sig.	Effect size (r)	Effect
NSEO	DML SD (cm)	0.0078	**	0.682	Large
	DAP SD (cm)	0.25	n.s.	0.323	Medium
	Vml mean (cm/s)	0.0195	*	0.309	Medium
	Vap mean (cm/s)	0.0547	n.s.	0.746	Huge
	CoP Pathlength (cm)	0.0117	*	0.528	Large
	CoP 95% PEA (cm^2)	0.0391	*	0.308	Medium
NSEC	DML SD (cm)	0.1953	n.s.	0.771	Huge
	DAP SD (cm)	0.8203	n.s.	0.808	Huge
	Vml mean (cm/s)	0.8867	n.s.	0.617	Large
	Vap mean (cm/s)	0.4961	n.s.	0.662	Large
	CoP Pathlength (cm)	0.7344	n.s.	0.697	Large
	CoP 95% PEA (cm^2)	0.7344	n.s.	0.706	Huge
TSEO	DML SD (cm)	0.0742	n.s.	0.78	Huge
	DAP SD (cm)	0.3594	n.s.	0.511	Large
	Vml mean (cm/s)	0.0078	**	0.906	Extreme
	Vap mean (cm/s)	0.0078	**	0.757	Huge
	CoP Pathlength (cm)	0.0039	**	0.9	Extreme
	CoP 95% PEA (cm^2)	0.0273	*	0.732	Huge
TSEC	DML SD (cm)	0.0078	**	0.435	Medium
	DAP SD (cm)	0.0117	*	0.609	Large
	Vml mean (cm/s)	0.0039	**	0.278	Small
	Vap mean (cm/s)	0.0117	*	0.452	Medium
	CoP Pathlength (cm)	0.0039	**	0.222	Small
	CoP 95% PEA (cm^2)	0.0039	**	0.597	Large

Table B-10: Summary of Significance and Effect Size Per Subject

Sub	Cond	Parameters	DML SD	DAP SD	Vml mean	Vap mean	CoP Pathlength	CoP 95% PEA
S01	NSEO	p-value	0.59663823	0.67059727	0.326810933	0.883897544	0.738116055	0.825983094
S01	NSEO	significance	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S01	NSEO	cohen's D	0.35992229	-0.284854	0.743311116	-0.095442714	0.221559926	0.144285623
S01	NSEO	effect	Small	Small	Medium	Very small	Small	Very small
S01	NSEC	p-value	0.13474005	0.12521354	0.012967009	0.02487353	0.019921438	0.344195102
S01	NSEC	significance	n.s.	n.s.	*	*	*	n.s.
S01	NSEC	cohen's D	-0.1236194	-0.1396997	-1.479088322	-3.123580759	-2.76283872	0.148197467
S01	NSEC	effect	Very small	Very small	Very large	Huge	Huge	Very small
S01	TSEO	p-value	0.68709655	0.00833747	0.042266449	0.395838764	0.197581084	0.873177755
S01	TSEO	significance	n.s.	*	*	n.s.	n.s.	n.s.
S01	TSEO	cohen's D	0.28057225	-6.1101009	2.547110812	0.619047619	1.051478925	-0.022456926
S01	TSEO	effect	Small	Huge	Huge	Medium	Large	Very small
S01	TSEC	p-value	0.06255885	0.00767339	0.060686016	0.018003074	0.038358671	0.077999704
S01	TSEC	significance	n.s.	*	n.s.	*	*	n.s.
S01	TSEC	cohen's D	1.45962073	2.5692087	2.035109001	4.155459474	2.834563997	1.312908982
S01	TSEC	effect	Very large	Huge	Huge	Huge	Huge	Very large
S02	NSEO	p-value	0.37838574	0.10479474	0.337733821	0.17129801	0.263996386	0.235328499
S02	NSEO	significance	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S02	NSEO	cohen's D	-0.6479391	-1.6401332	-0.721687836	-1.20894105	-0.887687316	-0.96885788
S02	NSEO	effect	Medium	Very large	Medium	Large	Large	Large
S02	NSEC	p-value	0.43699001	0.06501586	0.341956644	0.008128887	0.036813838	0.248995911
S02	NSEC	significance	n.s.	n.s.	n.s.	*	*	n.s.
S02	NSEC	cohen's D	-0.5562293	-2.152337	-0.71355165	-6.364487698	-2.925353252	-0.928660524
S02	NSEC	effect	Medium	Huge	Medium	Huge	Huge	Large

S02	**TSEO**	**p-value**	0.25357495	0.4475099	0.104188167	0.479693687	0.412371332	0.469193747
S02	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S02	**TSEO**	**cohen's D**	0.91582485	0.54120665	1.645751852	0.497468877	0.59297809	0.511392294
S02	**TSEO**	**effect**	Large	Medium	Very large	Small	Medium	Medium
S02	**TSEC**	**p-value**	0.32596448	0.82974869	0.271256613	0.191251528	0.227792447	0.213841253
S02	**TSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S02	**TSEC**	**cohen's D**	-0.745023	0.14106912	0.868907536	1.12273251	0.992342208	-1.038625183
S02	**TSEC**	**effect**	Medium	Very small	Large	Large	Large	Large
S03	**NSEO**	**p-value**	0.42677031	0.3444091	0.766746756	0.323349036	0.401883206	0.048762704
S03	**NSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	*
S03	**NSEO**	**cohen's D**	0.57120221	0.70888121	-0.195852878	0.750346669	0.609377172	2.517933798
S03	**NSEO**	**effect**	Medium	Medium	Very small	Medium	Medium	Huge
S03	**NSEC**	**p-value**	0.35325109	0.63729664	0.00617971	0.316564776	0.641655956	0.2491386
S03	**NSEC**	**significance**	n.s.	n.s.	*	n.s.	n.s.	n.s.
S03	**NSEC**	**cohen's D**	-0.6923642	0.31778579	-7.310299323	0.76440273	-0.313399685	0.92825598
S03	**NSEC**	**effect**	Medium	Small	Huge	Medium	Small	Large
S03	**TSEO**	**p-value**	0.85594066	0.66687651	0.442097507	0.657417708	0.469720976	0.78278407
S03	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S03	**TSEO**	**cohen's D**	-0.1188638	-0.2884707	0.548887706	0.29773379	0.510685502	0.181694273
S03	**TSEO**	**effect**	Very small	Small	Medium	Small	Medium	Very small
S03	**TSEC**	**p-value**	0.16347186	0.09960394	0.00294329	0.009629029	0.004124172	0.024918892
S03	**TSEC**	**significance**	n.s.	n.s.	*	*	*	*
S03	**TSEC**	**cohen's D**	0.57900666	1.23213938	3.656527931	4.622516762	5.164918436	3.119318738
S03	**TSEC**	**effect**	Medium	Large	Huge	Huge	Huge	Huge
S04	**NSEO**	**p-value**	0.16894797	0.18433052	0.071829854	0.135958174	0.114723802	0.226815792
S04	**NSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.

S04	**NSEO**	**cohen's D**	1.21998856	1.15120207	2.036370506	1.401388821	1.554242932	0.995459909
S04	**NSEO**	**effect**	Large	Large	Huge	Very large	Very large	Large
S04	**NSEC**	**p-value**	0.46321476	0.81483598	0.21795091	0.237599264	0.21249238	0.45812069
S04	**NSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S04	**NSEC**	**cohen's D**	0.51946603	-0.1538462	1.024587872	0.961972813	1.043310322	0.526431412
S04	**NSEC**	**effect**	Medium	Very small	Large	Large	Large	Medium
S04	**TSEO**	**p-value**	0.65372788	0.51439948	0.750487965	0.439688349	0.655146931	0.681106378
S04	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S04	**TSEO**	**cohen's D**	-0.3013748	-0.4535574	0.210379675	0.552339203	0.299972672	-0.274718462
S04	**TSEO**	**effect**	Small	Small	Small	Medium	Small	Small
S04	**TSEC**	**p-value**	0.08515509	0.07636077	0.02404624	0.001795706	0.012887558	0.08050724
S04	**TSEC**	**significance**	n.s.	n.s.	*	*	*	n.s.
S04	**TSEC**	**cohen's D**	1.15615551	1.08169683	2.016007049	5.037752737	2.806533028	1.026274932
S04	**TSEC**	**effect**	Large	Large	Huge	Huge	Huge	Large
S05	**NSEO**	**p-value**	0.03921569	0.18369628	0.062612396	0.144603401	0.028648555	0.836594252
S05	**NSEO**	**significance**	*	n.s.	n.s.	n.s.	*	n.s.
S05	**NSEO**	**cohen's D**	2.82901632	-1.1538829	2.197531736	1.348386072	3.337316315	0.135237975
S05	**NSEO**	**effect**	Huge	Large	Huge	Very large	Huge	Very small
S05	**NSEC**	**p-value**	0.29985996	0.37708479	0.013072458	0.238202303	0.426226302	0.592228953
S05	**NSEC**	**significance**	n.s.	n.s.	*	n.s.	n.s.	n.s.
S05	**NSEC**	**cohen's D**	-0.8006408	0.65015346	5	-0.960158717	-0.57201016	0.36463632
S05	**NSEC**	**effect**	Large	Medium	Huge	Large	Medium	Small
S05	**TSEO**	**p-value**	0.27410862	0.62636764	0.456108029	0.893334509	0.396367972	0.322279164
S05	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S05	**TSEO**	**cohen's D**	0.86170366	-0.3288887	0.529205934	0.087591723	0.618194315	0.752539403
S05	**TSEO**	**effect**	Large	Small	Medium	Very small	Medium	Medium

S05	**TSEC**	**p-value**	0.11092247	0.12035179	0.093183865	0.431795245	0.026350521	0.009767495
S05	**TSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	*	*
S05	**TSEC**	**cohen's D**	0.91359083	0.1434992	1.713492918	-0.132616927	1.489365908	2.446562277
S05	**TSEC**	**effect**	Large	Very small	Very large	Very small	Very large	Huge
S06	**NSEO**	**p-value**	0.05828028	0.00092208	0.029295995	0.027618752	0.018732489	0.869317947
S06	**NSEO**	**significance**	n.s.	*	*	*	*	n.s.
S06	**NSEO**	**cohen's D**	2.28571429	19	3.298574998	3.401680257	4.158842688	-0.107624401
S06	**NSEO**	**effect**	Huge	Huge	Huge	Huge	Huge	Very small
S06	**NSEC**	**p-value**	0.5752046	0.73431553	0.568461015	0.458472386	0.503449974	0.117970759
S06	**NSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S06	**NSEC**	**cohen's D**	-0.3831305	-0.2250176	-0.390591217	-0.525947907	-0.467082503	-1.528399836
S06	**NSEC**	**effect**	Small	Small	Small	Medium	Small	Very large
S06	**TSEO**	**p-value**	0.05486376	0.05828028	0.266971128	0.285034924	0.267315983	0.199587765
S06	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S06	**TSEO**	**cohen's D**	2.36227796	2.28571429	0.879914358	0.834953945	0.879020373	1.090222533
S06	**TSEO**	**effect**	Huge	Huge	Large	Large	Large	Large
S06	**TSEC**	**p-value**	0.10876058	0.2245194	0.014021945	0.050189528	0.019138845	0.084963832
S06	**TSEC**	**significance**	n.s.	n.s.	*	n.s.	*	n.s.
S06	**TSEC**	**cohen's D**	1.00674825	1.07989849	2.872825788	2.476120872	3.116513804	1.097183671
S06	**TSEC**	**effect**	Large	Large	Huge	Huge	Huge	Large
S07	**NSEO**	**p-value**	0.02992352	0.32181135	0.753817018	0.555409381	0.727987582	0.081358128
S07	**NSEO**	**significance**	*	n.s.	n.s.	n.s.	n.s.	n.s.
S07	**NSEO**	**cohen's D**	-3.262214	-0.753501	-0.207390339	-0.405261796	-0.230799807	-1.898467533
S07	**NSEO**	**effect**	Huge	Medium	Small	Small	Small	Very large
S07	**NSEC**	**p-value**	0.88990362	0.19461273	0.74180111	0.0648368	0.079064181	0.847287773
S07	**NSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.

S07	**NSEC**	**cohen's D**	-0.0904431	-1.1094004	-0.21821789	-2.155619955	-1.929467283	-0.126168884
S07	**NSEC**	**effect**	Very small	Large	Small	Huge	Very large	Very small
S07	**TSEO**	**p-value**	0.74227778	0.67159608	0.665060748	0.413940784	0.50231756	0.677920722
S07	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S07	**TSEO**	**cohen's D**	-0.2177863	-0.2838857	-0.290241164	-0.590564127	-0.468497889	-0.277778706
S07	**TSEO**	**effect**	Small	Small	Small	Medium	Small	Small
S07	**TSEC**	**p-value**	0.87993995	0.77747277	0.366496488	0.459218466	0.439958397	0.579166661
S07	**TSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S07	**TSEC**	**cohen's D**	-0.0739221	-0.1719238	0.775880177	0.591994038	0.674562681	0.27217929
S07	**TSEC**	**effect**	Very small	Very small	Medium	Medium	Medium	Small
S08	**NSEO**	**p-value**	0.50393893	0.64327781	0.379109777	0.990205958	0.685799472	0.560509656
S08	**NSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S08	**NSEO**	**cohen's D**	0.46647234	0.31177389	0.646710478	-0.007997185	0.27022896	0.399491719
S08	**NSEO**	**effect**	Small	Small	Medium	Very small	Small	Small
S08	**NSEC**	**p-value**	0.62598033	0.78861705	0.921523357	0.413413924	0.532389288	0.933431197
S08	**NSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S08	**NSEC**	**cohen's D**	-0.329285	-0.1765837	-0.064274135	-0.591373353	-0.431935332	-0.054474032
S08	**NSEC**	**effect**	Small	Very small	Very small	Medium	Small	Very small
S08	**TSEO**	**p-value**	0.4003266	0.31920013	0.839285563	0.495423434	0.830091653	0.327213091
S08	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S08	**TSEO**	**cohen's D**	0.61185183	0.75889956	-0.132951026	0.47718417	0.140776493	0.742499657
S08	**TSEO**	**effect**	Medium	Medium	Very small	Small	Very small	Medium
S08	**TSEC**	**p-value**	0.01155598	0.43922785	0.426760064	0.74180111	0.485709864	0.027123668
S08	**TSEC**	**significance**	*	n.s.	n.s.	n.s.	n.s.	*
S08	**TSEC**	**cohen's D**	1.69969791	-0.6405126	0.696216501	0.458831468	0.637122326	0.947199176
S08	**TSEC**	**effect**	Very large	Medium	Medium	Small	Medium	Large

S09	**NSEO**	**p-value**	0.3902986	0.3326749	0.49346839	0.568901437	0.596537578	0.266315793
S09	**NSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S09	**NSEO**	**cohen's D**	-0.6280589	-0.7315966	-0.479669527	-0.390101533	-0.360029564	-0.881617261
S09	**NSEO**	**effect**	Medium	Medium	Small	Small	Small	Large
S09	**NSEC**	**p-value**	0.25865427	0.09964699	0.883489654	0.605594681	0.71642807	0.098575862
S09	**NSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S09	**NSEC**	**cohen's D**	-0.90193	-1.6893434	0.095782629	0.350438322	0.241446663	-1.700038518
S09	**NSEC**	**effect**	Large	Very large	Very small	Small	Small	Very large
S09	**TSEO**	**p-value**	0.3296036	0.32102859	0.350683827	0.912281702	0.056133311	0.33097195
S09	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S09	**TSEO**	**cohen's D**	0.73770087	0.75511371	0.697109119	-0.071898838	2.333036109	0.734972831
S09	**TSEO**	**effect**	Medium	Medium	Medium	Very small	Huge	Medium
S09	**TSEC**	**p-value**	0.58858089	0.12481181	0.297898071	0.619936593	0.407933769	0.251824657
S09	**TSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S09	**TSEC**	**cohen's D**	-0.044585	0.80858312	0.609992547	0.016522297	0.378403665	0.453537042
S09	**TSEC**	**effect**	Very small	Large	Medium	Very small	Small	Small
S10	**NSEO**	**p-value**	0.32454916	0.13147556	0.284905042	0.130405441	0.231943498	0.353366071
S10	**NSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S10	**NSEO**	**cohen's D**	0.7478974	1.43075246	0.835264327	1.437969289	0.979283349	0.692152681
S10	**NSEO**	**effect**	Medium	Very large	Large	Very large	Large	Medium
S10	**NSEC**	**p-value**	0.97719604	0.09273529	0.855752559	0.904963723	0.918051372	0.896915675
S10	**NSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S10	**NSEC**	**cohen's D**	0.0186242	1.76140969	-0.119022319	-0.07794961	-0.067136584	0.084618795
S10	**NSEC**	**effect**	Very small	Very large	Very small	Very small	Very small	Very small
S10	**TSEO**	**p-value**	0.53276508	0.37628891	0.482671092	0.42831582	0.458196063	0.447966334
S10	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.

S10	**TSEO**	**cohen's D**	0.43149122	0.65151249	0.493577788	0.568912985	0.526327759	0.54056345
S10	**TSEO**	**effect**	Small	Medium	Small	Medium	Medium	Medium
S10	**TSEC**	**p-value**	0.10390491	0.11932214	0.062856853	0.054577311	0.067143976	0.127906113
S10	**TSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S10	**TSEC**	**cohen's D**	1.46448308	1.41695164	1.89851534	2.380107231	1.978717049	1.266165355
S10	**TSEC**	**effect**	Very large	Very large	Very large	Huge	Very large	Large
S11	**NSEO**	**p-value**	0.46704182	0.2395775	0.217120135	0.20728429	0.210875835	0.258644991
S11	**NSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S11	**NSEO**	**cohen's D**	0.51428571	0.95604396	1.02739726	1.061776062	1.04897724	0.901955063
S11	**NSEO**	**effect**	Medium	Large	Large	Large	Large	Large
S11	**NSEC**	**p-value**	1	0.7446001	0.670685761	0.819515095	0.712737523	0.718349241
S11	**NSEC**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S11	**NSEC**	**cohen's D**	0	-0.2156863	0.284768212	-0.149825784	0.244869619	-0.239669421
S11	**NSEC**	**effect**	Very small	Small	Small	Very small	Small	Small
S11	**TSEO**	**p-value**	0.62203553	0.18692811	0.198270399	0.284472638	0.244691897	0.13400819
S11	**TSEO**	**significance**	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
S11	**TSEO**	**cohen's D**	0.33333333	1.14035088	1.095238095	0.836298932	0.94100524	1.413994169
S11	**TSEO**	**effect**	Small	Large	Large	Large	Large	Very large
S11	**TSEC**	**p-value**	0.54916518	0.0527035	0.0030452	0.044814112	0.049250861	0.00408215
S11	**TSEC**	**significance**	n.s.	n.s.	*	*	*	*
S11	**TSEC**	**cohen's D**	0.57142857	1.06060606	6.36	1.187845304	1.112612613	4.86784141
S11	**TSEC**	**effect**	Medium	Large	Huge	Large	Large	Huge

Table B-11: Summary of CoP Parameters by Conditions

Cond	Stance		DML_{SD} (cm)	DAP_{SD} (cm)	Vml_{mean} (cm/s)	Vap_{mean} (cm/s)	CoP Pathlength (cm)	CoP 95% PEA (cm^2)
NO	**NSEC**	mean	0.407	0.739	0.672	0.962	38.855	5.511
		SD	0.197	0.253	0.335	0.368	16.266	4.716
		SEM	0.062	0.080	0.106	0.116	5.144	1.491
	NSEO	mean	0.526	0.754	0.734	0.892	38.652	8.161
		SD	0.316	0.307	0.372	0.545	21.117	7.273
		SEM	0.100	0.097	0.118	0.172	6.678	2.300
	TSEC	mean	1.692	1.359	4.576	3.920	197.852	46.160
		SD	0.935	0.389	2.584	1.268	87.403	47.157
		SEM	0.296	0.123	0.817	0.401	27.639	14.912
	TSEO	mean	1.039	0.779	2.052	1.865	92.508	18.601
		SD	0.530	0.304	0.919	0.567	33.732	22.516
		SEM	0.168	0.096	0.291	0.179	10.667	7.120
ON	**NSEC**	mean	0.492	0.776	0.652	1.010	38.044	4.831
		SD	0.339	0.201	0.292	0.295	10.979	2.233
		SEM	0.107	0.063	0.092	0.093	3.472	0.706
	NSEO	mean	0.408	0.669	0.515	0.666	27.205	4.176
		SD	0.200	0.191	0.147	0.169	5.570	2.594
		SEM	0.063	0.060	0.047	0.053	1.761	0.820
	TSEC	mean	1.087	1.090	2.469	2.792	123.371	19.371
		SD	0.311	0.173	1.109	0.916	41.963	9.206
		SEM	0.098	0.055	0.351	0.290	13.270	2.911
	TSEO	mean	0.847	0.690	1.611	1.585	74.802	10.049
		SD	0.375	0.177	0.511	0.290	17.067	5.918

		SEM	0.113	0.053	0.154	0.087	5.146	1.784
DIFF	**NSEC**	mean	-0.085	-0.037	0.02	-0.048	0.811	0.68
		SD	0.227	0.184	0.253	0.256	10.520	3.643
		SEM	0.068	0.055	0.076	0.077	3.172	1.098
	NSEO	mean	0.119	0.086	0.218	0.226	11.447	3.985
		SD	0.296	0.344	0.371	0.455	19.652	8.293
		SEM	0.089	0.104	0.112	0.137	5.925	2.500
	TSEC	mean	0.605	0.270	2.107	1.127	74.480	26.790
		SD	0.871	0.329	2.477	1.173	86.348	44.313
		SEM	0.263	0.099	0.747	0.354	26.035	13.361
	TSEO	mean	0.192	0.089	0.441	0.280	17.706	8.552
		SD	0.348	0.269	0.532	0.437	21.679	19.665
		SEM	0.105	0.081	0.160	0.132	6.537	5.929
% RED	**NSEC**	mean	-20.88%	-5.01%	2.98%	-4.99%	2.09%	12.34%
		SD	33.16%	24.13%	27.05%	24.87%	22.11%	71.55%
		SEM	10.00%	7.28%	8.16%	7.50%	6.67%	21.57%
	NSEO	mean	22.62%	11.41%	29.70%	25.34%	29.62%	48.83%
		SD	49.85%	36.58%	33.59%	30.48%	31.34%	80.59%
		SEM	15.03%	11.03%	10.13%	9.19%	9.45%	24.30%
	TSEC	mean	35.76%	19.87%	46.04%	28.75%	37.64%	58.04%
		SD	34.97%	18.24%	23.24%	22.22%	23.37%	42.64%
		SEM	10.55%	5.50%	7.01%	6.70%	7.05%	12.86%
	TSEO	mean	18.48%	11.42%	21.49%	15.01%	19.14%	45.98%
		SD	26.41%	24.68%	16.90%	19.91%	17.96%	35.39%
		SEM	7.96%	7.44%	5.10%	6.00%	5.42%	10.67%

Appendix C: Figures

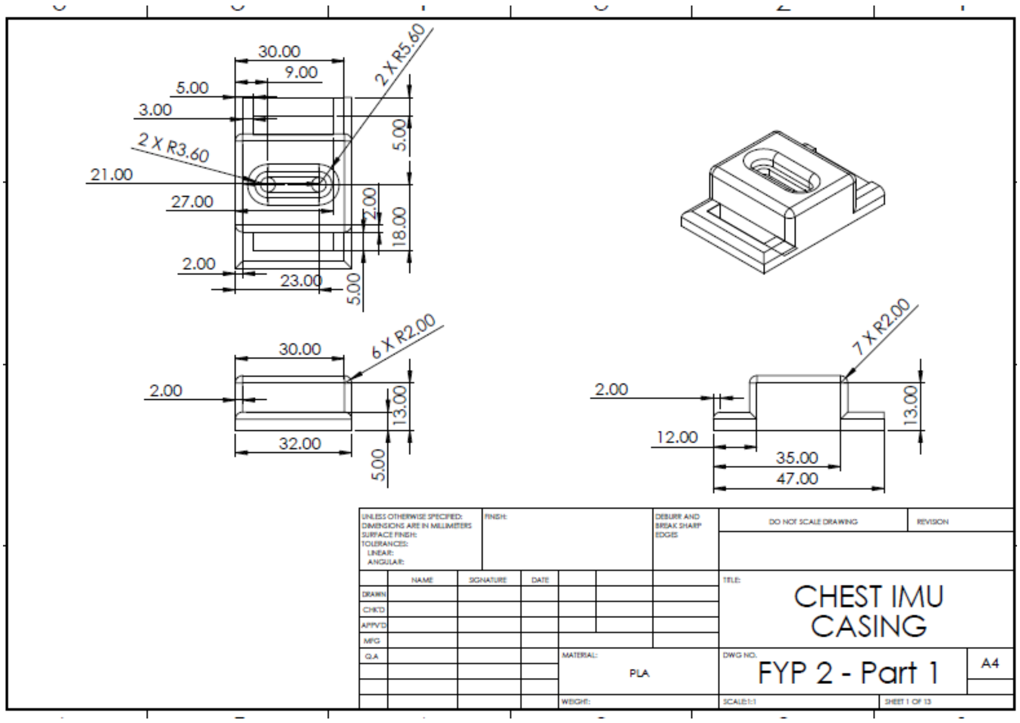


Figure C-1: SOLIDWORKS Drawing of Actual Chest IMU Casing

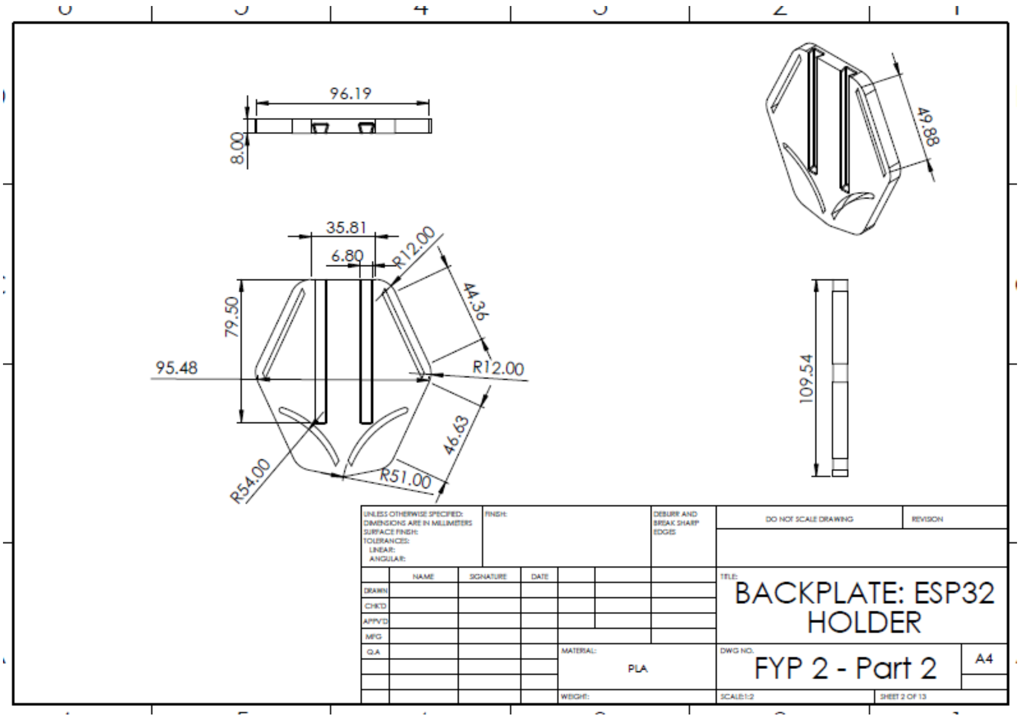


Figure C-2: SOLIDWORKS Drawing of Actual Backplate Part of ESP32 Holder

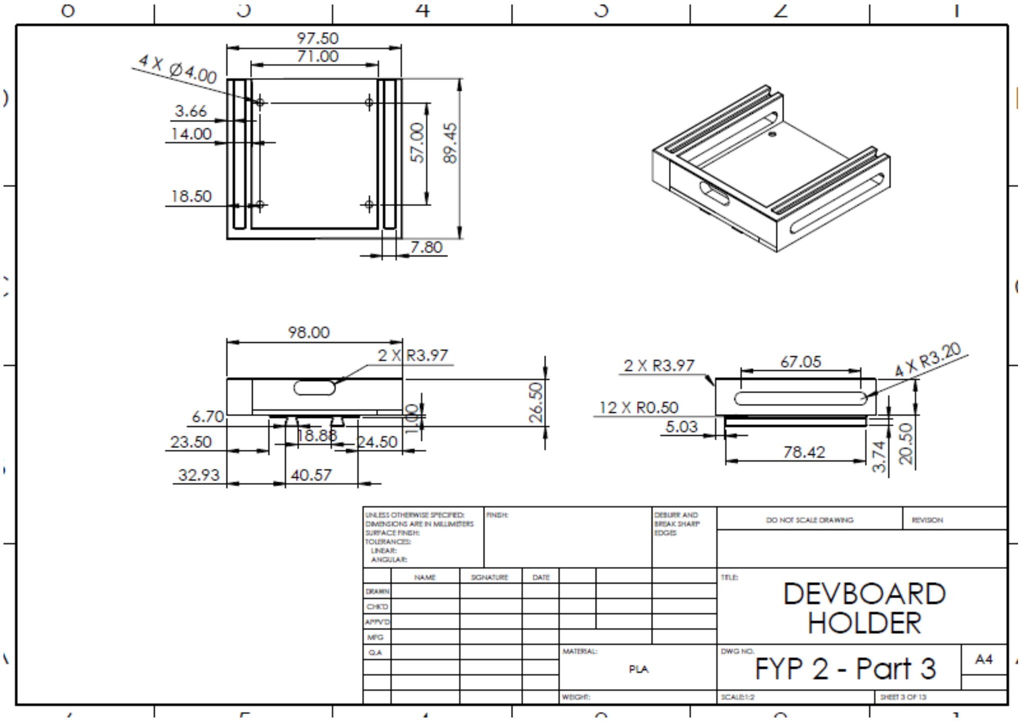


Figure C-3: SOLIDWORKS Drawing of Actual Devboard Holder

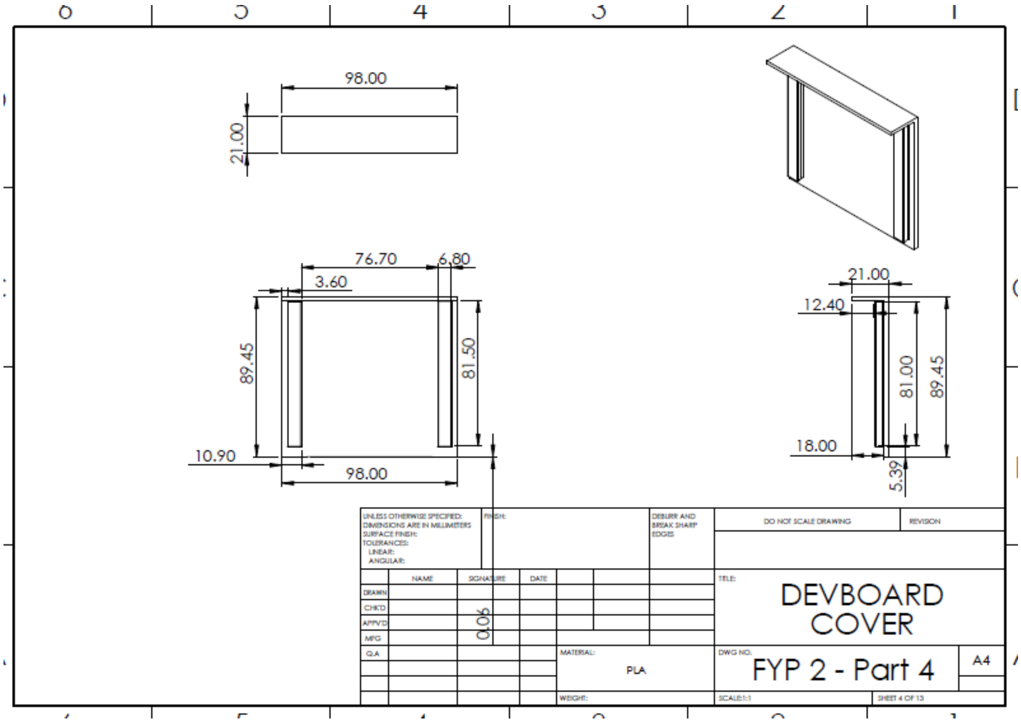


Figure C-4: SOLIDWORKS Drawing of Actual Devboard Cover

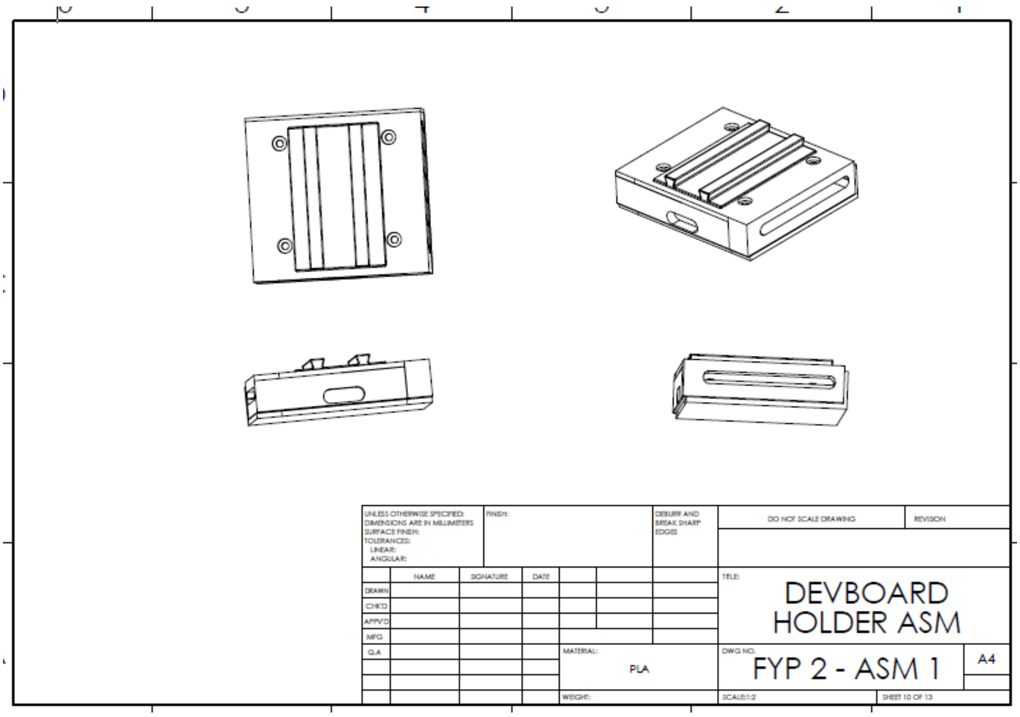


Figure C-5: SOLIDWORKS Drawing of Actual Devboard Holder Assembly

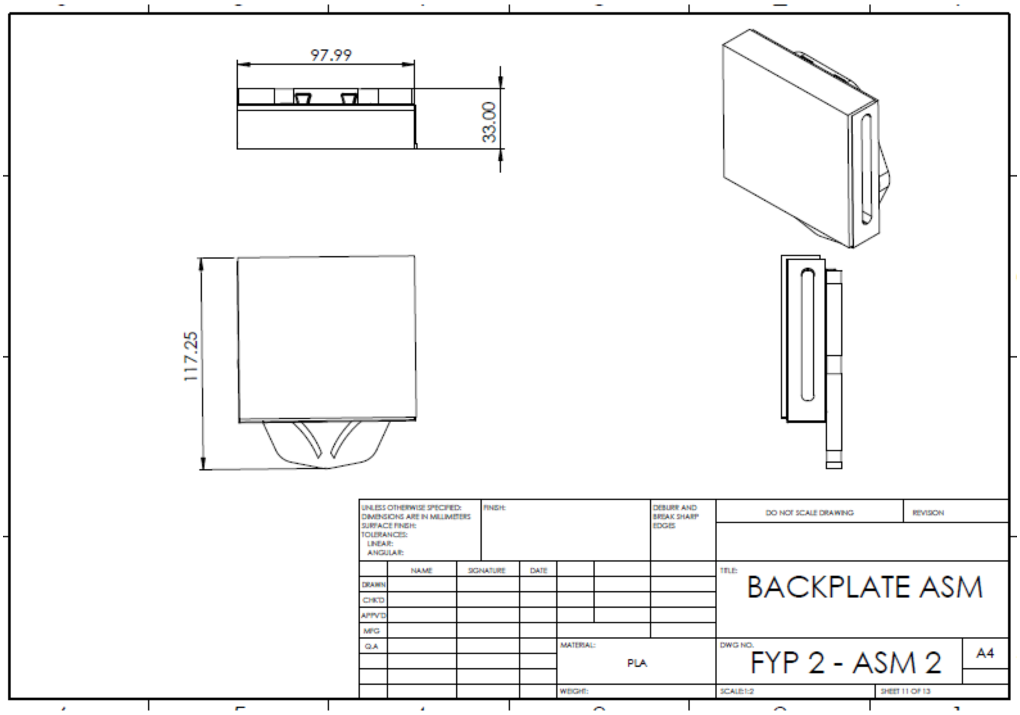


Figure C-6: SOLIDWORKS Drawing of Actual Backplate Assembly

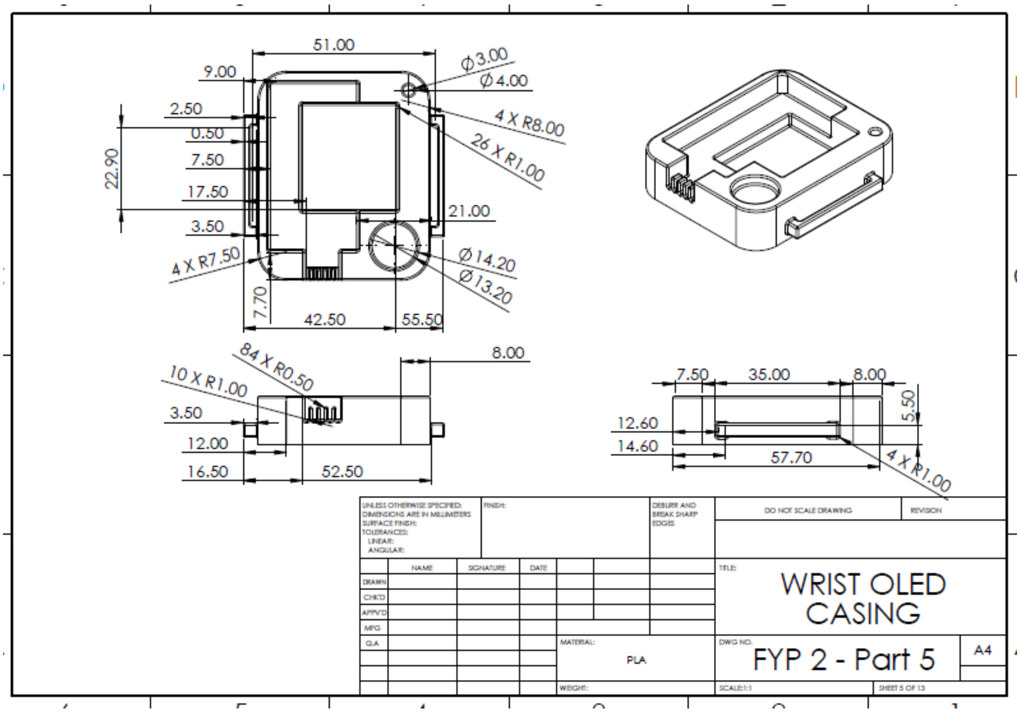


Figure C-7: SOLIDWORKS Drawing of Actual Wrist OLED Casing

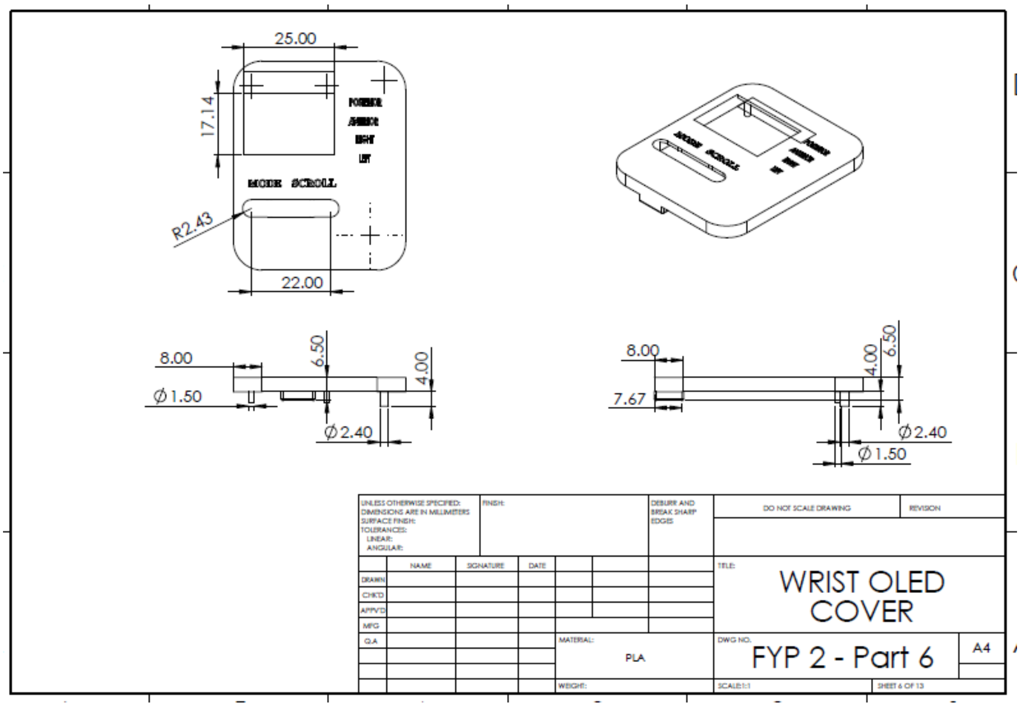


Figure C-8: SOLIDWORKS Drawing of Actual Wrist OLED Cover

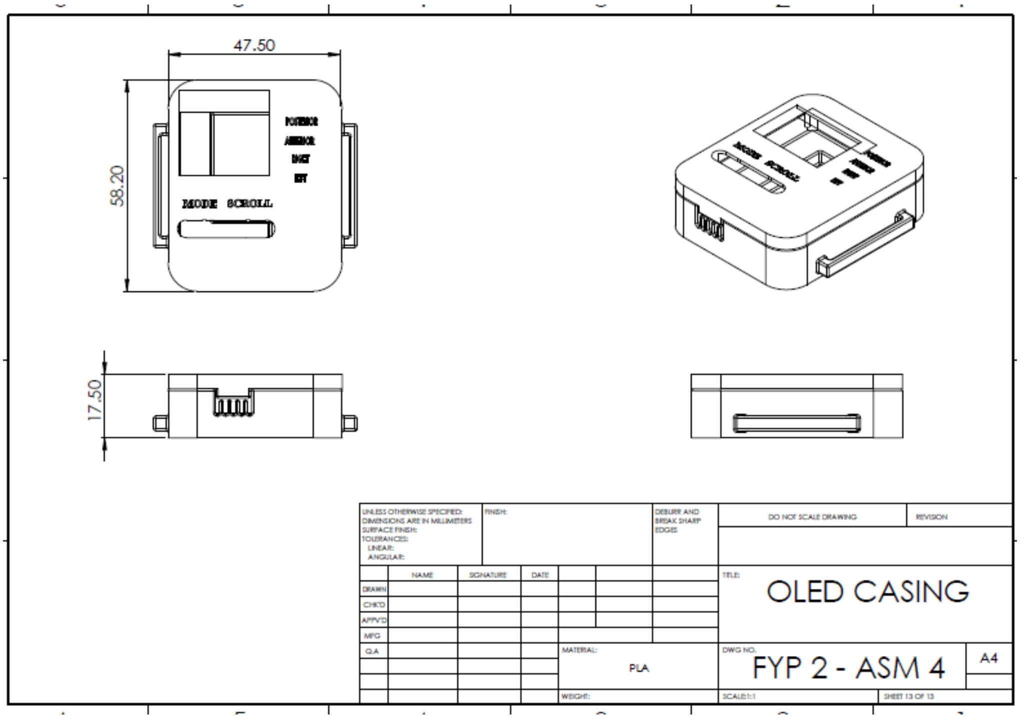


Figure C-9: SOLIDWORKS Drawing of Actual OLED Casing Assembly

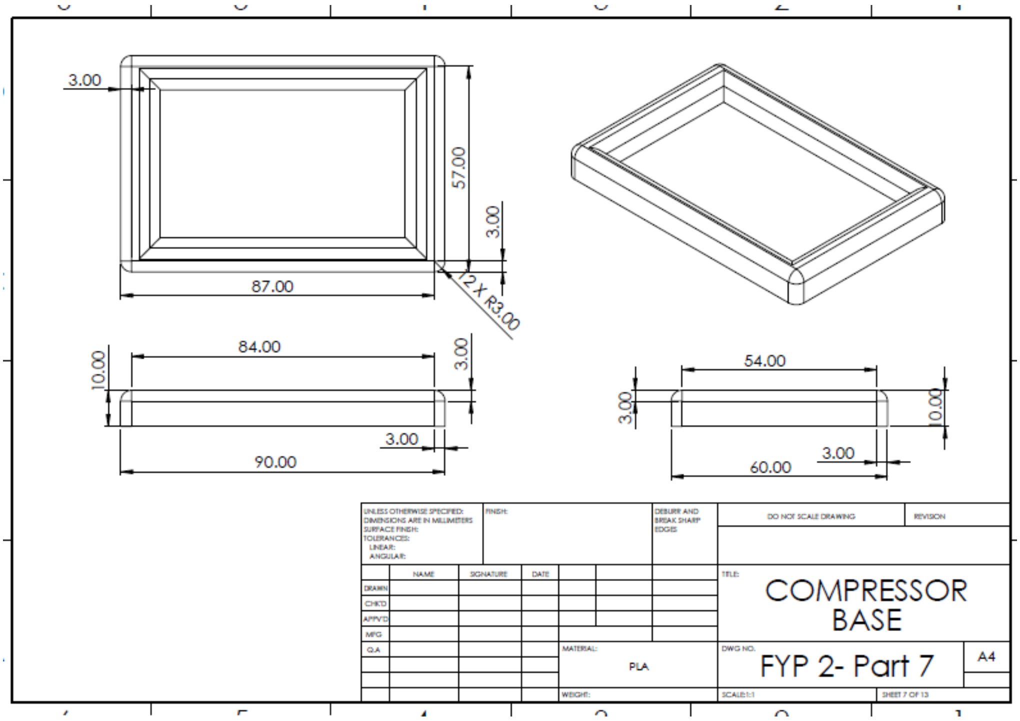


Figure C-10: SOLIDWORKS Drawing of Actual Compressor Base for Casing

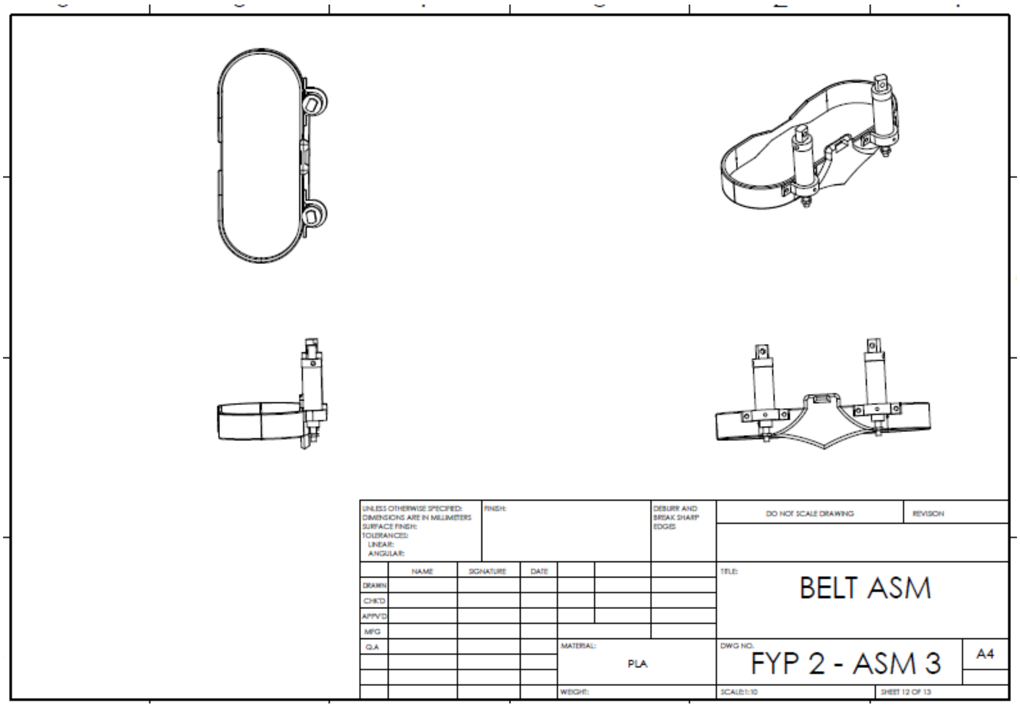
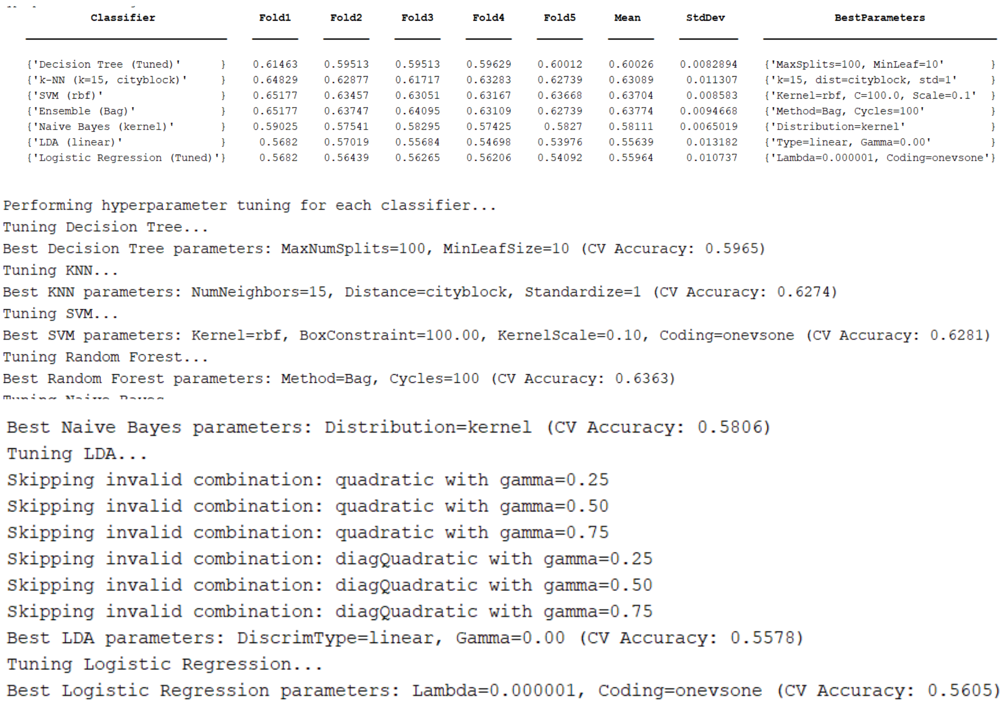


Figure C-11: SOLIDWORKS Drawing of Actual Belt Assembly



```
              Classifier                  Fold1      Fold2      Fold3      Fold4      Fold5      Mean       StdDev                   BestParameters
      _____________________________       _______    _______    _______    _______    _______    _______    _________    _________________________________________

      {'Decision Tree (Tuned)'       }    0.61463    0.59513    0.59513    0.59629    0.60012    0.60026    0.0082894    {'MaxSplits=100, MinLeaf=10'          }
      {'k-NN (k=15, cityblock)'      }    0.64829    0.62877    0.61717    0.63283    0.62739    0.63089     0.011307    {'k=15, dist=cityblock, std=1'        }
      {'SVM (rbf)'                   }    0.65177    0.63457    0.63051    0.63167    0.63668    0.63704     0.008583    {'Kernel=rbf, C=100.0, Scale=0.1'     }
      {'Ensemble (Bag)'              }    0.65177    0.63747    0.64095    0.63109    0.62739    0.63774    0.0094668    {'Method=Bag, Cycles=100'             }
      {'Naive Bayes (kernel)'        }    0.59025    0.57541    0.58295    0.57425     0.5827    0.58111    0.0065019    {'Distribution=kernel'                }
      {'LDA (linear)'                }     0.5682    0.57019    0.55684    0.54698    0.53976    0.55639     0.013182    {'Type=linear, Gamma=0.00'            }
      {'Logistic Regression (Tuned)'}     0.5682    0.56439    0.56265    0.56206    0.54092    0.55964     0.010737    {'Lambda=0.000001, Coding=onevsone'}

Performing hyperparameter tuning for each classifier...
Tuning Decision Tree...
Best Decision Tree parameters: MaxNumSplits=100, MinLeafSize=10 (CV Accuracy: 0.5965)
Tuning KNN...
Best KNN parameters: NumNeighbors=15, Distance=cityblock, Standardize=1 (CV Accuracy: 0.6274)
Tuning SVM...
Best SVM parameters: Kernel=rbf, BoxConstraint=100.00, KernelScale=0.10, Coding=onevsone (CV Accuracy: 0.6281)
Tuning Random Forest...
Best Random Forest parameters: Method=Bag, Cycles=100 (CV Accuracy: 0.6363)
Tuning Naive Bayes...
Best Naive Bayes parameters: Distribution=kernel (CV Accuracy: 0.5806)
Tuning LDA...
Skipping invalid combination: quadratic with gamma=0.25
Skipping invalid combination: quadratic with gamma=0.50
Skipping invalid combination: quadratic with gamma=0.75
Skipping invalid combination: diagQuadratic with gamma=0.25
Skipping invalid combination: diagQuadratic with gamma=0.50
Skipping invalid combination: diagQuadratic with gamma=0.75
Best LDA parameters: DiscrimType=linear, Gamma=0.00 (CV Accuracy: 0.5578)
Tuning Logistic Regression...
Best Logistic Regression parameters: Lambda=0.000001, Coding=onevsone (CV Accuracy: 0.5605)
```

Figure C-12: MATLAB Snippets of Classifier Training Results

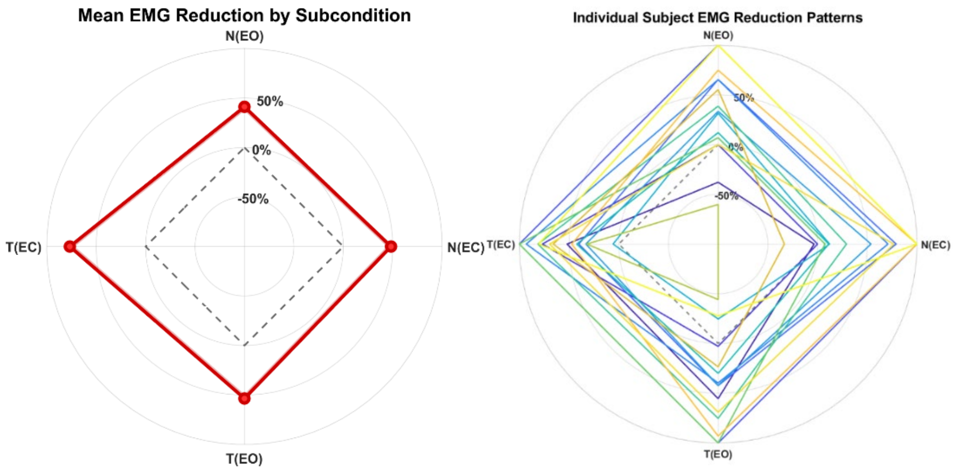


Figure C-13: Visualisation of EMG Reduction Based on Mean and Subject for External Obliques (EO)

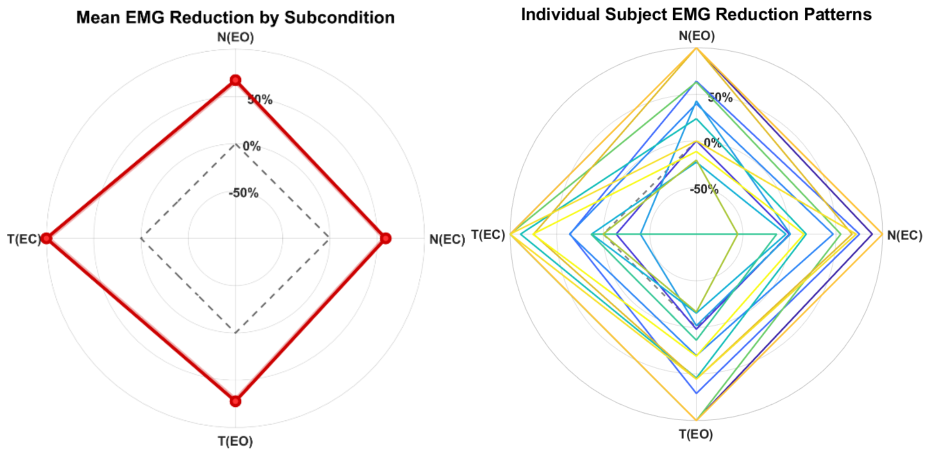


Figure C-14: Visualisation of EMG Reduction Based on Mean and Subject for Rectus Abdominus (RA)

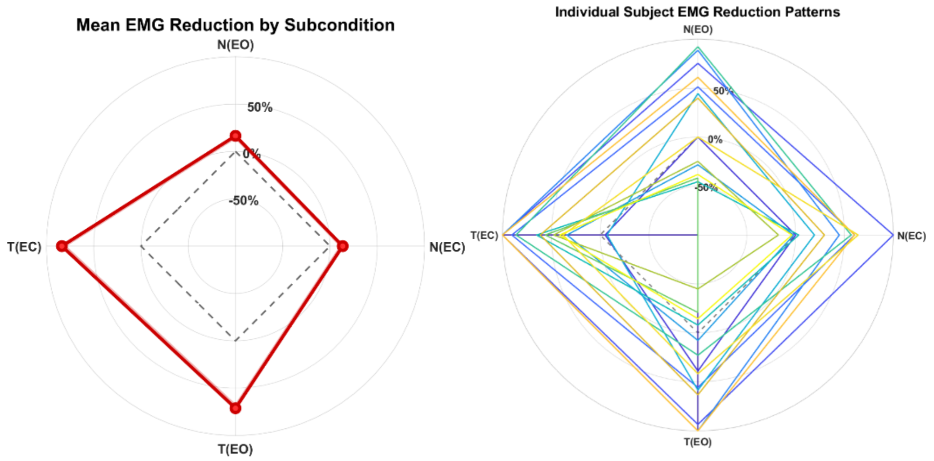
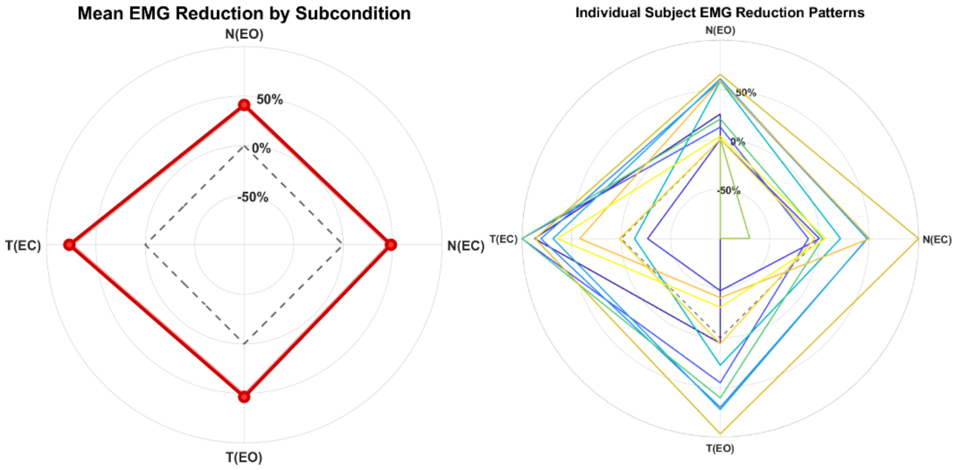


Figure C-15: Visualisation of EMG Reduction Based on Mean and Subject for Erecter Spinae (ES, L3, Right)



FigureC-16: Visualisation of EMG Reduction Based on Mean and Subject for Erecter Spinae (ES, L3, Left)

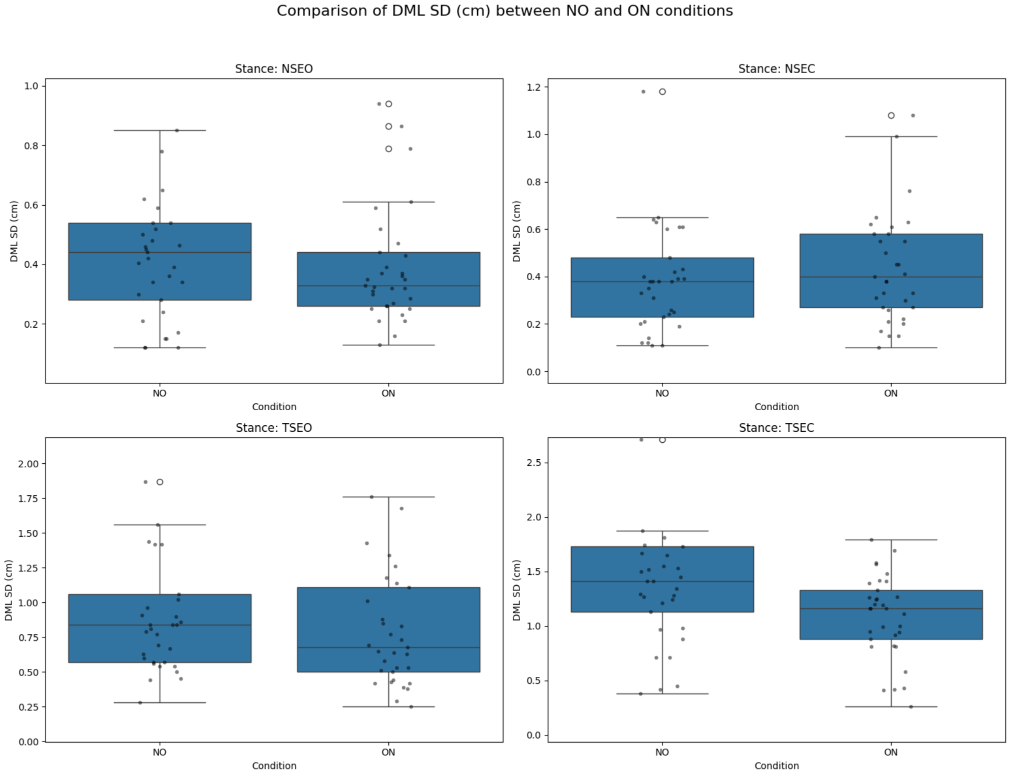


Figure C-17: Boxplots of DML SD Between Conditions With Outliers Retained

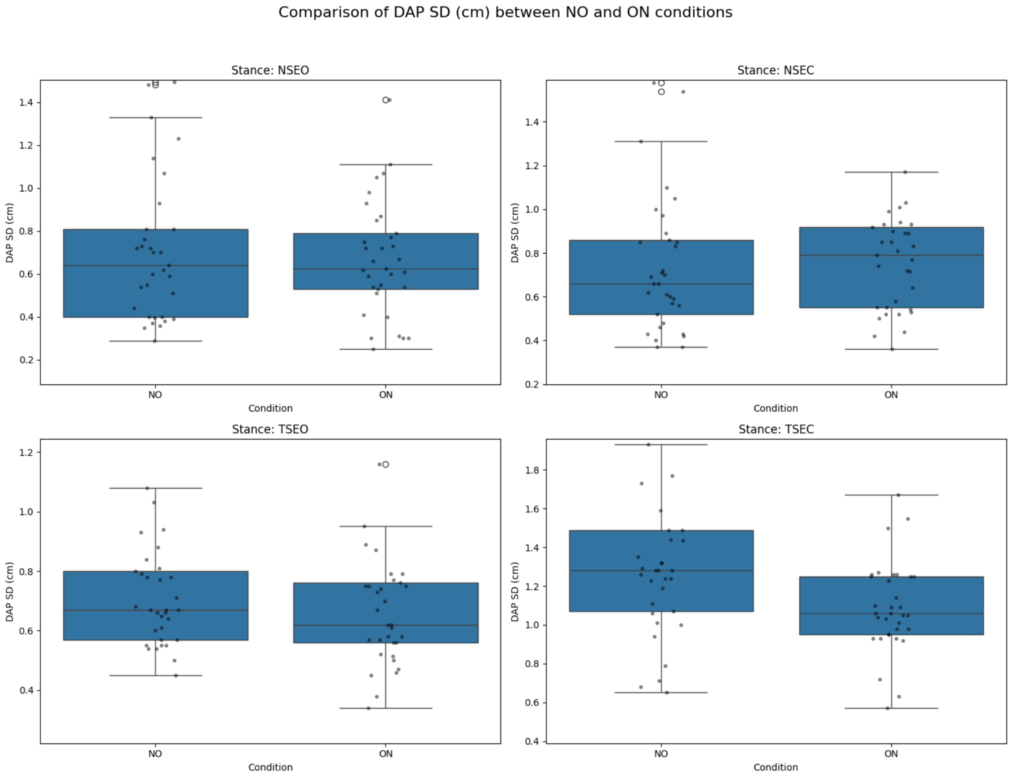


Figure C-18: Boxplots of DAP SD Between Conditions With Outliers Retained

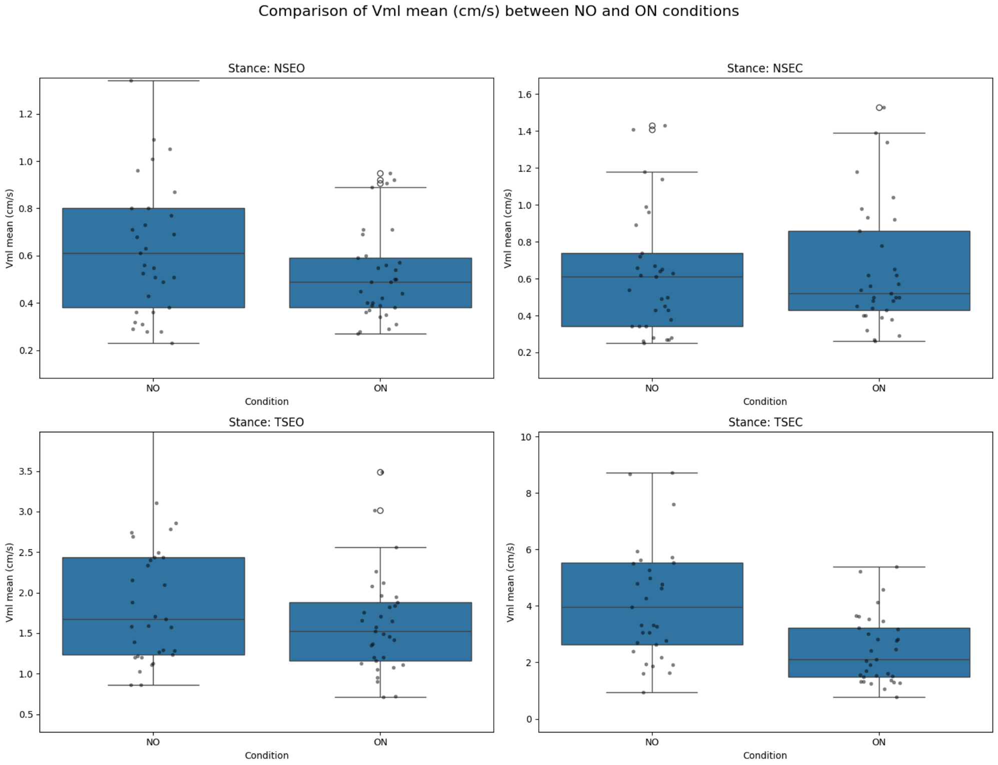


Figure C-19: Boxplots of Vml mean Between Conditions With Outliers Retained

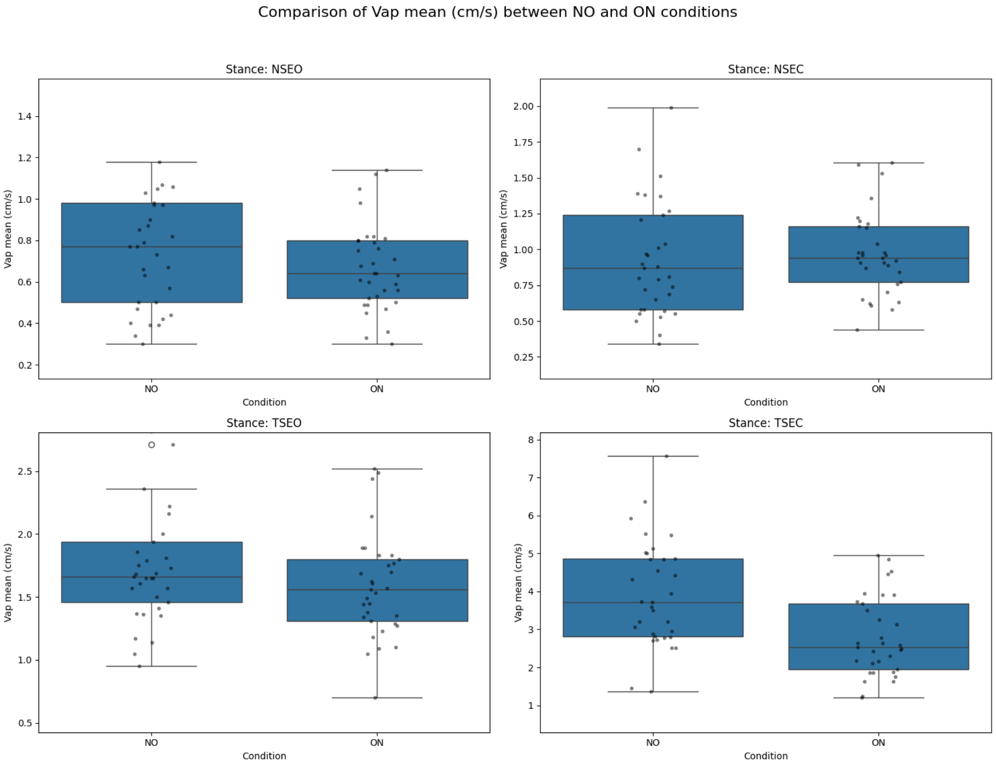


Figure C-20: Boxplots of Vap mean Between Conditions With Outliers Retained

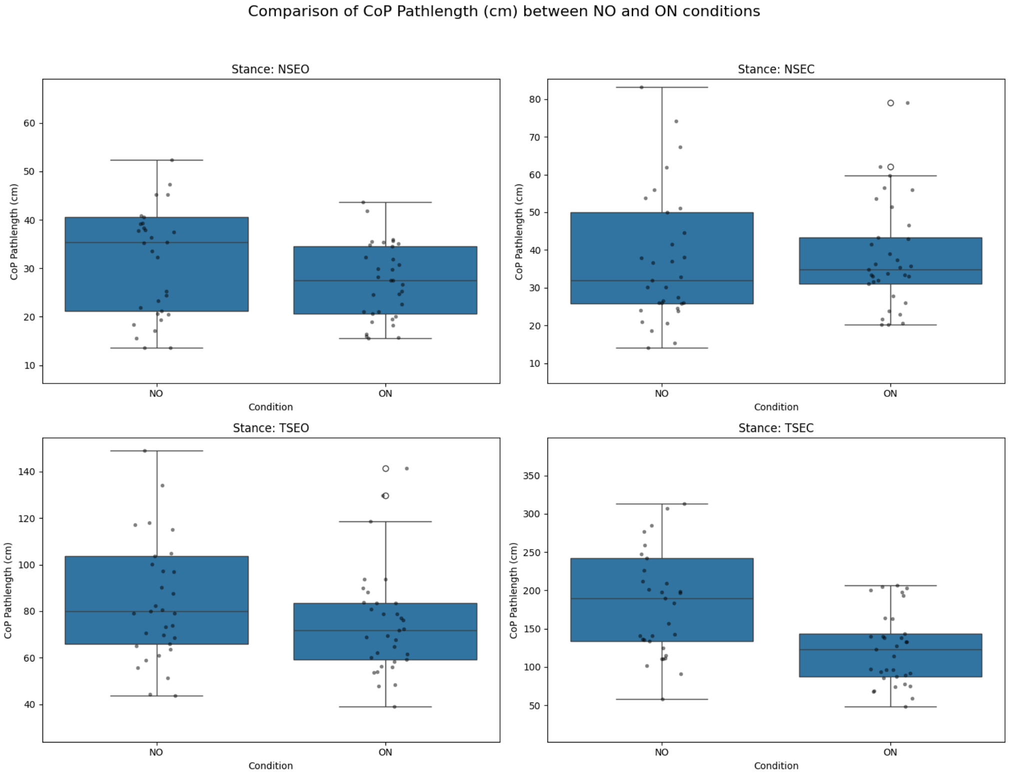


Figure C-21: Boxplots of Pathlength Between Conditions With Outliers Retained

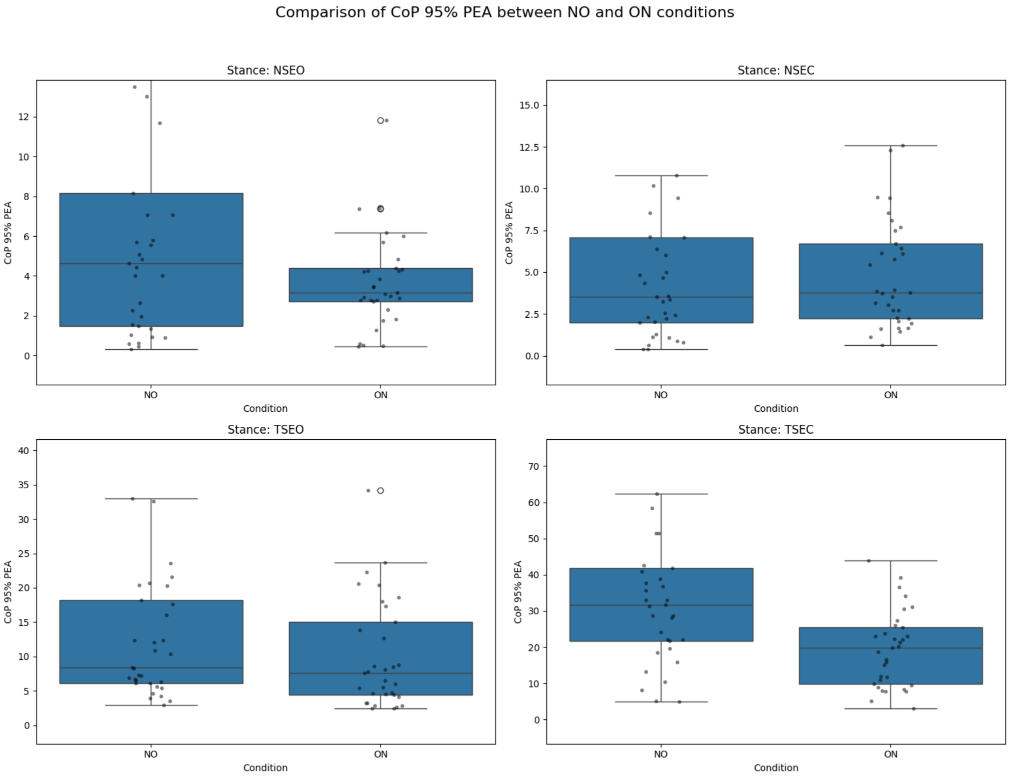


Figure C-22: Boxplots of 95% PEA Between Conditions With Outliers Retained

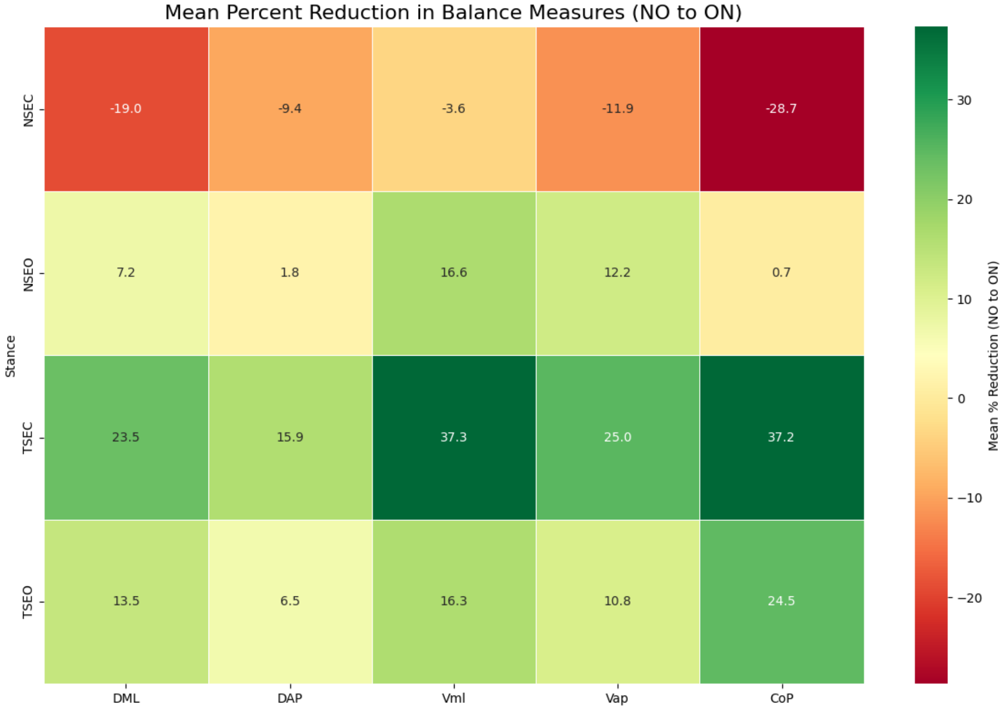


Figure C-23: Heatmap of Mean Percent Reduction in Parameters With Outliers Retained

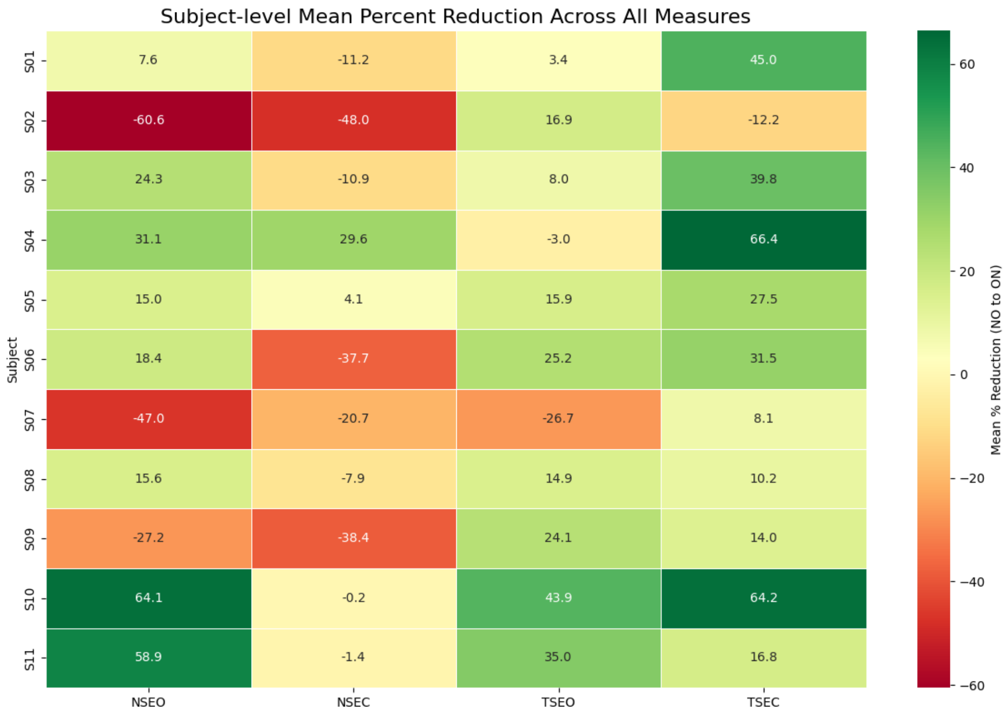


Figure C-24: Heatmap of Subject Level Percent Reduction Across Parameters With Outlier Retained

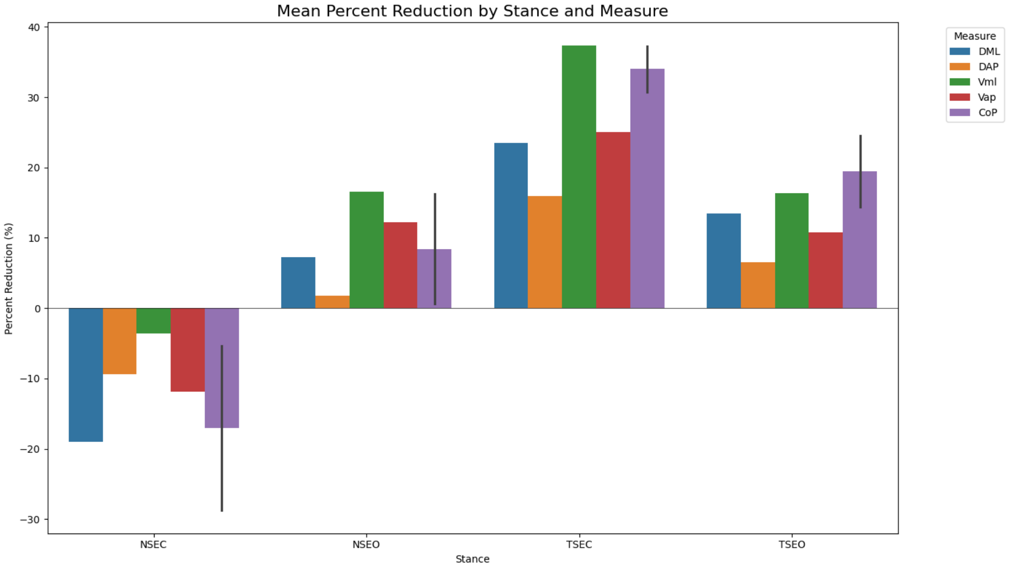


Figure C-25: Percent Reduction of Parameters by Stance With Outlier Retained

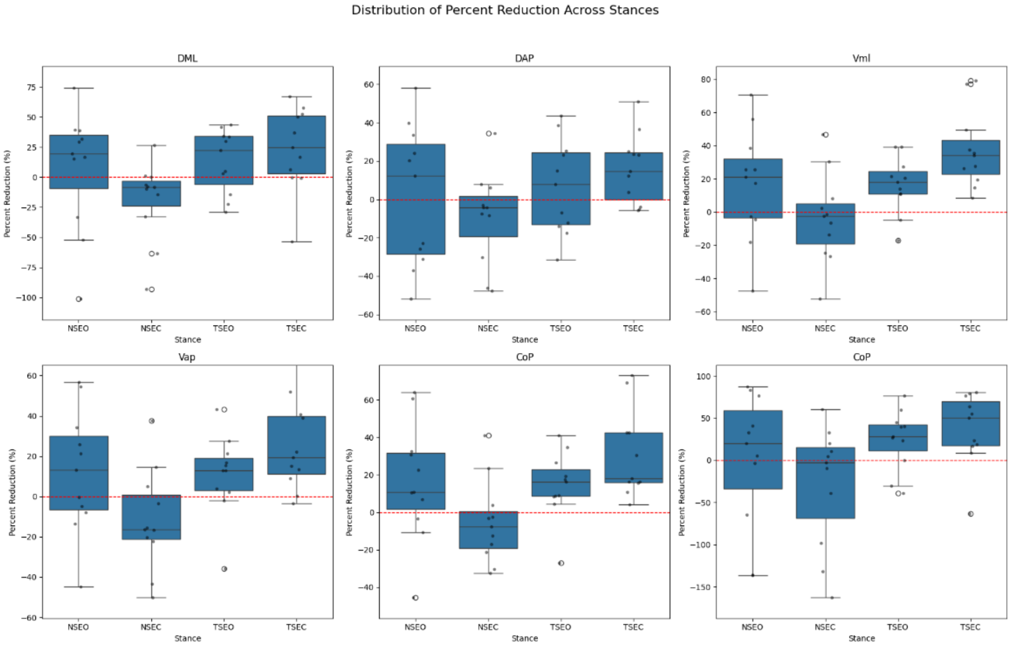


Figure C-26: Distribution of Percent Reduction Across Stances With Outliers Retained

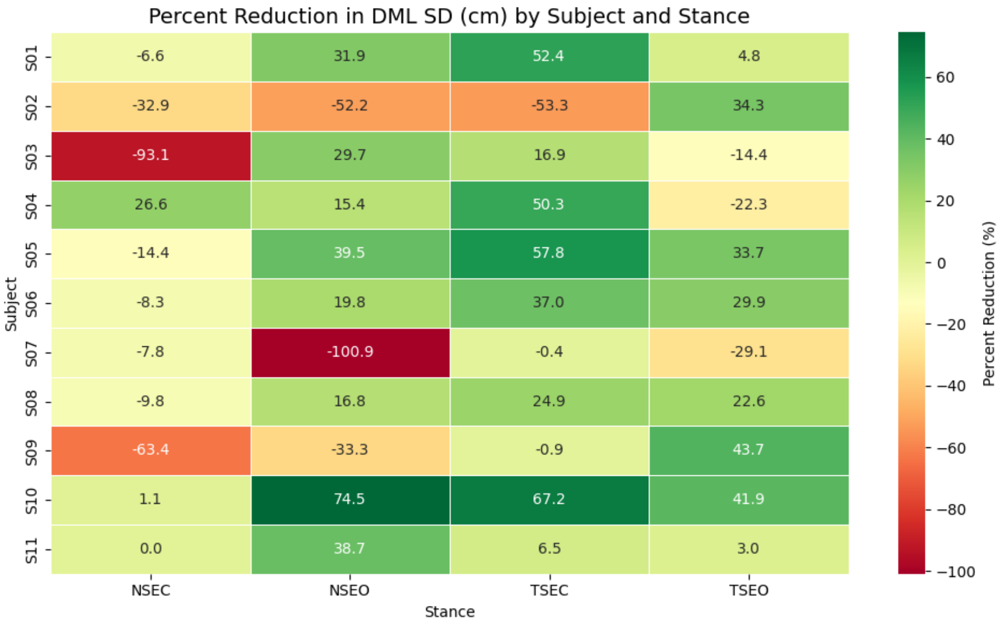


Figure C-27: Heatmap of Percent Reduction of DML SD by Subject With Outliers Retained

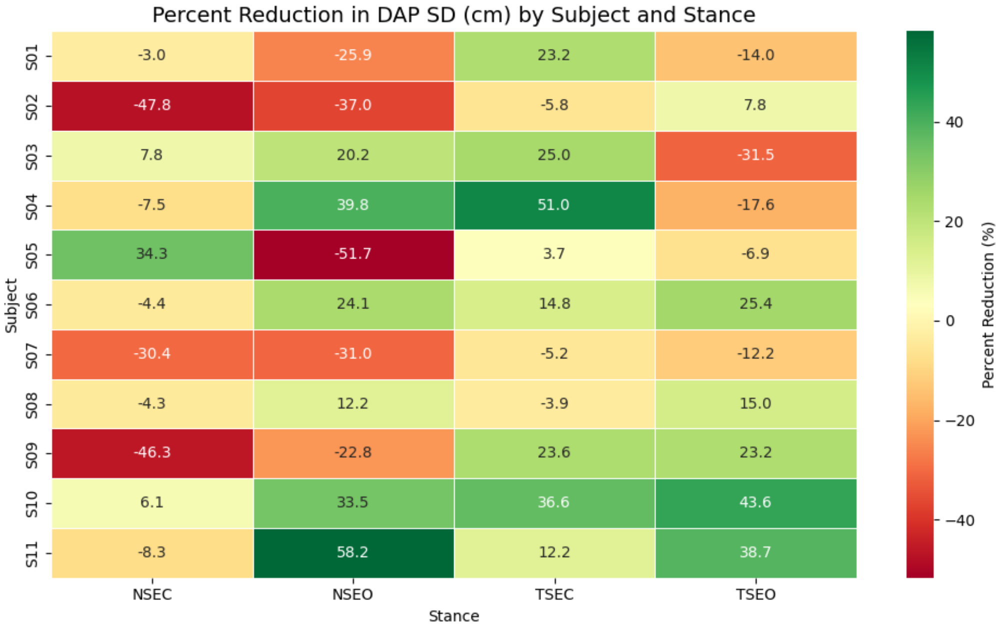


Figure C-28: Heatmap of Percent Reduction of DAP SD by Subject With Outliers Retained

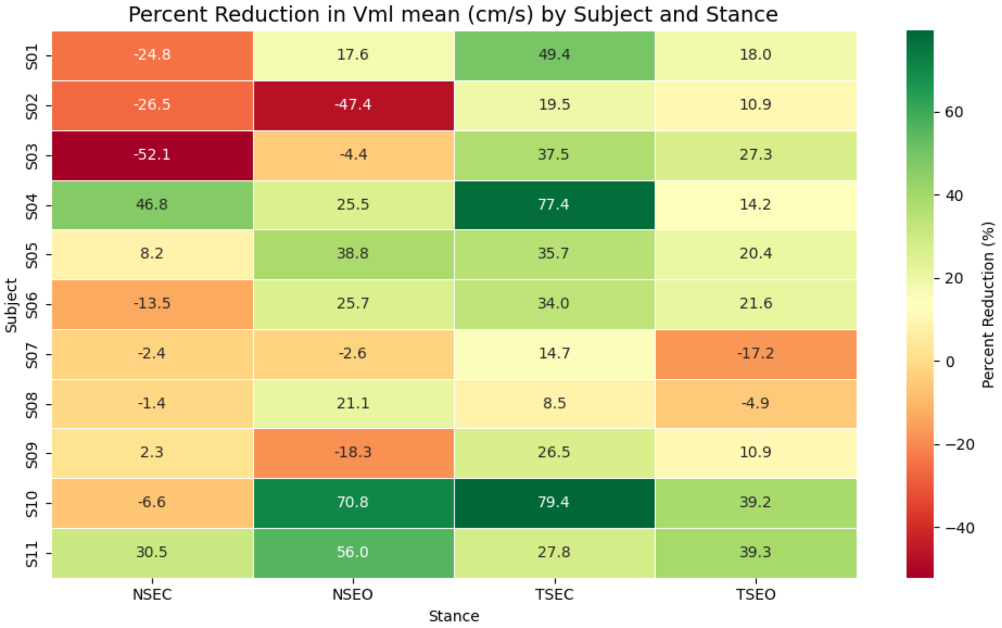


Figure C-29: Heatmap of Percent Reduction of Vml mean by Subject With Outliers Retained

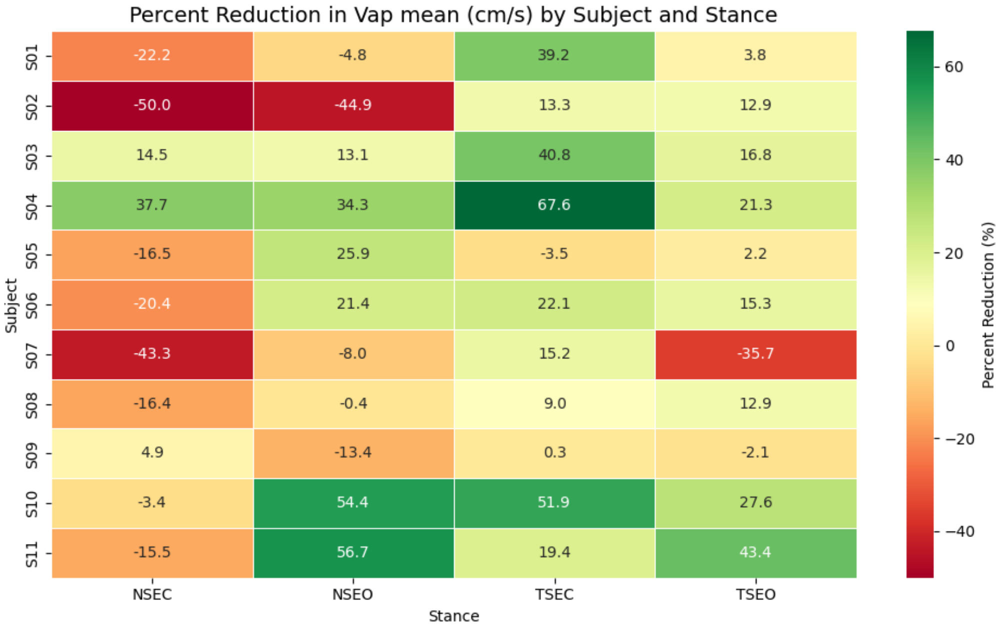


Figure C-30: Heatmap of Percent Reduction of Vap mean by Subject With Outliers Retained

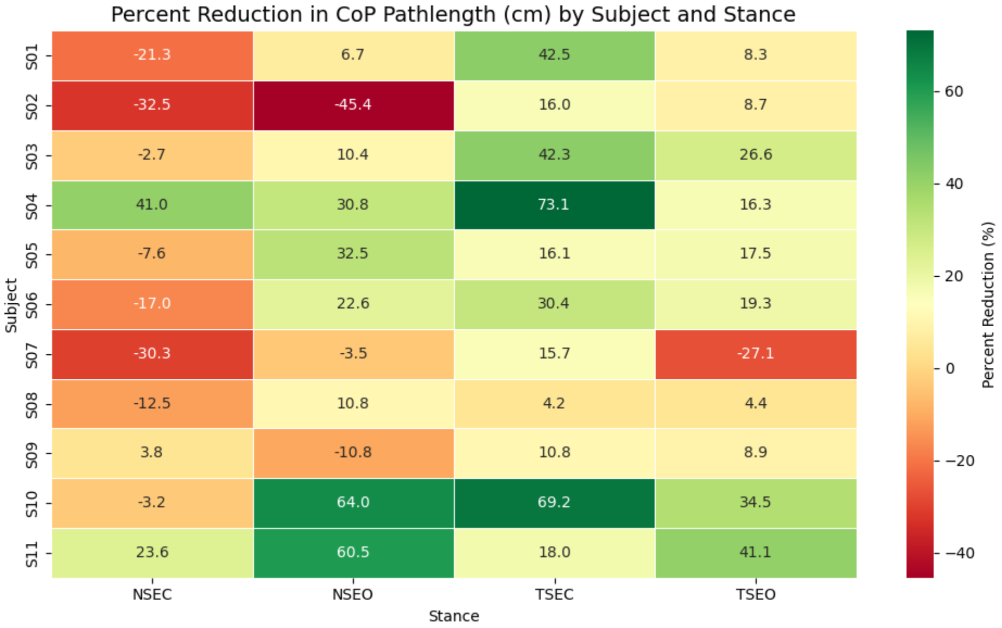
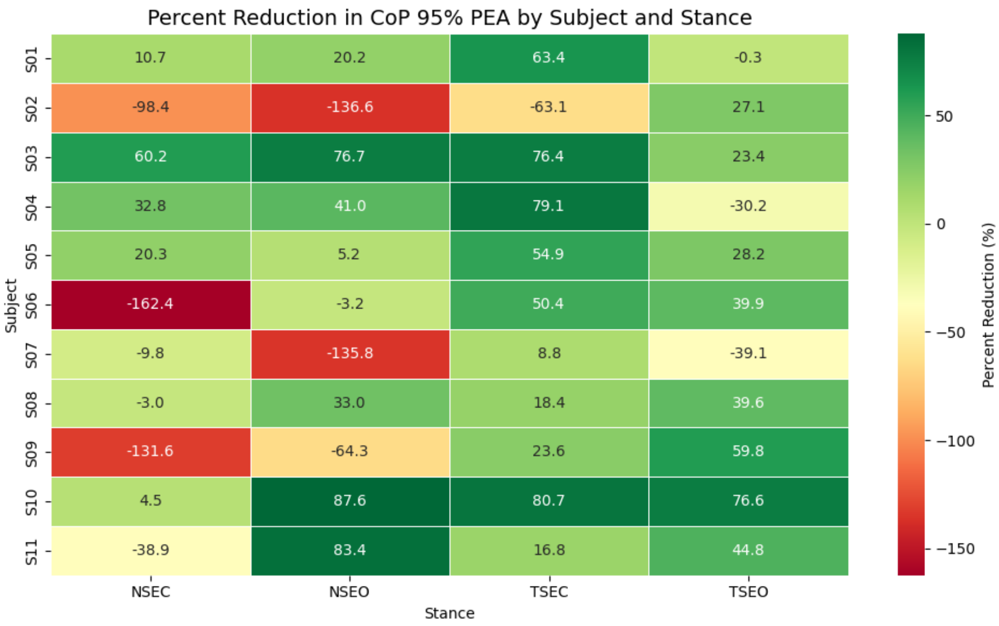
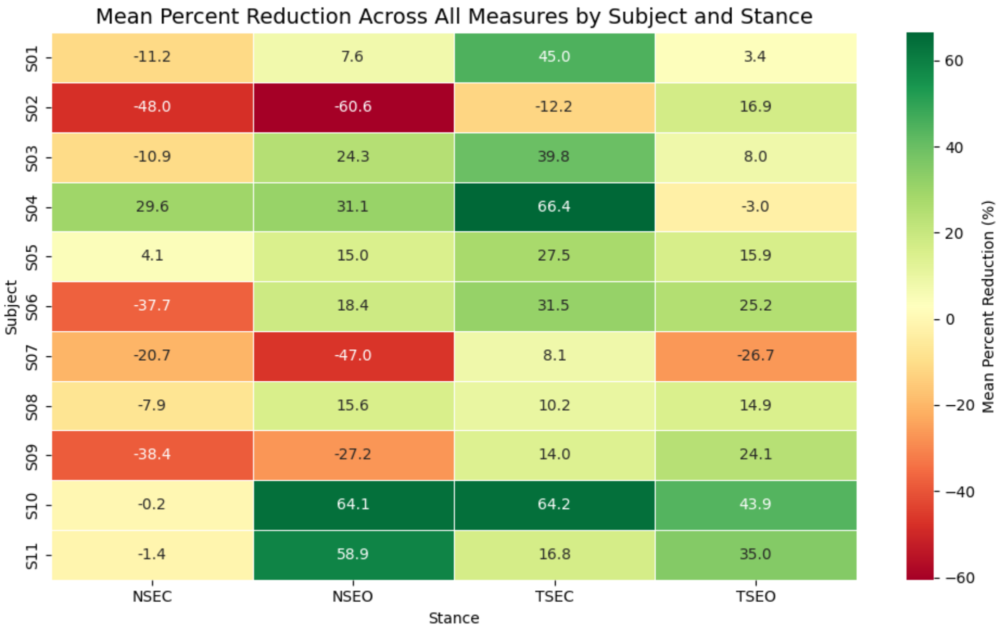


Figure C-31: Heatmap of Percent Reduction of Pathlength by Subject With Outliers Retained


Percent Reduction in CoP 95% PEA by Subject and Stance

Subject	NSEC	NSEO	TSEC	TSEO
S01	10.7	20.2	63.4	-0.3
S02	-98.4	-136.6	-63.1	27.1
S03	60.2	76.7	76.4	23.4
S04	32.8	41.0	79.1	-30.2
S05	20.3	5.2	54.9	28.2
S06	-162.4	-3.2	50.4	39.9
S07	-9.8	-135.8	8.8	-39.1
S08	-3.0	33.0	18.4	39.6
S09	-131.6	-64.3	23.6	59.8
S10	4.5	87.6	80.7	76.6
S11	-38.9	83.4	16.8	44.8

Figure C-32: Heatmap of Percent Reduction of 95% PEA by Subject With Outliers Retained


Mean Percent Reduction Across All Measures by Subject and Stance

Subject	NSEC	NSEO	TSEC	TSEO
S01	-11.2	7.6	45.0	3.4
S02	-48.0	-60.6	-12.2	16.9
S03	-10.9	24.3	39.8	8.0
S04	29.6	31.1	66.4	-3.0
S05	4.1	15.0	27.5	15.9
S06	-37.7	18.4	31.5	25.2
S07	-20.7	-47.0	8.1	-26.7
S08	-7.9	15.6	10.2	14.9
S09	-38.4	-27.2	14.0	24.1
S10	-0.2	64.1	64.2	43.9
S11	-1.4	58.9	16.8	35.0

Figure C-33: Heatmap of Percent Reduction of All Parameters by Subject With Outliers Retained

Appendix D: MATLAB Code with Some Sections Ommited Due to Privacy

```
%% Postural Sway Analysis and Threshold Optimization with ROC

% Parameters
dt = 0.034;  % Round to 3 decimal places for better numerical
stability
fs = round(1/dt);  % Round the sampling frequency
windowSize = round(5*fs);  % Ensure integer window size
windowOverlap = 0.5;      % 50% overlap for better transient
detection
CSF = 2.4477;             % 95% confidence scaling factor

% Outlier detection parameters
outlierOptions = struct();
outlierOptions.signalThreshold = 3.5;  % Z-score threshold for point
outliers
outlierOptions.windowQualityThreshold = 70; % Minimum percentage of
valid data per window
outlierOptions.featureZscoreThreshold = 3;  % Threshold for feature-
level outliers
outlierOptions.subjectOutlierThreshold = 2; % Threshold for subject-
level outliers
outlierOptions.waveletThreshold = 5;        % Threshold for wavelet-
based transient detection

% Butterworth filter design (0.01-0.60 Hz bandpass) [omitted, trivial]

% [section omitted – Data Handling, Subject Privacy]


    % ----------------- STEP 1: Signal-level Outlier Detection ------
-----------
    disp('  Detecting signal-level outliers...');

    % Use multiple methods to detect signal outliers
    % Method 1: Z-score based detection
    zX = zscore(accelX);
    zY = zscore(accelY);
    zZ = zscore(accelZ);

    outlierX_zscore = abs(zX) > outlierOptions.signalThreshold;
    outlierY_zscore = abs(zY) > outlierOptions.signalThreshold;
    outlierZ_zscore = abs(zZ) > outlierOptions.signalThreshold;

    % Method 2: Wavelet-based transient detection with improved error
handling
    % Decompose signal using wavelets to detect transients
    try
        % Initialize wavelet outputs as logical arrays
        outlierX_wavelet = false(size(accelX));
        outlierY_wavelet = false(size(accelY));
        outlierZ_wavelet = false(size(accelZ));

        % Try wavelet detection with error handling for each signal
        try
```

```
            % Use 'db4' wavelet with 4 levels for better noise
detection
            [cA, cD] = dwt(accelX, 'db4');
            % Detect large detail coefficients as potential outliers
            threshold = std(cD) * outlierOptions.waveletThreshold;
            outlierX_wavelet = abs(cD) > threshold;
            % Upsample to match original signal length
            outlierX_wavelet = interp1(1:length(outlierX_wavelet),
double(outlierX_wavelet), ...
                                     linspace(1,
length(outlierX_wavelet), length(accelX)), 'nearest');
            outlierX_wavelet = logical(outlierX_wavelet);
        catch wx
            disp(['  Warning: X-axis wavelet detection failed - '
wx.message]);
        end

        try
            [cA, cD] = dwt(accelY, 'db4');
            threshold = std(cD) * outlierOptions.waveletThreshold;
            outlierY_wavelet = abs(cD) > threshold;
            outlierY_wavelet = interp1(1:length(outlierY_wavelet),
double(outlierY_wavelet), ...
                                     linspace(1,
length(outlierY_wavelet), length(accelY)), 'nearest');
            outlierY_wavelet = logical(outlierY_wavelet);
        catch wy
            disp(['  Warning: Y-axis wavelet detection failed - '
wy.message]);
        end

        try
            [cA, cD] = dwt(accelZ, 'db4');
            threshold = std(cD) * outlierOptions.waveletThreshold;
            outlierZ_wavelet = abs(cD) > threshold;
            outlierZ_wavelet = interp1(1:length(outlierZ_wavelet),
double(outlierZ_wavelet), ...
                                     linspace(1,
length(outlierZ_wavelet), length(accelZ)), 'nearest');
            outlierZ_wavelet = logical(outlierZ_wavelet);
        catch wz
            disp(['  Warning: Z-axis wavelet detection failed - '
wz.message]);
        end

    catch e
        % If wavelet toolbox is not available or error occurs
        disp(['  Warning: Wavelet detection initialization failed - '
e.message]);
        outlierX_wavelet = false(size(accelX));
        outlierY_wavelet = false(size(accelY));
        outlierZ_wavelet = false(size(accelZ));
    end

    % Method 3: Moving median deviation
    windowLen = round(0.5 * fs);  % 0.5 second window
    medX = movmedian(accelX, windowLen);
    medY = movmedian(accelY, windowLen);
    medZ = movmedian(accelZ, windowLen);
    devX = abs(accelX - medX);
    devY = abs(accelY - medY);
    devZ = abs(accelZ - medZ);

    madX = movmedian(devX, windowLen);
    madY = movmedian(devY, windowLen);
    madZ = movmedian(devZ, windowLen);

    outlierX_mad = devX > 3.5 * madX;
    outlierY_mad = devY > 3.5 * madY;
    outlierZ_mad = devZ > 3.5 * madZ;

    % Combine all outlier detection methods
    outlierX = logical(outlierX_zscore) | outlierX_wavelet |
logical(outlierX_mad);
    outlierY = logical(outlierY_zscore) | outlierY_wavelet |
logical(outlierY_mad);
    outlierZ = logical(outlierZ_zscore) | outlierZ_wavelet |
logical(outlierZ_mad);

    % Combined outliers mask
    combinedOutliers = outlierX | outlierY | outlierZ;

    % Print outlier statistics
    percentOutliers = 100 * sum(combinedOutliers) /
length(combinedOutliers);
    disp(['    Found ' num2str(sum(combinedOutliers)) ' outlier
points (' num2str(percentOutliers, '%.2f') '%)']);

    % Replace outliers with NaN and then interpolate
    accelX_clean = accelX;
    accelY_clean = accelY;
    accelZ_clean = accelZ;

    accelX_clean(outlierX) = NaN;
    accelY_clean(outlierY) = NaN;
    accelZ_clean(outlierZ) = NaN;

    % Interpolate missing values
    accelX_clean = fillmissing(accelX_clean, 'linear');
    accelY_clean = fillmissing(accelY_clean, 'linear');
    accelZ_clean = fillmissing(accelZ_clean, 'linear');

    % Save outlier statistics for this file
    outlierStats(f).filename = filename;
    outlierStats(f).subject = subject;
    outlierStats(f).condition = condition;
    outlierStats(f).totalPoints = length(accelX);
    outlierStats(f).outlierPoints = sum(combinedOutliers);
    outlierStats(f).outlierPercent = percentOutliers;

    % Process in overlapping windows for better transient detection
    % Calculate number of windows with overlap
    stepSize = round(windowSize * (1 - windowOverlap));  % Ensure
integer step size
    numWindows = floor((length(accelX) - windowSize) / stepSize) + 1;
    fileFeatures = table();
    validWindows = 0;
    % Check if we have enough data for at least one window
    if numWindows < 1
        disp('  Not enough data points for window analysis');
        continue;
    end

    for w = 1:numWindows
        % Extract window
        startIdx = floor((w-1)*stepSize) + 1;
        endIdx = min(floor(startIdx + windowSize - 1),
length(accelX));

        % Check window quality - percentage of non-outlier data
points
        windowOutliers = combinedOutliers(startIdx:endIdx);
        windowQuality = 100 * (1 - sum(windowOutliers) /
length(windowOutliers));

        % Skip windows with too many outliers
        if windowQuality < outlierOptions.windowQualityThreshold
            disp(['    Skipping window ' num2str(w) ' due to poor
data quality (' num2str(windowQuality, '%.1f') '%)']);
            continue;
        end

        validWindows = validWindows + 1;

        % Window data (use cleaned version)
        Ax_raw = accelX_clean(startIdx:endIdx);
        Ay_raw = accelY_clean(startIdx:endIdx);
        Az_raw = accelZ_clean(startIdx:endIdx);

        % Apply filter to raw accelerometer data
        Ax_filt = filtfilt(b, a, Ax_raw);
        Ay_filt = filtfilt(b, a, Ay_raw);
        Az_filt = filtfilt(b, a, Az_raw);

        % Calculate angular displacements
        theta_pitch = atan2(Az_filt, sqrt(Ay_filt.^2 + Ax_filt.^2));
        theta_roll = atan2(Ax_filt, sqrt(Ay_filt.^2 + Az_filt.^2));

        % Calculate DAP and DML using patient-specific height values
        DAP = H1 * tan(theta_pitch);
        DML = H2 * tan(theta_roll);

        % Calculate displacement magnitude
        dp = sqrt(DAP.^2 + DML.^2);

        % Calculate velocity
        VEL = abs(diff(dp))/(1/fs);
        VEL = [VEL; VEL(end)]; % Padding

        % Calculate decomposition
        ddap = abs(diff(DAP));
        ddml = abs(diff(DML));
        DEP = sqrt(ddap.^2 + ddml.^2);
        DEP = [DEP; DEP(end)]; % Padding
        % Extract JF features
        DAP_max = max(DAP);
        DAP_min = min(DAP);
        DML_max = max(DML);
        DML_min = min(DML);

        % Rectangle area
        Rs = (DAP_max - DAP_min) * (DML_max - DML_min);

        % 95% Confidence Ellipse Area
        sigma_AP = std(DAP);
        sigma_ML = std(DML);
        a = CSF * sigma_AP;
        b = CSF * sigma_ML;
        CEA_95 = pi * a * b;

        % RMS displacement
        D_RMS = sqrt(mean(dp.^2));

        % Displacement range
        DR = max(dp) - min(dp);

        % Additional features: AP and ML sway ranges
        AP_range = DAP_max - DAP_min;
        ML_range = DML_max - DML_min;

        % Path length features
        path_length = sum(sqrt(diff(DAP).^2 + diff(DML).^2));
        mean_velocity = path_length / (length(DAP)/fs);

        % Store features for this window with metadata
        windowFeatures = table(DAP_max, DAP_min, DML_max, DML_min, 
Rs, CEA_95, D_RMS, DR, ...
                             mean(VEL), max(VEL), mean(DEP), 
max(DEP), ...
                             AP_range, ML_range, path_length, 
mean_velocity, ...
                             categorical({subject}), 
categorical({condition}), windowQuality, ...
                             'VariableNames', {'DAP_max', 'DAP_min', 
'DML_max', 'DML_min', ...
                                             'Rs', 'CEA_95', 'D_RMS', 
'DR', ...
                                             'VEL_mean', 'VEL_max', 
'DEP_mean', 'DEP_max', ...
                                             'AP_range', 'ML_range', 
'path_length', 'mean_velocity', ...
                                             'Subject', 'Condition', 
'WindowQuality'});
        fileFeatures = [fileFeatures; windowFeatures];

    % Update outlier stats with window information
    outlierStats(f).totalWindows = numWindows;
    outlierStats(f).validWindows = validWindows;
    outlierStats(f).percentValidWindows = 100 * validWindows / 
numWindows;

    % Only save features if we found valid windows
    if validWindows > 0
```

```
        % Store features for entire file
        try
            writetable(fileFeatures, fullfile(baseDir, [filename
'_features.csv']));
        catch e
            disp(['  Warning: Could not save features CSV - '
e.message]);
            % Try a simpler filename
            writetable(fileFeatures, fullfile(baseDir, ['features_'
num2str(f) '.csv']));
        end

        % Add to collection of all features
        allFeatures = [allFeatures; fileFeatures];
    else
        disp(['Warning: No valid data windows found in ' filename]);
    end
end

% Save outlier statistics
try
    save(fullfile(baseDir, 'outlier_statistics.mat'),
'outlierStats');
    writetable(struct2table(outlierStats), fullfile(baseDir,
'outlier_statistics.csv'));
catch e
    disp(['Warning: Could not save outlier statistics - '
e.message]);
end

disp(['Processed ' num2str(length(filenames)) ' files. Found '
num2str(height(allFeatures)) ' valid windows.']);

% Check if we have enough data to continue
if height(allFeatures) < 10
    error('Not enough valid data windows for analysis. Check input
data and parameters.');
end

%% STEP 2: Feature-level Outlier Detection
disp('Detecting feature-level outliers...');

% Feature columns to check for outliers
featureCols = {'DAP_max', 'DAP_min', 'DML_max', 'DML_min', 'Rs',
'CEA_95', 'D_RMS', 'DR', 'VEL_mean', 'VEL_max', 'AP_range',
'ML_range'};

% Get Z-scores for all numeric features
featureZScores = zeros(height(allFeatures), length(featureCols));
for i = 1:length(featureCols)
    featureZScores(:,i) = zscore(allFeatures.(featureCols{i}));
end

% Flag rows with extreme values in any feature
maxAbsZScores = max(abs(featureZScores), [], 2);
featureOutliers = maxAbsZScores >
outlierOptions.featureZscoreThreshold;

% Display feature outlier counts
```

```
disp(['  Found ' num2str(sum(featureOutliers)) ' feature-level
outliers out of ' num2str(height(allFeatures)) ' windows ('
num2str(100*sum(featureOutliers)/height(allFeatures), '%.2f') '%)']);

% Mark feature outliers in the dataset
allFeatures.FeatureOutlier = featureOutliers;

% Visualize feature distributions with outliers
featureFig = figure('Name', 'Feature Distributions with Outliers',
'NumberTitle', 'off');
for i = 1:length(featureCols)
    subplot(ceil(length(featureCols)/2), 2, i);
    
    % Non-outliers in blue, outliers in red
    boxplot(allFeatures.(featureCols{i}));
    hold on;
    
    % Highlight outliers
    featureOutlierIdx = abs(featureZScores(:,i)) >
outlierOptions.featureZscoreThreshold;
    scatter(ones(sum(featureOutlierIdx),1),
allFeatures.(featureCols{i})(featureOutlierIdx), 'r', 'filled');
    
    title(featureCols{i});
    grid on;
end
sgtitle('Feature Distributions with Outliers');

% Save feature distribution figure
try
    saveas(featureFig, fullfile(plotsDir,
'feature_distributions.png'));
    saveas(featureFig, fullfile(plotsDir,
'feature_distributions.fig'));
catch e
    disp(['Warning: Could not save feature distributions figure - '
e.message]);
    print(featureFig, fullfile(plotsDir,
'feature_distributions.png'), '-dpng');
end

%% STEP 3: Subject-level Outlier Detection
disp('Detecting subject-level outliers...');

% Get unique subjects
subjects = unique(allFeatures.Subject);
numSubjects = length(subjects);

% Calculate mean feature values for each subject
subjectFeatures = table();
for i = 1:numSubjects
    subjectIdx = allFeatures.Subject == subjects(i);
    
    % Skip subjects with feature outliers
    validSubjectIdx = subjectIdx & ~featureOutliers;
    
    if sum(validSubjectIdx) == 0
        disp(['  Warning: Subject ' char(subjects(i)) ' has no valid
data after feature outlier removal']);
```

```
        continue;
    end

    % Calculate mean of each feature for this subject
    subjectRow = table();
    subjectRow.Subject = subjects(i);
    for j = 1:length(featureCols)
        subjectRow.(featureCols{j}) =
mean(allFeatures.(featureCols{j})(validSubjectIdx));
    end

    subjectFeatures = [subjectFeatures; subjectRow];
end

% Calculate Z-scores for subject-level features
subjectZScores = zeros(height(subjectFeatures), length(featureCols));
for i = 1:length(featureCols)
    subjectZScores(:,i) = zscore(subjectFeatures.(featureCols{i}));
end

% Identify subject outliers
maxSubjectZScores = max(abs(subjectZScores), [], 2);
subjectOutliers = maxSubjectZScores >
outlierOptions.subjectOutlierThreshold;
outlierSubjects = subjectFeatures.Subject(subjectOutliers);

% Display subject outlier information
disp(['  Found ' num2str(sum(subjectOutliers)) ' outlier subjects out
of ' num2str(numSubjects) ' total subjects']);
if sum(subjectOutliers) > 0
    disp('  Outlier subjects:');
    for i = 1:length(outlierSubjects)
        disp(['    - ' char(outlierSubjects(i))]);
    end
end

% Mark subject outliers in the dataset
allFeatures.SubjectOutlier = false(height(allFeatures), 1);
for i = 1:length(outlierSubjects)
    allFeatures.SubjectOutlier(allFeatures.Subject ==
outlierSubjects(i)) = true;
end

% Visualize subject outliers
if height(subjectFeatures) > 3
    figure;
    [coeff, score, ~] = pca(subjectZScores);
    scatter(score(:,1), score(:,2), 50, ~subjectOutliers, 'filled');
    hold on;

    % Label outlier subjects
    for i = find(subjectOutliers)'
        text(score(i,1), score(i,2),
char(subjectFeatures.Subject(i)), 'FontSize', 8);
    end

    xlabel('Principal Component 1');
    ylabel('Principal Component 2');
    title('Subject Feature Space (PCA) with Outliers');
    colormap([1 0 0; 0 0 1]); % Red for outliers, blue for normal
    colorbar('YTick', [0.25 0.75], 'YTickLabel', {'Outlier',
'Normal'});
end

%% Create clean dataset (remove all outliers)
cleanFeatures = allFeatures(~allFeatures.FeatureOutlier &
~allFeatures.SubjectOutlier, :);

disp(['Final clean dataset: ' num2str(height(cleanFeatures)) '
windows (' ...
      num2str(100*height(cleanFeatures)/height(allFeatures), '%.2f')
'% of original data)']);

% Save clean features
writetable(cleanFeatures, [path filesep 'clean_features.csv']);

% Save all features with outlier flags
writetable(allFeatures, [path filesep
'all_features_with_outliers.csv']);

%% Label data based on condition information (if available)
disp('Assigning stability classes based on condition labels...');

% Try to extract true stability class from condition names
% Assuming conditions have names that indicate stability state
% e.g., "stable", "ap_sway", "ml_sway", "unstable"
conditionLabels = unique(cleanFeatures.Condition);
disp(['Found conditions: ' strjoin(cellstr(conditionLabels), ', ')]);

% Check if we need to manually map conditions to classes
if length(conditionLabels) > 0
    useManualMapping = true;
    try
        % Try to automatically map (customize based on your data)
        classMap = containers.Map();

        % Iterate through condition labels and try to identify
patterns
        for i = 1:length(conditionLabels)
            cond = lower(char(conditionLabels(i)));

            if contains(cond, 'stab') && ~contains(cond, 'unstab')
                classMap(char(conditionLabels(i))) = 0; % Stable
            elseif contains(cond, 'ap') || contains(cond, 'ant') ||
contains(cond, 'post')
                classMap(char(conditionLabels(i))) = 1; % Antero-
Posterior
            elseif contains(cond, 'ml') || contains(cond, 'med') ||
contains(cond, 'lat')
                classMap(char(conditionLabels(i))) = 2; % Medio-
Lateral
            elseif contains(cond, 'unstab') || contains(cond, 'dist')
                classMap(char(conditionLabels(i))) = 3; % Unstable
            else
                useManualMapping = true;
                break;
            end
            useManualMapping = false;
        end
    catch
        useManualMapping = true;
    end

    % If automatic mapping failed, ask user
    if useManualMapping
        disp('Could not automatically map conditions to stability
classes.');
        disp('Please map each condition to a stability class:');
        disp('  0 = Stable');
        disp('  1 = Antero-Posterior Sway');
        disp('  2 = Medio-Lateral Sway');
        disp('  3 = Unstable');

        classMap = containers.Map();
        for i = 1:length(conditionLabels)
            defaultClass = 0; % Default to stable
            classNum = input(['Enter class for condition "'
char(conditionLabels(i)) '" (0-3): ']);
            if isempty(classNum) || classNum < 0 || classNum > 3
                classNum = defaultClass;
            end
            classMap(char(conditionLabels(i))) = classNum;
        end
    end

    % Map conditions to classes
    cleanFeatures.TrueClass = zeros(height(cleanFeatures), 1);
    for i = 1:height(cleanFeatures)
        condition = char(cleanFeatures.Condition(i));
        if isKey(classMap, condition)
            cleanFeatures.TrueClass(i) = classMap(condition);
        end
    end

    disp('Class labels assigned based on conditions.');
else
    disp('No clear condition information for labeling. Will proceed
with unsupervised approach.');
    % Use clustering to define "true" classes
    cleanFeatures.TrueClass = NaN(height(cleanFeatures), 1);
end

%% Apply clustering to determine stability classes
disp('Clustering data to determine stability classes...');

% Select key features for clustering
X = [cleanFeatures.D_RMS, cleanFeatures.AP_range,
cleanFeatures.ML_range, cleanFeatures.CEA_95];

% Standardize features
X = zscore(X);

% Apply K-means clustering (4 classes as in original algorithm)
rng(1); % For reproducibility
[clusterIdx, centroids] = kmeans(X, 4, 'Replicates', 10);
```

```
% Map cluster indices to sway classes (0=Stable, 1=AP, 2=ML,
3=Unstable)
% Find the most stable cluster (lowest RMS displacement)
[~, stableClusterIdx] = min(centroids(:,1)); % First feature is D_RMS

% Find AP dominant cluster (largest AP range)
[~, apClusterIdx] = max(centroids(:,2));

% Find ML dominant cluster (largest ML range)
[~, mlClusterIdx] = max(centroids(:,3));

% The remaining cluster is unstable
unstableClusterIdx = setdiff(1:4, [stableClusterIdx, apClusterIdx,
mlClusterIdx]);
if length(unstableClusterIdx) > 1
    % If there's ambiguity, use the cluster with highest RMS but not
largest in specific direction
    rmsValues = centroids(:,1);
    rmsValues([stableClusterIdx, apClusterIdx, mlClusterIdx]) = -Inf;
    [~, unstableClusterIdx] = max(rmsValues);
end

% Create mapping from cluster indices to sway classes
clusterToClass = zeros(4,1);
clusterToClass(stableClusterIdx) = 0;
clusterToClass(apClusterIdx) = 1;
clusterToClass(mlClusterIdx) = 2;
clusterToClass(unstableClusterIdx) = 3;

% Apply mapping to get sway classes
clusteredClass = clusterToClass(clusterIdx);

% Use TrueClass if available, otherwise use clustered class
if all(isnan(cleanFeatures.TrueClass))
    cleanFeatures.Class = clusteredClass;
else
    cleanFeatures.Class = cleanFeatures.TrueClass;
    % Also save clustered class for comparison
    cleanFeatures.ClusteredClass = clusteredClass;
end

% [Section Omitted due to trivial - Display clustering results]

%% ROC Analysis for Threshold Optimization with Leave-One-Class-Out
disp('Performing ROC analysis with leave-one-class-out threshold
optimization...');

% Create ROC figure
rocFigure = figure('Name', 'ROC Curves with Leave-One-Class-Out',
'Position', [100, 100, 1200, 800]);

% Features to analyze with ROC
rocFeatures = {'D_RMS', 'AP_range', 'ML_range', 'CEA_95'};
optimalThresholds = struct();

% Get unique classes
uniqueClasses = unique(cleanFeatures.Class);
numClasses = length(uniqueClasses);
```

```
% Perform One-vs-Rest ROC analysis for each class and feature with
leave-one-class-out
for classIdx = 0:3
    % Create subplot for this class
    subplot(2, 2, classIdx+1);
    
    % Title for this subplot
    title(['ROC for Class ' num2str(classIdx) ' ('
classNames{classIdx+1} ') with Leave-One-Out']);
    hold on;
    
    % Initialize arrays to store thresholds and performances for each
holdout
    leaveOneOutThresholds = zeros(numClasses, length(rocFeatures));
    leaveOneOutPerformance = zeros(numClasses, length(rocFeatures));
    
    % For each possible holdout class
    for holdoutClass = uniqueClasses'
        % Skip if holdout class is the current target class (we need
some positive examples)
        if holdoutClass == classIdx
            continue;
        end
        
        % Get training indices (all samples except holdout class)
        trainIndices = cleanFeatures.Class ~= holdoutClass;
        
        % One-vs-Rest encoding for training set
        binaryLabels = (cleanFeatures.Class(trainIndices) ==
classIdx);
        
        % Analyze each feature
        for featIdx = 1:length(rocFeatures)
            featureName = rocFeatures{featIdx};
            featureValues =
cleanFeatures.(featureName)(trainIndices);
            
            % Sort values for ROC analysis
            [sortedVals, sortIdx] = sort(featureValues);
            sortedLabels = binaryLabels(sortIdx);
            
            % Calculate TPR and FPR for different thresholds
            nPoints = length(sortedVals);
            TPR = zeros(nPoints, 1);
            FPR = zeros(nPoints, 1);
            
            % Determine if higher values indicate the class (true) or
not (false)
            isGreaterBetter = true;
            if (classIdx == 0) % For stable class, lower values are
better
                isGreaterBetter = false;
            elseif (classIdx == 1 && strcmp(featureName,
'AP_range')) % For AP class, higher AP_range is better
                isGreaterBetter = true;
            elseif (classIdx == 2 && strcmp(featureName,
'ML_range')) % For ML class, higher ML_range is better
                isGreaterBetter = true;
            elseif (classIdx == 3) % For unstable class, higher
values are better
                isGreaterBetter = true;
            else
                isGreaterBetter = false;
            end

            % Calculate ROC points
            for i = 1:nPoints
                if isGreaterBetter
                    % Threshold: classify as positive if value ≥
threshold
                    predictions = featureValues >= sortedVals(i);
                else
                    % Threshold: classify as positive if value ≤
threshold
                    predictions = featureValues <= sortedVals(i);
                end

                % Calculate TPR and FPR with error handling for edge
cases
                TP = sum(predictions & binaryLabels);
                FP = sum(predictions & ~binaryLabels);
                TN = sum(~predictions & ~binaryLabels);
                FN = sum(~predictions & binaryLabels);

                % Handle division by zero
                if (TP + FN) == 0
                    TPR(i) = 0;
                else
                    TPR(i) = TP / (TP + FN);
                end

                if (FP + TN) == 0
                    FPR(i) = 0;
                else
                    FPR(i) = FP / (FP + TN);
                end
            end

            % Find optimal threshold (Youden's J statistic: max(TPR-
FPR))
            J = TPR - FPR;
            [maxJ, maxIdx] = max(J);
            optThreshold = sortedVals(maxIdx);

            % Store the threshold for this holdout
            leaveOneOutThresholds(holdoutClass+1, featIdx) =
optThreshold;

            % Evaluate on holdout data
            holdoutIndices = cleanFeatures.Class == holdoutClass;
            holdoutFeatureValues =
cleanFeatures.(featureName)(holdoutIndices);
            holdoutLabels = (cleanFeatures.Class(holdoutIndices) ==
classIdx);

            % Apply threshold to holdout
            if isGreaterBetter
```

```
                holdoutPredictions = holdoutFeatureValues >=
optThreshold;
            else
                holdoutPredictions = holdoutFeatureValues <=
optThreshold;
            end

            % Calculate accuracy on holdout
            holdoutAccuracy = sum(holdoutPredictions ==
holdoutLabels) / length(holdoutLabels);
            leaveOneOutPerformance(holdoutClass+1, featIdx) =
holdoutAccuracy;
        end
    end

    % For each feature, compute the mean threshold and evaluate
overall performance
    for featIdx = 1:length(rocFeatures)
        featureName = rocFeatures{featIdx};

        % Calculate mean threshold (excluding zeros which are from
skipped iterations)
        thresholds = leaveOneOutThresholds(:, featIdx);
        thresholds = thresholds(thresholds ~= 0);
        meanThreshold = mean(thresholds);

        % Apply mean threshold to full dataset
        featureValues = cleanFeatures.(featureName);
        fullBinaryLabels = (cleanFeatures.Class == classIdx);

        % Determine if higher values indicate the class (true) or not
(false)
        isGreaterBetter = true;
        if (classIdx == 0) % For stable class, lower values are
better
            isGreaterBetter = false;
        elseif (classIdx == 1 && strcmp(featureName, 'AP_range')) %
For AP class, higher AP_range is better
            isGreaterBetter = true;
        elseif (classIdx == 2 && strcmp(featureName, 'ML_range')) %
For ML class, higher ML_range is better
            isGreaterBetter = true;
        elseif (classIdx == 3) % For unstable class, higher values
are better
            isGreaterBetter = true;
        else
            isGreaterBetter = false;
        end

        % Calculate ROC for the full dataset using varying thresholds
        threshRange = linspace(min(featureValues),
max(featureValues), 100);
        fullTPR = zeros(length(threshRange), 1);
        fullFPR = zeros(length(threshRange), 1);

        for i = 1:length(threshRange)
            threshold = threshRange(i);

            if isGreaterBetter
```

```
                predictions = featureValues >= threshold;
            else
                predictions = featureValues <= threshold;
            end

            % Calculate TPR and FPR
            TP = sum(predictions & fullBinaryLabels);
            FP = sum(predictions & ~fullBinaryLabels);
            TN = sum(~predictions & ~fullBinaryLabels);
            FN = sum(~predictions & fullBinaryLabels);

            % Handle division by zero
            if (TP + FN) == 0
                fullTPR(i) = 0;
            else
                fullTPR(i) = TP / (TP + FN);
            end

            if (FP + TN) == 0
                fullFPR(i) = 0;
            else
                fullFPR(i) = FP / (FP + TN);
            end
        end

        % Plot ROC curve
        plot(fullFPR, fullTPR, 'LineWidth', 2, 'DisplayName',
featureName);

        % Calculate AUC
        AUC = trapz(fullFPR, fullTPR);

        % Store optimal threshold
        if ~isfield(optimalThresholds, featureName)
            optimalThresholds.(featureName) = zeros(4,1);
        end
        optimalThresholds.(featureName)(classIdx+1) = meanThreshold;

        % Find performance metrics at mean threshold
        if isGreaterBetter
            predictions = featureValues >= meanThreshold;
        else
            predictions = featureValues <= meanThreshold;
        end

        TP = sum(predictions & fullBinaryLabels);
        FP = sum(predictions & ~fullBinaryLabels);
        TN = sum(~predictions & ~fullBinaryLabels);
        FN = sum(~predictions & fullBinaryLabels);

        if (TP + FN) > 0
            optTPR = TP / (TP + FN);
        else
            optTPR = 0;
        end

        if (FP + TN) > 0
            optFPR = FP / (FP + TN);
        else
```

```
            optFPR = 0;
        end
        
        % Mark optimal threshold on the curve
        plot(optFPR, optTPR, 'ro', 'MarkerSize', 8, 
'MarkerFaceColor', 'r');
        
        % Display results
        disp(['  Class ' num2str(classIdx) ' - ' featureName ': AUC = 
' num2str(AUC, '%.3f') ...
            ', Leave-One-Out Threshold = ' num2str(meanThreshold, 
'%.4f') ...
            ' (TPR = ' num2str(optTPR, '%.3f') ...
            ', FPR = ' num2str(optFPR, '%.3f') ')']);
        
        % Also print the variation in thresholds
        disp(['    Threshold std: ' num2str(std(thresholds), 
'%.4f') ...
              ', min: ' num2str(min(thresholds), '%.4f') ...
              ', max: ' num2str(max(thresholds), '%.4f')]);
    end
    
    % Add reference line
    plot([0,1], [0,1], 'k--', 'DisplayName', 'Random');
    xlabel('False Positive Rate');
    ylabel('True Positive Rate');
    grid on;
    legend('show', 'Location', 'southeast');
    hold off;
end

%% Feature Importance Analysis based on ROC
disp('Calculating feature importance using RI metric...');

% Get unique features
featureNames = rocFeatures;
classes = unique(cleanFeatures.Class);
numClasses = length(classes);

% Create structure for feature importance
featureImportance = struct();

% Calculate Relative Importance (RI) for each feature
for i = 1:length(featureNames)
    featureName = featureNames{i};
    featureValues = cleanFeatures.(featureName);
    
    % Calculate RI for each class
    for j = 1:numClasses
        classIdx = classes(j);
        className = classNames{classIdx+1};
        
        % Get threshold for this feature and class from optimal 
thresholds
        Jth = optimalThresholds.(featureName)(classIdx+1);
        
        % Calculate |JF_i,q - Jth_q| / max(JF_i,q - Jth_q) for each 
sample
        diffValues = abs(featureValues - Jth);
        maxDiff = max(diffValues);

        % Handle case where maxDiff is 0 to avoid division by zero
        if maxDiff == 0
            normalizedDiffs = zeros(size(diffValues));
        else
            normalizedDiffs = diffValues / maxDiff;
        end

        % Calculate RI_q: RI_q% = 100 * (1/N * sum(...))
        RI = 100 * (1 - mean(normalizedDiffs));

        % Store RI value
        if ~isfield(featureImportance, featureName)
            featureImportance.(featureName) = zeros(numClasses, 1);
        end
        featureImportance.(featureName)(classIdx+1) = RI;

        disp(['  Feature ' featureName ' importance for class '
className ': RI = ' num2str(RI, '%.2f') '%']);
    end
end

% Store best features with valid field names
bestFeatures = struct();
for j = 1:numClasses
    classIdx = classes(j);
    className = classNames{classIdx+1};
    % Convert class name to valid field name
    fieldName = matlab.lang.makeValidName(className);

    % Find feature with highest RI for this class
    maxRI = -Inf;
    bestFeature = '';

    for i = 1:length(featureNames)
        featureName = featureNames{i};
        if featureImportance.(featureName)(classIdx+1) > maxRI
            maxRI = featureImportance.(featureName)(classIdx+1);
            bestFeature = featureName;
        end
    end

    bestFeatures.(fieldName) = bestFeature;
    disp(['  Best feature for ' className ': ' bestFeature ' (RI = '
num2str(maxRI, '%.2f') '%)']);
end

% Save feature importance information
featureImportanceTable = struct2table(featureImportance);
writetable(featureImportanceTable, [path filesep
'feature_importance.csv']);

% Create feature importance visualization
figure('Name', 'Feature Importance by Class');
bar(cell2mat(struct2cell(featureImportance)'));
xticklabels(classNames);
legend(featureNames);
ylabel('Relative Importance (%)');
```

```
title('Feature Importance by Stability Class');
grid on;

%% training
%% Hyperparameter Tuning for Multiple Classifiers
disp('Performing hyperparameter tuning for each classifier...');

% Use only valid labeled data
validIdx = ~isnan(cleanFeatures.TrueClass);
X = cleanFeatures{validIdx, rocFeatures};
Y = cleanFeatures.TrueClass(validIdx);

% Set cross-validation settings for hyperparameter tuning
% We'll use fewer folds for tuning to speed up the process
tuningCVPartition = cvpartition(Y, 'KFold', 3);

% Initialize results
classifierNames = {};
numFolds = 5; % For final evaluation
allAccuracies = zeros(0, numFolds);
bestParams = {};
tuningResults = {};

% Set cross-validation settings for final evaluation
finalCVPartition = cvpartition(Y, 'KFold', numFolds);

%% 1. Decision Tree Tuning
disp('Tuning Decision Tree...');
% Define parameter grid for Decision Tree
dtParams = struct();
dtParams.MaxNumSplits = [5, 10, 20, 50, 100];
dtParams.MinLeafSize = [1, 5, 10, 20];

% Initialize storage for results
dtResults = zeros(length(dtParams.MaxNumSplits),
length(dtParams.MinLeafSize));

% Perform grid search
for i = 1:length(dtParams.MaxNumSplits)
    for j = 1:length(dtParams.MinLeafSize)
        maxSplits = dtParams.MaxNumSplits(i);
        minLeaf = dtParams.MinLeafSize(j);
        
        % Use cross-validation to evaluate this parameter combination
        cvAcc = zeros(1, tuningCVPartition.NumTestSets);
        for k = 1:tuningCVPartition.NumTestSets
            trainIdx = tuningCVPartition.training(k);
            testIdx = tuningCVPartition.test(k);
            
            dtModel = fitctree(X(trainIdx,:), Y(trainIdx), ...
                'MaxNumSplits', maxSplits, 'MinLeafSize', minLeaf);
            
            predictions = predict(dtModel, X(testIdx,:));
            cvAcc(k) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
        end
        
        % Store average accuracy for this parameter combination
        dtResults(i, j) = mean(cvAcc);
    end
end

% Find best parameters
[maxVal, maxIdx] = max(dtResults(:));
[i_best, j_best] = ind2sub(size(dtResults), maxIdx);
bestDTMaxSplits = dtParams.MaxNumSplits(i_best);
bestDTMinLeaf = dtParams.MinLeafSize(j_best);

% Store tuning results
dtTuningResult = struct();
dtTuningResult.paramGrid = dtResults;
dtTuningResult.bestAccuracy = maxVal;
dtTuningResult.bestParams = struct('MaxNumSplits', bestDTMaxSplits,
'MinLeafSize', bestDTMinLeaf);
tuningResults{end+1} = dtTuningResult;

% Display best parameters
fprintf('Best Decision Tree parameters: MaxNumSplits=%d,
MinLeafSize=%d (CV Accuracy: %.4f)\n', ...
    bestDTMaxSplits, bestDTMinLeaf, maxVal);

% Evaluate best model on all folds
dtFoldAcc = zeros(1, numFolds);
for i = 1:numFolds
    trainIdx = finalCVPartition.training(i);
    testIdx = finalCVPartition.test(i);

    % Train with best parameters
    dtModel = fitctree(X(trainIdx,:), Y(trainIdx), ...
        'MaxNumSplits', bestDTMaxSplits, 'MinLeafSize',
bestDTMinLeaf);

    predictions = predict(dtModel, X(testIdx,:));
    dtFoldAcc(i) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
end

classifierNames{end+1} = 'Decision Tree (Tuned)';
allAccuracies(end+1,:) = dtFoldAcc;
bestParams{end+1} = sprintf('MaxSplits=%d, MinLeaf=%d',
bestDTMaxSplits, bestDTMinLeaf);

%% 2. KNN Tuning
disp('Tuning KNN...');
% Define parameter grid for KNN
knnParams = struct();
knnParams.NumNeighbors = [1, 3, 5, 7, 9, 11, 15];
knnParams.Distance = {'euclidean', 'cityblock', 'cosine',
'correlation'};
knnParams.Standardize = [true, false];

% Initialize storage for results
knnResults = zeros(length(knnParams.NumNeighbors),
length(knnParams.Distance), length(knnParams.Standardize));

% Perform grid search
for i = 1:length(knnParams.NumNeighbors)
    for j = 1:length(knnParams.Distance)
```

```
        for k = 1:length(knnParams.Standardize)
            numNeighbors = knnParams.NumNeighbors(i);
            distance = knnParams.Distance{j};
            standardize = knnParams.Standardize(k);

            % Use cross-validation to evaluate this parameter
combination
            cvAcc = zeros(1, tuningCVPartition.NumTestSets);
            for fold = 1:tuningCVPartition.NumTestSets
                trainIdx = tuningCVPartition.training(fold);
                testIdx = tuningCVPartition.test(fold);

                knnModel = fitcknn(X(trainIdx,:), Y(trainIdx), ...
                    'NumNeighbors', numNeighbors, 'Distance',
distance, 'Standardize', standardize);

                predictions = predict(knnModel, X(testIdx,:));
                cvAcc(fold) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
            end

            % Store average accuracy for this parameter combination
            knnResults(i, j, k) = mean(cvAcc);
        end
    end
end

% Find best parameters
[maxVal, maxIdx] = max(knnResults(:));
[i_best, j_best, k_best] = ind2sub(size(knnResults), maxIdx);
bestKnnNeighbors = knnParams.NumNeighbors(i_best);
bestKnnDistance = knnParams.Distance{j_best};
bestKnnStandardize = knnParams.Standardize(k_best);

% Store tuning results
knnTuningResult = struct();
knnTuningResult.paramGrid = knnResults;
knnTuningResult.bestAccuracy = maxVal;
knnTuningResult.bestParams = struct('NumNeighbors',
bestKnnNeighbors, ...
                                   'Distance', bestKnnDistance, ...
                                   'Standardize',
bestKnnStandardize);
tuningResults{end+1} = knnTuningResult;

% Display best parameters
fprintf('Best KNN parameters: NumNeighbors=%d, Distance=%s,
Standardize=%d (CV Accuracy: %.4f)\n', ...
    bestKnnNeighbors, bestKnnDistance, bestKnnStandardize, maxVal);

% Evaluate best model on all folds
knnFoldAcc = zeros(1, numFolds);
for i = 1:numFolds
    trainIdx = finalCVPartition.training(i);
    testIdx = finalCVPartition.test(i);

    % Train with best parameters
    knnModel = fitcknn(X(trainIdx,:), Y(trainIdx), ...
        'NumNeighbors', bestKnnNeighbors, 'Distance',
bestKnnDistance, 'Standardize', bestKnnStandardize);

    predictions = predict(knnModel, X(testIdx,:));
    knnFoldAcc(i) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
end

classifierNames{end+1} = sprintf('k-NN (k=%d, %s)', bestKnnNeighbors,
bestKnnDistance);
allAccuracies(end+1,:) = knnFoldAcc;
bestParams{end+1} = sprintf('k=%d, dist=%s, std=%d',
bestKnnNeighbors, bestKnnDistance, bestKnnStandardize);

%% 3. SVM Tuning
disp('Tuning SVM...');
% Define parameter grid for SVM
svmParams = struct();
svmParams.KernelFunction = {'linear', 'rbf', 'polynomial'};
svmParams.BoxConstraint = [0.1, 1, 10, 100];
svmParams.KernelScale = [0.1, 1, 10];
svmParams.Coding = {'onevsone', 'onevsall'};

% Initialize storage for results - this will be sparse because not
all combinations make sense
svmResults = zeros(length(svmParams.KernelFunction),
length(svmParams.BoxConstraint), ...
                 length(svmParams.KernelScale),
length(svmParams.Coding));

% Perform grid search - Using a simpler approach for SVM due to
computational intensity
bestSVMAcc = 0;
bestSVMKernel = '';
bestSVMBoxConstraint = 0;
bestSVMKernelScale = 0;
bestSVMCoding = '';

% We'll use a more focused search for SVM due to computational
complexity
for i = 1:length(svmParams.KernelFunction)
    for j = 1:length(svmParams.BoxConstraint)
        for k = 1:length(svmParams.KernelScale)
            for l = 1:length(svmParams.Coding)
                kernelFunc = svmParams.KernelFunction{i};
                boxConstraint = svmParams.BoxConstraint(j);
                kernelScale = svmParams.KernelScale(k);
                coding = svmParams.Coding{l};

                % Use cross-validation to evaluate this parameter
combination
                cvAcc = 0;

                % For SVM we'll just do a single fold to save time
during tuning
                fold = 1;
                trainIdx = tuningCVPartition.training(fold);
                testIdx = tuningCVPartition.test(fold);
                try
                    svmModel = fitcecoc(X(trainIdx,:), 
Y(trainIdx), ...
                        'Learners', templateSVM('KernelFunction', 
kernelFunc, ...
                                               'BoxConstraint', 
boxConstraint, ...
                                               'KernelScale', 
kernelScale), ...
                        'Coding', coding);

                    predictions = predict(svmModel, X(testIdx,:));
                    cvAcc = sum(predictions == Y(testIdx)) / 
length(Y(testIdx));

                    % Store result
                    svmResults(i, j, k, l) = cvAcc;

                    % Update best if better
                    if cvAcc > bestSVMAcc
                        bestSVMAcc = cvAcc;
                        bestSVMKernel = kernelFunc;
                        bestSVMBoxConstraint = boxConstraint;
                        bestSVMKernelScale = kernelScale;
                        bestSVMCoding = coding;
                    end
                catch
                    % Some parameter combinations might cause errors
                    svmResults(i, j, k, l) = 0;
                end
            end
        end
    end
end

% Store tuning results
svmTuningResult = struct();
svmTuningResult.paramGrid = svmResults;
svmTuningResult.bestAccuracy = bestSVMAcc;
svmTuningResult.bestParams = struct('KernelFunction', 
bestSVMKernel, ...
                                   'BoxConstraint', 
bestSVMBoxConstraint, ...
                                   'KernelScale', 
bestSVMKernelScale, ...
                                   'Coding', bestSVMCoding);
tuningResults{end+1} = svmTuningResult;

% Display best parameters
fprintf('Best SVM parameters: Kernel=%s, BoxConstraint=%.2f, 
KernelScale=%.2f, Coding=%s (CV Accuracy: %.4f)\n', ...
    bestSVMKernel, bestSVMBoxConstraint, bestSVMKernelScale, 
bestSVMCoding, bestSVMAcc);

% Evaluate best model on all folds
svmFoldAcc = zeros(1, numFolds);
for i = 1:numFolds
    trainIdx = finalCVPartition.training(i);
    testIdx = finalCVPartition.test(i);
    % Train with best parameters
    svmModel = fitcecoc(X(trainIdx,:), Y(trainIdx), ...
        'Learners', templateSVM('KernelFunction', bestSVMKernel, ...
                               'BoxConstraint',
bestSVMBoxConstraint, ...
                               'KernelScale',
bestSVMKernelScale), ...
        'Coding', bestSVMCoding);

    predictions = predict(svmModel, X(testIdx,:));
    svmFoldAcc(i) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
end

classifierNames{end+1} = sprintf('SVM (%s)', bestSVMKernel);
allAccuracies(end+1,:) = svmFoldAcc;
bestParams{end+1} = sprintf('Kernel=%s, C=%.1f, Scale=%.1f',
bestSVMKernel, bestSVMBoxConstraint, bestSVMKernelScale);

%% 4. Random Forest / Ensemble Tuning
disp('Tuning Random Forest...');
% Define parameter grid for Random Forest
rfParams = struct();
rfParams.Method = {'Bag', 'GentleBoost', 'LogitBoost', 'AdaBoostM2'};
rfParams.NumLearningCycles = [10, 50, 100, 200];
rfParams.LearnRate = [0.1, 0.5, 1.0]; % Only for boosting methods

% Initialize storage for results
rfResults = zeros(length(rfParams.Method),
length(rfParams.NumLearningCycles), length(rfParams.LearnRate));

% Perform grid search
bestRFAcc = 0;
bestRFMethod = '';
bestRFCycles = 0;
bestRFLearnRate = 0;

for i = 1:length(rfParams.Method)
    for j = 1:length(rfParams.NumLearningCycles)
        for k = 1:length(rfParams.LearnRate)
            method = rfParams.Method{i};
            cycles = rfParams.NumLearningCycles(j);
            learnRate = rfParams.LearnRate(k);

            % Skip LearnRate for Bagging
            if strcmp(method, 'Bag') && k > 1
                continue;
            end

            % Use cross-validation to evaluate this parameter
combination
            cvAcc = zeros(1, tuningCVPartition.NumTestSets);
            try
                for fold = 1:tuningCVPartition.NumTestSets
                    trainIdx = tuningCVPartition.training(fold);
                    testIdx = tuningCVPartition.test(fold);

                    if strcmp(method, 'Bag')
```

```
                        % Bagging doesn't use learning rate
                        rfModel = fitcensemble(X(trainIdx,:), 
Y(trainIdx), ...
                            'Method', method, 'NumLearningCycles', 
cycles);
                    else
                        % Boosting methods use learning rate
                        rfModel = fitcensemble(X(trainIdx,:), 
Y(trainIdx), ...
                            'Method', method, 'NumLearningCycles', 
cycles, 'LearnRate', learnRate);
                    end
                    
                    predictions = predict(rfModel, X(testIdx,:));
                    cvAcc(fold) = sum(predictions == Y(testIdx)) / 
length(Y(testIdx));
                end
                
                % Store average accuracy for this parameter 
combination
                meanAcc = mean(cvAcc);
                rfResults(i, j, k) = meanAcc;
                
                % Update best if better
                if meanAcc > bestRFAcc
                    bestRFAcc = meanAcc;
                    bestRFMethod = method;
                    bestRFCycles = cycles;
                    if ~strcmp(method, 'Bag')
                        bestRFLearnRate = learnRate;
                    end
                end
            catch
                % Some combinations might cause errors
                rfResults(i, j, k) = 0;
            end
        end
    end
end

% Store tuning results
rfTuningResult = struct();
rfTuningResult.paramGrid = rfResults;
rfTuningResult.bestAccuracy = bestRFAcc;
if strcmp(bestRFMethod, 'Bag')
    rfTuningResult.bestParams = struct('Method', bestRFMethod, 
'NumLearningCycles', bestRFCycles);
    paramStr = sprintf('Method=%s, Cycles=%d', bestRFMethod, 
bestRFCycles);
else
    rfTuningResult.bestParams = struct('Method', bestRFMethod, ...
                                       'NumLearningCycles', 
bestRFCycles, ...
                                       'LearnRate', bestRFLearnRate);
    paramStr = sprintf('Method=%s, Cycles=%d, LearnRate=%.1f', ...
                       bestRFMethod, bestRFCycles, bestRFLearnRate);
end
tuningResults{end+1} = rfTuningResult;
```

```
% Display best parameters
fprintf('Best Random Forest parameters: %s (CV Accuracy: %.4f)\n',
paramStr, bestRFAcc);

% Evaluate best model on all folds
rfFoldAcc = zeros(1, numFolds);
for i = 1:numFolds
    trainIdx = finalCVPartition.training(i);
    testIdx = finalCVPartition.test(i);

    % Train with best parameters
    if strcmp(bestRFMethod, 'Bag')
        rfModel = fitcensemble(X(trainIdx,:), Y(trainIdx), ...
            'Method', bestRFMethod, 'NumLearningCycles',
bestRFCycles);
    else
        rfModel = fitcensemble(X(trainIdx,:), Y(trainIdx), ...
            'Method', bestRFMethod, 'NumLearningCycles',
bestRFCycles, 'LearnRate', bestRFLearnRate);
    end

    predictions = predict(rfModel, X(testIdx,:));
    rfFoldAcc(i) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
end

classifierNames{end+1} = sprintf('Ensemble (%s)', bestRFMethod);
allAccuracies(end+1,:) = rfFoldAcc;
bestParams{end+1} = paramStr;

%% 5. Naive Bayes Tuning
disp('Tuning Naive Bayes...');
% Define parameter grid for Naive Bayes
nbParams = struct();
nbParams.DistributionNames = {
    'normal',  % Single distribution for all features
    'kernel',  % Single distribution for all features
    'mvmn'     % Multivariate multinormal distribution
};

% For per-predictor distributions (if needed)
% Get number of predictors
numPredictors = size(X, 2);

% Create cell arrays with per-predictor distributions
normalAll = repmat({'normal'}, 1, numPredictors);
kernelAll = repmat({'kernel'}, 1, numPredictors);
mixedDist = cell(1, numPredictors);
for i = 1:numPredictors
    if mod(i, 2) == 0
        mixedDist{i} = 'normal';
    else
        mixedDist{i} = 'kernel';
    end
end

% Add per-predictor distribution options
nbParams.DistributionNames{end+1} = normalAll;  % All normal
nbParams.DistributionNames{end+1} = kernelAll;  % All kernel
nbParams.DistributionNames{end+1} = mixedDist;  % Mixed normal/kernel

% Initialize storage for results
nbResults = zeros(length(nbParams.DistributionNames), 1);

% Perform grid search
for i = 1:length(nbParams.DistributionNames)
    distNames = nbParams.DistributionNames{i};

    % Use cross-validation to evaluate this parameter combination
    cvAcc = zeros(1, tuningCVPartition.NumTestSets);
    for fold = 1:tuningCVPartition.NumTestSets
        trainIdx = tuningCVPartition.training(fold);
        testIdx = tuningCVPartition.test(fold);

        % Try-catch to handle potential errors
        try
            nbModel = fitcnb(X(trainIdx,:), Y(trainIdx),
'DistributionNames', distNames);

            predictions = predict(nbModel, X(testIdx,:));
            cvAcc(fold) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
        catch ME
            % If error occurs, log it and set accuracy to 0
            fprintf('Error with distribution type %d: %s\n', i,
ME.message);
            cvAcc(fold) = 0;
        end
    end

    % Store average accuracy for this parameter
    nbResults(i) = mean(cvAcc);
end

% Find best parameters
[bestNBAcc, bestNBIdx] = max(nbResults);
bestNBDist = nbParams.DistributionNames{bestNBIdx};

% Store tuning results
nbTuningResult = struct();
nbTuningResult.paramGrid = nbResults;
nbTuningResult.bestAccuracy = bestNBAcc;
nbTuningResult.bestParams = struct('DistributionNames', bestNBDist);
tuningResults{end+1} = nbTuningResult;

% Display best parameters
if iscell(bestNBDist) && length(bestNBDist) > 1
    % For per-predictor distributions, summarize
    uniqueDists = unique(bestNBDist);
    distCounts = cellfun(@(x) sum(strcmp(bestNBDist, x)),
uniqueDists);
    distStr = '';
    for d = 1:length(uniqueDists)
        distStr = [distStr, sprintf('%s(%d) ', uniqueDists{d},
distCounts(d))];
    end
    distStr = ['Mixed: ', distStr];
elseif iscell(bestNBDist) && length(bestNBDist) == 1
```

```
    distStr = bestNBDist{1};
else
    distStr = bestNBDist;
end

fprintf('Best Naive Bayes parameters: Distribution=%s (CV
Accuracy: %.4f)\n', distStr, bestNBAcc);

% Evaluate best model on all folds
nbFoldAcc = zeros(1, numFolds);
for i = 1:numFolds
    trainIdx = finalCVPartition.training(i);
    testIdx = finalCVPartition.test(i);

    % Train with best parameters
    nbModel = fitcnb(X(trainIdx,:), Y(trainIdx), 'DistributionNames',
bestNBDist);

    predictions = predict(nbModel, X(testIdx,:));
    nbFoldAcc(i) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
end

classifierNames{end+1} = sprintf('Naive Bayes (%s)', distStr);
allAccuracies(end+1,:) = nbFoldAcc;
bestParams{end+1} = sprintf('Distribution=%s', distStr);

%% 6. LDA Tuning
disp('Tuning LDA...');

% Define parameter grid for LDA
ldaParams = struct();
% Full grid of discriminant types and gamma values (will be validated
for each combination)
ldaParams.DiscrimType = {'linear', 'quadratic', 'diagLinear',
'diagQuadratic'};
ldaParams.Gamma = [0, 0.25, 0.5, 0.75, 1];

% Initialize storage for results
ldaResults = zeros(length(ldaParams.DiscrimType),
length(ldaParams.Gamma));

% Perform grid search
for i = 1:length(ldaParams.DiscrimType)
    for j = 1:length(ldaParams.Gamma)
        discrimType = ldaParams.DiscrimType{i};
        gamma = ldaParams.Gamma(j);

        % Skip invalid combinations (quadratic types with gamma not 0
or 1)
        if (strcmp(discrimType, 'quadratic') || strcmp(discrimType,
'diagQuadratic')) && ...
                (gamma > 0 && gamma < 1)
            fprintf('Skipping invalid combination: %s with
gamma=%.2f\n', discrimType, gamma);
            ldaResults(i, j) = -Inf;  % Mark as invalid with -Inf
            continue;
        end
        % Use cross-validation to evaluate this parameter combination
        cvAcc = zeros(1, tuningCVPartition.NumTestSets);
        for fold = 1:tuningCVPartition.NumTestSets
            trainIdx = tuningCVPartition.training(fold);
            testIdx = tuningCVPartition.test(fold);
            
            try
                ldaModel = fitcdiscr(X(trainIdx,:), Y(trainIdx), ...
                                  'DiscrimType', discrimType, 
'Gamma', gamma);
                
                predictions = predict(ldaModel, X(testIdx,:));
                cvAcc(fold) = sum(predictions == Y(testIdx)) / 
length(Y(testIdx));
            catch ME
                fprintf('Error with %s discriminant, 
gamma=%.2f: %s\n', ...
                    discrimType, gamma, ME.message);
                cvAcc(fold) = 0;  % Set accuracy to 0 if an error 
occurs
            end
        end
        
        % Store average accuracy for this parameter combination
        if all(cvAcc == 0)
            ldaResults(i, j) = -Inf;  % Mark as invalid if all folds 
failed
        else
            ldaResults(i, j) = mean(cvAcc);
        end
    end
end

% Find best parameters (ignore -Inf values)
validResults = ldaResults;
validResults(validResults == -Inf) = -1;  % Convert -Inf to -1 for 
max function
[maxVal, maxIdx] = max(validResults(:));
[i_best, j_best] = ind2sub(size(validResults), maxIdx);
bestLDAType = ldaParams.DiscrimType{i_best};
bestLDAGamma = ldaParams.Gamma(j_best);

% Store tuning results
ldaTuningResult = struct();
ldaTuningResult.paramGrid = ldaResults;
ldaTuningResult.bestAccuracy = maxVal;
ldaTuningResult.bestParams = struct('DiscrimType', bestLDAType, 
'Gamma', bestLDAGamma);
tuningResults{end+1} = ldaTuningResult;

% Display best parameters
fprintf('Best LDA parameters: DiscrimType=%s, Gamma=%.2f (CV 
Accuracy: %.4f)\n', ...
    bestLDAType, bestLDAGamma, maxVal);

% Evaluate best model on all folds
ldaFoldAcc = zeros(1, numFolds);
for i = 1:numFolds
    trainIdx = finalCVPartition.training(i);
    testIdx = finalCVPartition.test(i);

    % Train with best parameters
    ldaModel = fitcdiscr(X(trainIdx,:), Y(trainIdx), ...
                       'DiscrimType', bestLDAType, 'Gamma',
bestLDAGamma);

    predictions = predict(ldaModel, X(testIdx,:));
    ldaFoldAcc(i) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
end

classifierNames{end+1} = sprintf('LDA (%s)', bestLDAType);
allAccuracies(end+1,:) = ldaFoldAcc;
bestParams{end+1} = sprintf('Type=%s, Gamma=%.2f', bestLDAType,
bestLDAGamma);

%% 7. Logistic Regression Tuning
disp('Tuning Logistic Regression...');
% Define parameter grid for Logistic Regression
logitParams = struct();
logitParams.Lambda = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1];
logitParams.Coding = {'onevsone', 'onevsall'};

% Initialize storage for results
logitResults = zeros(length(logitParams.Lambda),
length(logitParams.Coding));

% Perform grid search
for i = 1:length(logitParams.Lambda)
    for j = 1:length(logitParams.Coding)
        lambda = logitParams.Lambda(i);
        coding = logitParams.Coding{j};

        % Use cross-validation to evaluate this parameter combination
        cvAcc = zeros(1, tuningCVPartition.NumTestSets);
        for fold = 1:tuningCVPartition.NumTestSets
            trainIdx = tuningCVPartition.training(fold);
            testIdx = tuningCVPartition.test(fold);

            template = templateLinear('Learner', 'logistic',
'Lambda', lambda);
            logitModel = fitcecoc(X(trainIdx,:), Y(trainIdx), ...
                                'Learners', template, 'Coding',
coding);

            predictions = predict(logitModel, X(testIdx,:));
            cvAcc(fold) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
        end

        % Store average accuracy for this parameter combination
        logitResults(i, j) = mean(cvAcc);
    end
end

% Find best parameters
[maxVal, maxIdx] = max(logitResults(:));
[i_best, j_best] = ind2sub(size(logitResults), maxIdx);
```

```
bestLogitLambda = logitParams.Lambda(i_best);
bestLogitCoding = logitParams.Coding{j_best};

% Store tuning results
logitTuningResult = struct();
logitTuningResult.paramGrid = logitResults;
logitTuningResult.bestAccuracy = maxVal;
logitTuningResult.bestParams = struct('Lambda', bestLogitLambda,
'Coding', bestLogitCoding);
tuningResults{end+1} = logitTuningResult;

% Display best parameters
fprintf('Best Logistic Regression parameters: Lambda=%.6f, Coding=%s
(CV Accuracy: %.4f)\n', ...
    bestLogitLambda, bestLogitCoding, maxVal);

% Evaluate best model on all folds
logitFoldAcc = zeros(1, numFolds);
for i = 1:numFolds
    trainIdx = finalCVPartition.training(i);
    testIdx = finalCVPartition.test(i);

    % Train with best parameters
    template = templateLinear('Learner', 'logistic', 'Lambda',
bestLogitLambda);
    logitModel = fitcecoc(X(trainIdx,:), Y(trainIdx), ...
                        'Learners', template, 'Coding',
bestLogitCoding);

    predictions = predict(logitModel, X(testIdx,:));
    logitFoldAcc(i) = sum(predictions == Y(testIdx)) /
length(Y(testIdx));
end

classifierNames{end+1} = 'Logistic Regression (Tuned)';
allAccuracies(end+1,:) = logitFoldAcc;
bestParams{end+1} = sprintf('Lambda=%.6f, Coding=%s',
bestLogitLambda, bestLogitCoding);

%% Summarize tuned model results

% Calculate mean accuracies across folds
meanAccuracies = mean(allAccuracies, 2);
stdAccuracies = std(allAccuracies, 0, 2);

% Create detailed results table
foldNames = cell(1, numFolds);
for i = 1:numFolds
    foldNames{i} = sprintf('Fold%d', i);
end

% Create a table with fold-by-fold results
detailedResultTable = array2table(allAccuracies, 'VariableNames',
foldNames);
detailedResultTable.Classifier = classifierNames';
detailedResultTable.Mean = meanAccuracies;
detailedResultTable.StdDev = stdAccuracies;
detailedResultTable.BestParameters = bestParams';
```

```
% Reorder columns to put Classifier first
detailedResultTable = detailedResultTable(:, ['Classifier',
foldNames, {'Mean', 'StdDev', 'BestParameters'}]);

% Display the detailed results
disp('Hyperparameter Tuning Results:');
disp(detailedResultTable);

% Save to CSV
writetable(detailedResultTable, fullfile(baseDir,
'tuned_classifier_results.csv'));

% Create a boxplot to visualize the distribution of accuracies across
folds
figure('Position', [100, 100, 1200, 600], 'Name', 'Tuned Classifier
Performance Distribution');

% Prepare data for boxplot
boxplotData = allAccuracies';  % Transpose to get classifiers as
groups
boxplot(boxplotData, 'Labels', classifierNames);
title('Distribution of Accuracies Across CV Folds (Tuned Models)',
'FontSize', 14, 'FontWeight', 'bold');
ylabel('Accuracy', 'FontSize', 12);
grid on;
xtickangle(45);

% Add individual points as scatter plot
hold on;
for i = 1:size(allAccuracies, 1)
    % Plot the individual fold results as scattered points
    scatter(repmat(i, 1, numFolds), allAccuracies(i,:), 50,
'MarkerEdgeColor', [0 0.4 0.7], 'MarkerFaceColor', [0.3 0.6 0.9],
'LineWidth', 1.5);
end
hold off;

% Save the figure
saveas(gcf, fullfile(baseDir, 'tuned_classifier_distributions.png'));
saveas(gcf, fullfile(baseDir, 'tuned_classifier_distributions.fig'));

%% Create bar chart for easier comparison of mean accuracies
figure('Position', [100, 100, 1200, 600], 'Name', 'Tuned Classifier
Mean Performance');
% Sort the classifiers by mean accuracy
[sortedAccuracies, sortIdx] = sort(meanAccuracies, 'descend');
sortedClassifiers = classifierNames(sortIdx);
sortedStdAccuracies = stdAccuracies(sortIdx);
% Create bar chart
barHandle = bar(sortedAccuracies, 'FaceColor', [0.3 0.6 0.9]);
hold on;
% Add error bars
errorbar(1:length(sortedAccuracies), sortedAccuracies,
sortedStdAccuracies, 'k', 'LineStyle', 'none', 'LineWidth', 1.5);
% Customize the plot
set(gca, 'XTick', 1:length(sortedClassifiers), 'XTickLabel',
sortedClassifiers);
title('Mean Accuracy of Tuned Classifiers (Ranked)', 'FontSize', 14,
'FontWeight', 'bold');
```

```
ylabel('Mean Accuracy', 'FontSize', 12);
grid on;
xtickangle(45);
% Add value labels on top of bars
for i = 1:length(sortedAccuracies)
    text(i, sortedAccuracies(i) + 0.01, sprintf('%.4f',
sortedAccuracies(i)), ...
        'HorizontalAlignment', 'center', 'VerticalAlignment',
'bottom', 'FontWeight', 'bold');
end
% Save the figure
saveas(gcf, fullfile(baseDir, 'tuned_classifier_ranking.png'));
saveas(gcf, fullfile(baseDir, 'tuned_classifier_ranking.fig'));

%% Create heatmaps for selected parameter grids
disp('Generating parameter heatmaps...');

% Create a heatmap for Decision Tree parameters
figure('Position', [100, 100, 900, 700], 'Name', 'Decision Tree
Parameter Tuning');
dtGrid = tuningResults{1}.paramGrid;
[X, Y] = meshgrid(dtParams.MinLeafSize, dtParams.MaxNumSplits);
surf(X, Y, dtGrid);
title('Decision Tree Parameter Tuning', 'FontSize', 14, 'FontWeight',
'bold');
xlabel('Min Leaf Size', 'FontSize', 12);
ylabel('Max Num Splits', 'FontSize', 12);
zlabel('CV Accuracy', 'FontSize', 12);
colormap('jet');
colorbar;
view(45, 30);
grid on;
saveas(gcf, fullfile(baseDir, 'dt_parameter_tuning_3d.png'));

% Create a 2D heatmap version for easier viewing
figure('Position', [100, 100, 900, 700], 'Name', 'Decision Tree
Parameter Tuning (Heatmap)');
h = heatmap(dtParams.MinLeafSize, dtParams.MaxNumSplits, dtGrid);
h.Title = 'Decision Tree Parameter Tuning';
h.XLabel = 'Min Leaf Size';
h.YLabel = 'Max Num Splits';
colormap('jet');
saveas(gcf, fullfile(baseDir, 'dt_parameter_tuning_heatmap.png'));

% Create a heatmap for KNN parameters (using the first standardize
setting)
figure('Position', [100, 100, 900, 700], 'Name', 'KNN Parameter
Tuning');
knnGrid = squeeze(knnResults(:,:,1)); % First standardize setting
h = heatmap(knnParams.Distance, knnParams.NumNeighbors, knnGrid);
h.Title = 'KNN Parameter Tuning (Standardize=false)';
h.XLabel = 'Distance Metric';
h.YLabel = 'Number of Neighbors';
colormap('jet');
saveas(gcf, fullfile(baseDir, 'knn_parameter_tuning_heatmap.png'));

%% Create visualizations for Random Forest / Ensemble tuning

% Prepare to visualize Random Forest results
```

```
figure('Position', [100, 100, 1200, 700], 'Name', 'Random Forest
Parameter Tuning');

% Since RF results are 3D (method, cycles, learn rate), we need to
visualize carefully
% Create one subplot for each ensemble method
methods = rfParams.Method;
cycles = rfParams.NumLearningCycles;
rates = rfParams.LearnRate;

% Create a 2x2 subplot layout
for m = 1:length(methods)
    subplot(2, 2, m);
    
    % Extract data for this method
    methodData = squeeze(rfResults(m, :, :));
    
    % For 'Bag' method, we only have data for first learn rate, so
handle specially
    if strcmp(methods{m}, 'Bag')
        % For Bag method, just plot NumLearningCycles vs Accuracy
        plot(cycles, methodData(:,1), 'o-', 'LineWidth', 2,
'MarkerSize', 8, 'MarkerFaceColor', 'auto');
        title(['Method: ', methods{m}]);
        xlabel('Number of Learning Cycles');
        ylabel('Cross-Validation Accuracy');
        grid on;
    else
        % For boosting methods, create heatmap of cycles vs learn
rates
        imagesc(methodData);
        colormap('jet');
        colorbar;
        title(['Method: ', methods{m}]);
        xlabel('Learning Rate');
        ylabel('Number of Learning Cycles');
        
        % Set axis ticks
        xticks(1:length(rates));
        xticklabels(arrayfun(@(x) sprintf('%.1f', x), rates,
'UniformOutput', false));
        yticks(1:length(cycles));
        yticklabels(arrayfun(@(x) sprintf('%d', x), cycles,
'UniformOutput', false));
    end
end

% Add overall title and adjust layout
sgtitle('Random Forest / Ensemble Parameter Tuning', 'FontSize', 16,
'FontWeight', 'bold');
set(gcf, 'Color', 'white');
saveas(gcf, fullfile(baseDir, 'rf_parameter_tuning.png'));
saveas(gcf, fullfile(baseDir, 'rf_parameter_tuning.fig'));

%% Create bar chart for each ensemble method showing best performance

% Extract best accuracy for each method
methodBestAcc = zeros(1, length(methods));
for m = 1:length(methods)
```

```
    methodSlice = rfResults(m, :, :);
    methodBestAcc(m) = max(methodSlice(:));
end

% Create bar chart
figure('Position', [100, 100, 800, 600], 'Name', 'Ensemble Methods
Comparison');
bar(methodBestAcc, 'FaceColor', [0.2 0.6 0.8]);
grid on;
title('Best Accuracy by Ensemble Method', 'FontSize', 14,
'FontWeight', 'bold');
xlabel('Ensemble Method');
ylabel('Best Cross-Validation Accuracy');
xticks(1:length(methods));
xticklabels(methods);
xtickangle(45);

% Add data labels
for i = 1:length(methodBestAcc)
    text(i, methodBestAcc(i) + 0.01, sprintf('%.4f',
methodBestAcc(i)), ...
        'HorizontalAlignment', 'center', 'VerticalAlignment',
'bottom', 'FontWeight', 'bold');
end
saveas(gcf, fullfile(baseDir, 'ensemble_method_comparison.png'));

%% Visualize Naive Bayes results
figure('Position', [100, 100, 800, 600], 'Name', 'Naive Bayes
Parameter Tuning');

% Convert complex distribution names to simpler labels for
visualization
distLabels = cell(size(nbParams.DistributionNames));
for i = 1:length(nbParams.DistributionNames)
    dist = nbParams.DistributionNames{i};
    if iscell(dist)
        if length(dist) == numPredictors
            if all(strcmp(dist, 'normal'))
                distLabels{i} = 'All Normal';
            elseif all(strcmp(dist, 'kernel'))
                distLabels{i} = 'All Kernel';
            else
                % Count distributions
                normalCount = sum(strcmp(dist, 'normal'));
                kernelCount = sum(strcmp(dist, 'kernel'));
                distLabels{i} = sprintf('Mixed: %d normal, %d
kernel', normalCount, kernelCount);
            end
        else
            distLabels{i} = 'Custom';
        end
    else
        distLabels{i} = dist;
    end
end

% Create bar chart of results
bar(nbResults);
grid on;
```

```
title('Naive Bayes Distribution Performance', 'FontSize', 14,
'FontWeight', 'bold');
xlabel('Distribution Type');
ylabel('Cross-Validation Accuracy');
xticks(1:length(distLabels));
xticklabels(distLabels);
xtickangle(45);

% Add data labels
for i = 1:length(nbResults)
    text(i, nbResults(i) + 0.01, sprintf('%.4f', nbResults(i)), ...
        'HorizontalAlignment', 'center', 'VerticalAlignment',
'bottom', 'FontWeight', 'bold');
end
saveas(gcf, fullfile(baseDir, 'nb_parameter_tuning.png'));

%% Visualize LDA results
figure('Position', [100, 100, 900, 700], 'Name', 'LDA Parameter
Tuning');

% Since we have issues with the heatmap function, let's create a
visualization
% that shows the grid of results without using heatmap

% First, replace -Inf with NaN for visualization
ldaMat = ldaResults;
ldaMat(ldaMat == -Inf) = NaN;

% Create a 2D visualization
imagesc(ldaMat);
colormap('jet');
colorbar;

% Set axis labels
title('LDA Parameter Tuning', 'FontSize', 14, 'FontWeight', 'bold');
xlabel('Gamma');
ylabel('Discriminant Type');

% Set axis ticks
xticks(1:length(ldaParams.Gamma));
xticklabels(arrayfun(@(x) sprintf('%.2f', x), ldaParams.Gamma,
'UniformOutput', false));
yticks(1:length(ldaParams.DiscrimType));
yticklabels(ldaParams.DiscrimType);

% Add grid and save
grid on;
saveas(gcf, fullfile(baseDir, 'lda_parameter_tuning_fixed.png'));

%% Create combination 3D surface plot for best ensemble method
figure('Position', [100, 100, 1000, 800], 'Name', 'Best Ensemble
Method 3D Performance');

% Find the index of the best method
[~, bestMethodIdx] = max(methodBestAcc);
bestMethod = methods{bestMethodIdx};

% If the best method is not 'Bag', create 3D surface
if ~strcmp(bestMethod, 'Bag')
```

```
    % Extract the data for the best method
    methodData = squeeze(rfResults(bestMethodIdx, :, :));

    % Create mesh grid for 3D surface
    [X, Y] = meshgrid(rates, cycles);

    % Create surface plot
    surf(X, Y, methodData);
    title(['3D Performance Surface for ', bestMethod], 'FontSize',
14, 'FontWeight', 'bold');
    xlabel('Learning Rate');
    ylabel('Number of Learning Cycles');
    zlabel('Cross-Validation Accuracy');
    colormap('jet');
    colorbar;
    view(45, 30);
else
    % For Bag method, create a different visualization
    bagData = squeeze(rfResults(1, :, 1)); % Only first column has
data

    % Plot the performance vs cycles
    plot(cycles, bagData, 'o-', 'LineWidth', 2, 'MarkerSize', 8,
'MarkerFaceColor', 'auto');
    title(['Performance for ', bestMethod, ' Method'], 'FontSize',
14, 'FontWeight', 'bold');
    xlabel('Number of Learning Cycles');
    ylabel('Cross-Validation Accuracy');
    grid on;
end

saveas(gcf, fullfile(baseDir, ['best_ensemble_', bestMethod,
'_performance.png']));

%% Create ROC curves for the best models
%% Create ROC curves for the best models
disp('Generating ROC curves for best models...');

% Create a new figure for ROC curves
figure('Position', [100, 100, 1200, 800], 'Name', 'ROC Curves for
Best Models');

% Define color map for different classifiers
colorMap = jet(length(classifierNames));
lineStyles = {'-', '--', ':', '-.', '-', '--', ':'};

% AUC values to store
aucValues = zeros(length(classifierNames), 1);

% Debug information
fprintf('Number of observations: %d\n', length(Y));
fprintf('Number of folds: %d\n', numFolds);

% Check if finalCVPartition is valid
if ~exist('finalCVPartition', 'var') || isempty(finalCVPartition)
    disp('Warning: finalCVPartition is not valid, creating a new
one...');
    finalCVPartition = cvpartition(Y, 'KFold', numFolds);
end
```

```
% We'll use the last fold for ROC curve demonstration (with safety
check)
testFold = min(numFolds, finalCVPartition.NumTestSets);
fprintf('Using test fold %d of %d\n', testFold,
finalCVPartition.NumTestSets);

% Get training and test indices safely
try
    trainIdx = finalCVPartition.training(testFold);
    testIdx = finalCVPartition.test(testFold);

    % Verify indices
    fprintf('Number of training samples: %d\n', sum(trainIdx));
    fprintf('Number of test samples: %d\n', sum(testIdx));

    if max(find(trainIdx)) > size(X, 1) || max(find(testIdx)) >
size(X, 1)
        error('Indices out of bounds for data matrix');
    end
catch ME
    disp('Error getting partition indices, using simple split
instead');
    disp(ME.message);

    % Create a simple 80/20 split as fallback
    n = length(Y);
    trainFraction = 0.8;
    shuffledIndices = randperm(n);
    trainSize = floor(trainFraction * n);

    trainIdx = false(n, 1);
    trainIdx(shuffledIndices(1:trainSize)) = true;
    testIdx = ~trainIdx;

    fprintf('Fallback: %d training samples, %d test samples\n',
sum(trainIdx), sum(testIdx));
end

% Plot ROC for each classifier
hold on;
legends = {};

for i = 1:length(classifierNames)
    try
        % Extract training and testing data
        Xtrain = X(trainIdx,:);
        Ytrain = Y(trainIdx);
        Xtest = X(testIdx,:);
        Ytest = Y(testIdx);

        fprintf('Training classifier %d: %s\n', i,
classifierNames{i});

        % Train the model with best parameters
        switch i
            case 1 % Decision Tree
                fprintf('  Training Decision Tree with MaxSplits=%d,
MinLeaf=%d\n', bestDTMaxSplits, bestDTMinLeaf);
```

```
                model = fitctree(Xtrain, Ytrain, 'MaxNumSplits',
bestDTMaxSplits, 'MinLeafSize', bestDTMinLeaf);
            case 2 % KNN
                fprintf('  Training KNN with k=%d, Distance=%s,
Standardize=%d\n', bestKnnNeighbors, bestKnnDistance,
bestKnnStandardize);
                model = fitcknn(Xtrain, Ytrain, 'NumNeighbors',
bestKnnNeighbors, 'Distance', bestKnnDistance, 'Standardize',
bestKnnStandardize);
            case 3 % SVM
                fprintf('  Training SVM with Kernel=%s, C=%f,
Scale=%f\n', bestSVMKernel, bestSVMBoxConstraint,
bestSVMKernelScale);
                model = fitcecoc(Xtrain, Ytrain, 'Learners',
templateSVM('KernelFunction', bestSVMKernel, 'BoxConstraint',
bestSVMBoxConstraint, 'KernelScale', bestSVMKernelScale), 'Coding',
bestSVMCoding);
            case 4 % Random Forest/Ensemble
                if strcmp(bestRFMethod, 'Bag')
                    fprintf('  Training %s with Cycles=%d\n',
bestRFMethod, bestRFCycles);
                    model = fitcensemble(Xtrain, Ytrain, 'Method',
bestRFMethod, 'NumLearningCycles', bestRFCycles);
                else
                    fprintf('  Training %s with Cycles=%d,
LearnRate=%f\n', bestRFMethod, bestRFCycles, bestRFLearnRate);
                    model = fitcensemble(Xtrain, Ytrain, 'Method',
bestRFMethod, 'NumLearningCycles', bestRFCycles, 'LearnRate',
bestRFLearnRate);
                end
            case 5 % Naive Bayes
                fprintf('  Training Naive Bayes\n');
                model = fitcnb(Xtrain, Ytrain, 'DistributionNames',
bestNBDist);
            case 6 % LDA
                fprintf('  Training LDA with Type=%s, Gamma=%f\n',
bestLDAType, bestLDAGamma);
                model = fitcdiscr(Xtrain, Ytrain, 'DiscrimType',
bestLDAType, 'Gamma', bestLDAGamma);
            case 7 % Logistic Regression
                fprintf('  Training Logistic Regression with
Lambda=%f, Coding=%s\n', bestLogitLambda, bestLogitCoding);
                template = templateLinear('Learner', 'logistic',
'Lambda', bestLogitLambda);
                model = fitcecoc(Xtrain, Ytrain, 'Learners',
template, 'Coding', bestLogitCoding);
        end

        % Get ROC curve data
        % For multiclass problems, we'll use one-vs-all approach
        uniqueClasses = unique(Ytrain);
        numClasses = length(uniqueClasses);

        fprintf('  Generating ROC curve (Classes=%d)\n', numClasses);

        % Convert to binary classification problem (first class vs
others)
        % For simplicity, we'll just use the first class vs rest
        yTestBinary = double(Ytest == uniqueClasses(1));
        % Get prediction scores
        try
            if isprop(model, 'ClassNames') &&
length(model.ClassNames) == 2
                % Binary classifier case
                [~, scores] = predict(model, Xtest);
                scores = scores(:,2); % Use scores for positive class
            else
                % Multiclass case
                [~, scores] = predict(model, Xtest);
                if size(scores, 2) >= numClasses
                    % Use scores for the first class
                    classIdx = find(model.ClassNames ==
uniqueClasses(1));
                    if ~isempty(classIdx)
                        scores = scores(:, classIdx);
                    else
                        scores = scores(:, 1);
                    end
                else
                    % Fall back to binary predictions
                    preds = predict(model, Xtest);
                    scores = double(preds == uniqueClasses(1));
                end
            end
        catch ME
            fprintf('  Error getting scores: %s\n', ME.message);
            % If we can't get scores, just use binary predictions
            preds = predict(model, Xtest);
            scores = double(preds == uniqueClasses(1));
        end

        % Calculate ROC curve
        [fpr, tpr, ~] = perfcurve(yTestBinary, scores, 1);
        auc = trapz(fpr, tpr); % Calculate AUC
        aucValues(i) = auc;

        % Plot ROC curve
        plot(fpr, tpr, 'Color', colorMap(i,:), 'LineWidth', 2,
'LineStyle', lineStyles{mod(i-1, length(lineStyles))+1});
        legends{end+1} = sprintf('%s (AUC=%.4f)', classifierNames{i},
auc);

        fprintf('  AUC for %s: %.4f\n', classifierNames{i}, auc);
    catch ME
        fprintf('Error processing classifier %d (%s): %s\n', i,
classifierNames{i}, ME.message);
        aucValues(i) = NaN;
        legends{end+1} = sprintf('%s (Error)', classifierNames{i});
    end
end

% Plot reference line
plot([0 1], [0 1], 'k--');
legends{end+1} = 'Random Guess';

% Customize the plot
xlabel('False Positive Rate', 'FontSize', 12);
```

```
ylabel('True Positive Rate', 'FontSize', 12);
title('ROC Curves for Tuned Classifiers (First Class vs Rest)',
'FontSize', 14, 'FontWeight', 'bold');
legend(legends, 'Location', 'southeast', 'FontSize', 10);
grid on;
axis square;

% Save the figure
saveas(gcf, fullfile(baseDir, 'roc_curves.png'));
saveas(gcf, fullfile(baseDir, 'roc_curves.fig'));

%% Create AUC comparison bar chart
figure('Position', [100, 100, 1000, 600], 'Name', 'AUC Comparison');

% Create bar chart of AUC values
bar(aucValues, 'FaceColor', [0.3 0.6 0.9]);
grid on;
title('Area Under ROC Curve (AUC) by Classifier', 'FontSize', 14,
'FontWeight', 'bold');
xlabel('Classifier');
ylabel('AUC');
xticks(1:length(classifierNames));
xticklabels(classifierNames);
xtickangle(45);

% Add data labels
for i = 1:length(aucValues)
    if ~isnan(aucValues(i))
        text(i, aucValues(i) + 0.01, sprintf('%.4f',
aucValues(i)), ...
            'HorizontalAlignment', 'center', 'VerticalAlignment',
'bottom', 'FontWeight', 'bold');
    end
end

saveas(gcf, fullfile(baseDir, 'auc_comparison.png'));
saveas(gcf, fullfile(baseDir, 'auc_comparison.fig'));


%% Create confusion matrices for best performing classifier

% Find the best classifier based on AUC
[bestAUC, bestClassifierIdx] = max(aucValues);
bestClassifierName = classifierNames{bestClassifierIdx};

fprintf('Creating confusion matrix for best classifier: %s
(AUC=%.4f)\n', bestClassifierName, bestAUC);

figure('Position', [100, 100, 800, 600], 'Name', ['Confusion Matrix -
', bestClassifierName]);

% Train the best model
Xtrain = X(trainIdx,:);
Ytrain = Y(trainIdx);
Xtest = X(testIdx,:);
Ytest = Y(testIdx);

% Train model with best parameters
switch bestClassifierIdx
```

```
    case 1 % Decision Tree
        model = fitctree(Xtrain, Ytrain, 'MaxNumSplits',
bestDTMaxSplits, 'MinLeafSize', bestDTMinLeaf);
    case 2 % KNN
        model = fitcknn(Xtrain, Ytrain, 'NumNeighbors',
bestKnnNeighbors, 'Distance', bestKnnDistance, 'Standardize',
bestKnnStandardize);
    case 3 % SVM
        model = fitcecoc(Xtrain, Ytrain, 'Learners',
templateSVM('KernelFunction', bestSVMKernel, 'BoxConstraint',
bestSVMBoxConstraint, 'KernelScale', bestSVMKernelScale), 'Coding',
bestSVMCoding);
    case 4 % Random Forest/Ensemble
        if strcmp(bestRFMethod, 'Bag')
            model = fitcensemble(Xtrain, Ytrain, 'Method',
bestRFMethod, 'NumLearningCycles', bestRFCycles);
        else
            model = fitcensemble(Xtrain, Ytrain, 'Method',
bestRFMethod, 'NumLearningCycles', bestRFCycles, 'LearnRate',
bestRFLearnRate);
        end
    case 5 % Naive Bayes
        model = fitcnb(Xtrain, Ytrain, 'DistributionNames',
bestNBDist);
    case 6 % LDA
        model = fitcdiscr(Xtrain, Ytrain, 'DiscrimType', bestLDAType,
'Gamma', bestLDAGamma);
    case 7 % Logistic Regression
        template = templateLinear('Learner', 'logistic', 'Lambda',
bestLogitLambda);
        model = fitcecoc(Xtrain, Ytrain, 'Learners', template,
'Coding', bestLogitCoding);
end

% Get predictions
predictions = predict(model, Xtest);

% Get unique classes and ensure they're in the correct order
uniqueClasses = unique([Ytrain; Ytest]);
numClasses = length(uniqueClasses);

% Create confusion matrix
cm = confusionmat(Ytest, predictions);

% In case not all classes appear in the test set, ensure the
confusion matrix has the right dimensions
if size(cm, 1) < numClasses
    tempCM = zeros(numClasses, numClasses);
    tempCM(1:size(cm,1), 1:size(cm,2)) = cm;
    cm = tempCM;
end

% Plot confusion matrix
h = heatmap(cm);
h.XDisplayLabels = cellstr(string(uniqueClasses));
h.YDisplayLabels = cellstr(string(uniqueClasses));
h.Title = sprintf('Confusion Matrix - %s', bestClassifierName);
h.XLabel = 'Predicted Class';
h.YLabel = 'True Class';
```

```
h.ColorbarVisible = 'on';

% Add text annotations for percentages
% Convert to accuracy on a per-class basis
rowSums = sum(cm, 2);
cm_percent = cm ./ rowSums * 100;

% Save the figure
saveas(gcf, fullfile(baseDir, 'best_classifier_confusion.png'));
saveas(gcf, fullfile(baseDir, 'best_classifier_confusion.fig'));


%% Save the tuning results for future reference
disp('Saving tuning results...');
save(fullfile(baseDir, 'hyperparameter_tuning_results.mat'), ...
    'tuningResults', 'bestParams', 'classifierNames',
'allAccuracies', 'X', 'Y');

%% Train the best overall model
disp('Training the best overall model...');

% Find the best classifier
[bestAccuracy, bestClassifierIdx] = max(meanAccuracies);
bestClassifierName = classifierNames{bestClassifierIdx};

% Train on all data
disp(['Best classifier: ' bestClassifierName ' with mean accuracy: '
num2str(bestAccuracy)]);
disp('Training final model on all data...');

% Train final model based on the best classifier
switch bestClassifierIdx
    case 1 % Decision Tree
        finalModel = fitctree(X, Y, ...
            'MaxNumSplits', bestDTMaxSplits, 'MinLeafSize',
bestDTMinLeaf);
    case 2 % KNN
        finalModel = fitcknn(X, Y, ...
            'NumNeighbors', bestKnnNeighbors, 'Distance',
bestKnnDistance, ...
            'Standardize', bestKnnStandardize);
    case 3 % SVM
        finalModel = fitcecoc(X, Y, ...
            'Learners', templateSVM('KernelFunction',
bestSVMKernel, ...
                                  'BoxConstraint',
bestSVMBoxConstraint, ...
                                  'KernelScale',
bestSVMKernelScale), ...
            'Coding', bestSVMCoding);
    case 4 % Random Forest/Ensemble
        if strcmp(bestRFMethod, 'Bag')
            finalModel = fitcensemble(X, Y, ...
                'Method', bestRFMethod, 'NumLearningCycles',
bestRFCycles);
        else
            finalModel = fitcensemble(X, Y, ...
                'Method', bestRFMethod, 'NumLearningCycles',
bestRFCycles, ...
```

```
                'LearnRate', bestRFLearnRate);
        end
    case 5 % Naive Bayes
        finalModel = fitcnb(X, Y, 'DistributionNames', bestNBDist);
    case 6 % LDA
        finalModel = fitcdiscr(X, Y, ...
            'DiscrimType', bestLDAType, 'Gamma', bestLDAGamma);
    case 7 % Logistic Regression
        template = templateLinear('Learner', 'logistic', 'Lambda', 
bestLogitLambda);
        finalModel = fitcecoc(X, Y, ...
            'Learners', template, 'Coding', bestLogitCoding);
end

% Save the final model
save(fullfile(baseDir, 'final_best_model.mat'), 'finalModel', 
'bestClassifierName', 'bestClassifierIdx', 'bestParams');

disp('Hyperparameter tuning complete. The best model has been 
saved.');

%% Plot feature importance for applicable models
if ismember(bestClassifierIdx, [1, 4]) % Decision Tree or Random 
Forest
    disp('Calculating feature importance...');

    figure('Position', [100, 100, 1000, 600], 'Name', 'Feature 
Importance');

    if bestClassifierIdx == 1 % Decision Tree
        % Get importance
        importance = finalModel.predictorImportance;
    else % Random Forest/Ensemble
        % Get importance
        importance = finalModel.predictorImportance;
    end

    % Sort features by importance
    [sortedImp, sortIdx] = sort(importance, 'descend');
    featureNames = rocFeatures(sortIdx);

    % Plot
    barh(sortedImp);
    yticks(1:length(featureNames));
    yticklabels(featureNames);
    xlabel('Predictor Importance', 'FontSize', 12);
    title(['Feature Importance for ' bestClassifierName], 'FontSize', 
14, 'FontWeight', 'bold');
    grid on;

    % Save the figure
    saveas(gcf, fullfile(baseDir, 'feature_importance.png'));
    saveas(gcf, fullfile(baseDir, 'feature_importance.fig'));
end

%% Final report
disp('Generating final report...');

% Create a summary figure with key results
```

```
figure('Position', [100, 100, 1200, 900], 'Name', 'Hyperparameter
Tuning Summary');

% Create a 2x2 subplot layout
subplot(2, 2, 1);
barHandle = bar(sortedAccuracies, 'FaceColor', [0.3 0.6 0.9]);
hold on;
errorbar(1:length(sortedAccuracies), sortedAccuracies,
sortedStdAccuracies, 'k', 'LineStyle', 'none', 'LineWidth', 1.5);
set(gca, 'XTick', 1:length(sortedClassifiers), 'XTickLabel',
sortedClassifiers);
title('Mean Accuracy of Tuned Classifiers', 'FontSize', 12);
ylabel('Mean Accuracy', 'FontSize', 10);
grid on;
xtickangle(45);
set(gca, 'FontSize', 8);

% Best model confusion matrix
subplot(2, 2, 2);
bestCM = allConfMats(:,:,bestClassifierIdx);
bestCMNorm = bestCM ./ sum(bestCM, 2);
imagesc(bestCMNorm);
colormap('jet');
colorbar;
title(['Confusion Matrix - ' bestClassifierName], 'FontSize', 12);
xlabel('Predicted Class', 'FontSize', 10);
ylabel('True Class', 'FontSize', 10);
set(gca, 'FontSize', 8);
axis square;

% ROC curves (simplified version)
subplot(2, 2, 3);
[~, topClassifiers] = sort(aucValues, 'descend');
top3Classifiers = topClassifiers(1:min(3, length(topClassifiers)));

hold on;
for i = 1:length(top3Classifiers)
    idx = top3Classifiers(i);

    % Train the model with best parameters for ROC
    switch idx
        case 1 % Decision Tree
            model = fitctree(X(trainIdx,:), Y(trainIdx), ...
                'MaxNumSplits', bestDTMaxSplits, 'MinLeafSize',
bestDTMinLeaf);
        case 2 % KNN
            model = fitcknn(X(trainIdx,:), Y(trainIdx), ...
                'NumNeighbors', bestKnnNeighbors, 'Distance',
bestKnnDistance, ...
                'Standardize', bestKnnStandardize);
        case 3 % SVM
            model = fitcecoc(X(trainIdx,:), Y(trainIdx), ...
                'Learners', templateSVM('KernelFunction',
bestSVMKernel, ...
                                              'BoxConstraint',
bestSVMBoxConstraint, ...
                                              'KernelScale',
bestSVMKernelScale), ...
                'Coding', bestSVMCoding);
        case 4 % Random Forest/Ensemble
            if strcmp(bestRFMethod, 'Bag')
                model = fitcensemble(X(trainIdx,:), Y(trainIdx), ...
                    'Method', bestRFMethod, 'NumLearningCycles',
bestRFCycles);
            else
                model = fitcensemble(X(trainIdx,:), Y(trainIdx), ...
                    'Method', bestRFMethod, 'NumLearningCycles',
bestRFCycles, ...
                    'LearnRate', bestRFLearnRate);
            end
        case 5 % Naive Bayes
            model = fitcnb(X(trainIdx,:), Y(trainIdx),
'DistributionNames', bestNBDist);
        case 6 % LDA
            model = fitcdiscr(X(trainIdx,:), Y(trainIdx), ...
                'DiscrimType', bestLDAType, 'Gamma', bestLDAGamma);
        case 7 % Logistic Regression
            template = templateLinear('Learner', 'logistic',
'Lambda', bestLogitLambda);
            model = fitcecoc(X(trainIdx,:), Y(trainIdx), ...
                'Learners', template, 'Coding', bestLogitCoding);
    end

    % Get ROC curve data (simplified)
    yTest = Y(testIdx);
    yTestBinary = double(yTest == uniqueClasses(1));

    % Get predictions
    preds = predict(model, X(testIdx,:));

    % Calculate ROC curve (simplified)
    [fpr, tpr, ~] = perfcurve(yTestBinary, double(preds ==
uniqueClasses(1)), 1);
    plot(fpr, tpr, 'LineWidth', 2, 'DisplayName',
classifierNames{idx});
end

% Plot reference line
plot([0 1], [0 1], 'k--', 'DisplayName', 'Random');
xlabel('False Positive Rate', 'FontSize', 10);
ylabel('True Positive Rate', 'FontSize', 10);
title('ROC Curves (Top 3 Models)', 'FontSize', 12);
legend('Location', 'southeast', 'FontSize', 8);
grid on;
axis square;
set(gca, 'FontSize', 8);

% Best model parameters
subplot(2, 2, 4);
text(0.5, 0.5, sprintf('Best Model: %s\n\nParameters:\n%s\n\nMean
Accuracy: %.4f ± %.4f', ...
    bestClassifierName, bestParams{bestClassifierIdx},
meanAccuracies(bestClassifierIdx),
stdAccuracies(bestClassifierIdx)), ...
    'HorizontalAlignment', 'center', 'FontSize', 10);
axis off;

% Save the summary figure
```

```
saveas(gcf, fullfile(baseDir, 'hyperparameter_tuning_summary.png'));
saveas(gcf, fullfile(baseDir, 'hyperparameter_tuning_summary.fig'));

%% Quality Percentage (QP) Calculation with Cross-Validation
disp('Calculating Quality Percentage (QP) metrics with cross-
validation...');

% Prepare data for cross-validation
X_cv = cleanFeatures{:, rocFeatures};
Y_cv = cleanFeatures.Class;

% Set up cross-validation
rng(1); % For reproducibility
cvFolds = 5;
cv = cvpartition(Y_cv, 'KFold', cvFolds);

% Initialize arrays to store results
cvAccuracy = zeros(cvFolds, 1);
cvPredictions = zeros(size(Y_cv));

% Perform cross-validation
for k = 1:cvFolds
    % Split data
    trainIdx = cv.training(k);
    testIdx = cv.test(k);

    % Train decision tree on training set
    cvTree = fitctree(X_cv(trainIdx,:), Y_cv(trainIdx),
'PredictorNames', rocFeatures, ...
                     'MaxNumSplits', 10, 'MinLeafSize', 5);

    % Predict on test set
    cvPredictions(testIdx) = predict(cvTree, X_cv(testIdx,:));

    % Calculate accuracy for this fold
    cvAccuracy(k) = sum(cvPredictions(testIdx) == Y_cv(testIdx)) /
sum(testIdx);

    disp(['  CV Fold ' num2str(k) ' Accuracy: '
num2str(cvAccuracy(k)*100, '%.2f') '%']);
end

% Store cross-validated predictions
cleanFeatures.CVPredictedClass = cvPredictions;

% Display overall cross-validation results
disp(['  Mean CV Accuracy: ' num2str(mean(cvAccuracy)*100, '%.2f') '%
(SD: ' num2str(std(cvAccuracy)*100, '%.2f') '%)']);

% Apply ROC-optimized threshold classification
cleanFeatures.PredictedClass_Thresholds =
zeros(height(cleanFeatures), 1);
for i = 1:height(cleanFeatures)
    row = cleanFeatures(i,:);
    cleanFeatures.PredictedClass_Thresholds(i) =
classifyStability(row, optimalThresholds);
end

 Apply decision tree classification - fix the X_tree reference
```

```
X_for_prediction = cleanFeatures{:, rocFeatures}; % Using the same
features for prediction
cleanFeatures.PredictedClass_DT = predict(dtModel, X_for_prediction);

% Calculate accuracies for each method
accuracyThresholds = sum(cleanFeatures.PredictedClass_Thresholds ==
cleanFeatures.Class) / height(cleanFeatures);
accuracyDT = sum(cleanFeatures.PredictedClass_DT ==
cleanFeatures.Class) / height(cleanFeatures);

disp(['ROC-optimized thresholds accuracy: '
num2str(accuracyThresholds*100, '%.2f') '%']);
disp(['Decision tree accuracy: ' num2str(accuracyDT*100, '%.2f')
'%']);
disp(['Cross-validation accuracy: ' num2str(mean(cvAccuracy)*100,
'%.2f') '%']);

% Choose best model based on accuracy and assign to PredictedClass
if mean(cvAccuracy) > max(accuracyThresholds, accuracyDT)
    cleanFeatures.PredictedClass = cleanFeatures.CVPredictedClass;
    disp('Using cross-validated model as final model (best
performance)');
elseif accuracyDT > accuracyThresholds
    cleanFeatures.PredictedClass = cleanFeatures.PredictedClass_DT;
    disp('Using decision tree as final model (best performance)');
else
    cleanFeatures.PredictedClass =
cleanFeatures.PredictedClass_Thresholds;
    disp('Using ROC-optimized thresholds as final model (best
performance)');
end

% For each subject and condition
subjects = unique(cleanFeatures.Subject);
conditions = unique(cleanFeatures.Condition);

% Pre-allocate qpResults for better performance
qpResults = table('Size', [length(subjects)*length(conditions),
4], ...
                 'VariableTypes', {'string', 'string', 'double',
'double'}, ...
                 'VariableNames', {'Subject', 'Condition', 'QP',
'CV_QP'});
rowIdx = 1;

for s = 1:length(subjects)
    subject = subjects(s);

    for c = 1:length(conditions)
        condition = conditions(c);

        % Get data for this subject and condition
        subCondIdx = (cleanFeatures.Subject == subject &
cleanFeatures.Condition == condition);

        if sum(subCondIdx) == 0
            continue;
        end
        subCondData = cleanFeatures(subCondIdx, :);

        % Calculate percentage of correctly classified instances
        % For final model
        correctClassification = (subCondData.PredictedClass ==
subCondData.Class);
        qp = 100 * mean(correctClassification);

        % For cross-validated model
        cvCorrectClassification = (subCondData.CVPredictedClass ==
subCondData.Class);
        cvQp = 100 * mean(cvCorrectClassification);

        % Store results
        qpResults.Subject(rowIdx) = subject;
        qpResults.Condition(rowIdx) = condition;
        qpResults.QP(rowIdx) = qp;
        qpResults.CV_QP(rowIdx) = cvQp;
        rowIdx = rowIdx + 1;

        disp(['  QP for Subject ' char(subject) ', Condition '
char(condition) ...
              ': ' num2str(qp, '%.2f') '% (CV: ' num2str(cvQp,
'%.2f') '%)']);
    end
end

% Trim any unused rows
qpResults = qpResults(1:rowIdx-1,:);

% Save QP results
writetable(qpResults, [path filesep 'quality_percentage.csv']);

% Create QP visualization
figure('Name', 'Quality Percentage by Subject');
uniqueSubjects = unique(qpResults.Subject);
meanQP = zeros(length(uniqueSubjects), 1);
meanCVQP = zeros(length(uniqueSubjects), 1);

for i = 1:length(uniqueSubjects)
    subjIdx = strcmp(qpResults.Subject, uniqueSubjects(i));
    meanQP(i) = mean(qpResults.QP(subjIdx));
    meanCVQP(i) = mean(qpResults.CV_QP(subjIdx));
end

bar([meanQP, meanCVQP]);
xticklabels(uniqueSubjects);
xtickangle(45);
legend('Original Model', 'Cross-Validated');
ylabel('Quality Percentage (%)');
title('Classification Quality by Subject');
grid on;


%% Create and validate classification model
disp('Creating and validating stability classification model...');

% Create a model based on ROC-optimized thresholds
```

```
classifyWithThresholds = @(data, thresholds) classifyStability(data,
thresholds);

% Apply classification to the clean dataset
cleanFeatures.PredictedClass_Thresholds =
zeros(height(cleanFeatures), 1);
for i = 1:height(cleanFeatures)
    row = cleanFeatures(i,:);
    cleanFeatures.PredictedClass_Thresholds(i) =
classifyStability(row, optimalThresholds);
end

% Apply decision tree classification
cleanFeatures.PredictedClass_DT = predict(dtModel, X_tree);

% Evaluate classification accuracy
accuracyThresholds = sum(cleanFeatures.PredictedClass_Thresholds ==
cleanFeatures.Class) / height(cleanFeatures);
accuracyDT = sum(cleanFeatures.PredictedClass_DT ==
cleanFeatures.Class) / height(cleanFeatures);

disp(['ROC-optimized thresholds accuracy: '
num2str(accuracyThresholds*100, '%.2f') '%']);
disp(['Decision tree accuracy: ' num2str(accuracyDT*100, '%.2f')
'%']);
disp(['Cross-validation accuracy: ' num2str(mean(cvAccuracy)*100,
'%.2f') '%']);

% Create confusion matrices

% Choose best model and save as final predicted class
if mean(cvAccuracy) > max(accuracyThresholds, accuracyDT)
    cleanFeatures.PredictedClass = cleanFeatures.CVPredictedClass;
    disp('Using cross-validated model as final model (best
performance)');
elseif accuracyDT > accuracyThresholds
    cleanFeatures.PredictedClass = cleanFeatures.PredictedClass_DT;
    disp('Using decision tree as final model (best performance)');
else
    cleanFeatures.PredictedClass =
cleanFeatures.PredictedClass_Thresholds;
    disp('Using ROC-optimized thresholds as final model (best
performance)');
end

% Save classification results
writetable(cleanFeatures, [path filesep 'classified_features.csv']);

% Export optimal thresholds to a CSV file
thresholdTable = struct2table(optimalThresholds);
writetable(thresholdTable, [path filesep 'optimal_thresholds.csv']);

% Export trained model
save([path filesep 'stability_model.mat'], 'dtModel',
'optimalThresholds', ...
     'featureCols', 'classNames', 'bestFeatures');

%% [section omitted – basic Summary of data and visualisation,
trivial]
```

```
% Boxplots and Data Summary

%% Classification function using ROC-optimized thresholds
function class = classifyStability(data, thresholds)
    % Default to unknown class (C=4 per Image 3b)
    class = 4;

    % Follow the exact hierarchy from Image 3(b)
    % Check if below threshold for stable class (identified class:
C=0)
    if data.D_RMS < thresholds.D_RMS(1)
        class = 0; % Stable
    % Check if above AP threshold (identified class: C=1)
    elseif data.AP_range > thresholds.AP_range(2) && data.ML_range <
thresholds.ML_range(2)
        class = 1; % Antero-Posterior
    % Check if below ML threshold (identified class: C=2)
    elseif data.ML_range > thresholds.ML_range(3) && data.AP_range <
thresholds.AP_range(3)
        class = 2; % Medio-Lateral
    % Check for unstable (identified class: C=3)
    elseif data.D_RMS > thresholds.D_RMS(4)
        class = 3; % Unstable
    end
    % Otherwise remains as unknown (class = 4)
end

% Helper function for cross-validation
function acc = crossValidateModel(model, X, Y, cv)
    numFolds = cv.NumTestSets;
    predictions = zeros(size(Y));

    for i = 1:numFolds
        trainIdx = cv.training(i);
        testIdx = cv.test(i);

        % Train model on training set
        trainedModel = model.fit(X(trainIdx,:), Y(trainIdx));

        % Predict on test set
        predictions(testIdx) = trainedModel.predict(X(testIdx,:));
    end

    % Calculate accuracy
    acc = sum(predictions == Y) / length(Y);
end
```