

**School of Electrical Engineering, Computing, and
Mathematical Sciences**

**A Framework for Gamification of Human Joint
Remote Rehabilitation, Incorporating Non-Invasive
Sensors**

Siavash Khaksar

0000-0002-1944-1418

**This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University**

October 2023

Declaration

Declaration Statement: To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Human Ethics: The research presented and reported in this thesis was conducted in accordance with the National Health and Medical Research Council National Statement on Ethical Conduct in Human Research (2007) – updated March 2014. The proposed research study received human research ethics approval from the Curtin University Human Research Ethics Committee. The proposal names and ethic approval numbers have been provided below.

- An investigation into the feasibility of using three-dimensional motion analysis and portable inertial motion units to measure active and passive upper limb range of motion in children (RDHS-11-16)
- Classification of resting in bed positions for adults in a hospital bed using non-invasive sensors and machine learning (HRE2022-0585)
- Gamification of human joint rehabilitation (HRE2022-0583)
- Monitoring the human health condition in hostile environments utilising non-invasive sensors (HRE2021-0047)
- A single case study of the feasibility of using immersive virtual reality to improve upper limb function following spinal cord injury (HRE2019-0006)
- Reciprocal - iWHOTrial: A Multicentre Randomized Controlled Trial of Rigid Wrist and Orthoses for Young Children with CP (HRE223/2015)

Signature:

Date: 14/08/2023

Acknowledgment of Country

We acknowledge that Curtin University works across hundreds of traditional lands and custodial groups in Australia, and with First Nations people around the globe. We wish to pay our deepest respects to their ancestors and members of their communities, past, present, and to their emerging leaders. Our passion and commitment to work with all Australians and peoples from across the world, including our First Nations peoples are at the core of the work we do, reflective of our institutions' values and commitment to our role as leaders in the Reconciliation space in Australia.

Abstract

Patients who have suffered soft tissue injuries or undergone surgery often experience reduced muscle strength, flexibility, and pain in the affected area, which can interfere with daily activities such as eating, showering, and working. Rehabilitation exercises are crucial in reducing symptoms and returning patients to normal activities. However, performing exercises at home can be challenging without the supervision of a clinician to ensure the safety, efficacy, and correctness. Furthermore, patients may lose motivation due to the repetitive nature of the exercises and lack of feedback outside clinical environments. This research presents a framework for human joint rehabilitation that enables clinicians to set engaging rehabilitation tasks for their patients. The developed framework utilises non-invasive sensors and machine learning algorithms to precisely measure and document joint movements by providing real time feedback on the progress of the exercises for patients and clinicians. The implementation of gamification will add an entertaining and interactive dimension to the rehabilitation process, helping to increase patient engagement, which is a vital component of long-term rehabilitation success that will reduce the risk of exercises being abandoned.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Professor Iain Murray AM and Professor Tele Tan AM for their constant support, valuable guidance, and patience during my PhD research. I am honoured to have had the opportunity to conduct this research under their mentorship which went beyond technical support. Their trust and constant encouragement were my main driving forces to overcome the different challenges of this multidisciplinary research. They helped me grow as a researcher, as an educator, and above all as a person. I will forever be in their debt and continue to be inspired by their passion for research that helps better people's lives.

I am deeply grateful for the support I have received from the discipline leads of Electrical Engineering at Curtin School of Electrical Engineering, Computing, and Mathematical Sciences (EECMS), namely Professor Sumedha Rajakaruna, Professor Ahmed Abu-Siada, and Professor David Davidson. I would also like to thank Associate Professor Nicola Armstrong, the Director of Graduate Research at EECMS for her support and feedback when finalising my thesis writing. I would also like to thank Professor Andrew Rohl and Professor David Davidson for their support as my thesis chairpersons. I also want to express my thanks to other academics within the school who have helped and supported me by providing feedback during internal presentations related this research, namely Associate Professor Cesar Ortega-Sanchez, Adjunct Professor Rod Love, Dr Nazanin Mohammadi, Dr Himanshu Agrawal, Dr Hanieh Bakhshayesh, Dr Yifei Ren, Dr Nimsiri Abhayasinghe, Dr Shiva Sharif, Dr Reza Ryan, Dr Nimalika Fernando Thudugala Mudalige, and Ms Bitu Borazjani. I also want to thank my colleagues at Curtin University's Innovation Central Perth (ICP) for their support and encouragement during my research namely, Mr Andrew Bell, Mr Nick Schols.

Due to the nature of my research, I had the pleasure of collaborating with researchers outside engineering in areas of health science such as physiotherapy, occupation therapy, and biomechanical science. I would like to take this opportunity to thank these collaborators for their support, namely: Professor Catherine Elliott, Professor Christine Imms, Professor Kylie Hill, Professor Lisa Tee, Professor Yogi Yogesan, Professor Warren Mansell, Associate Professor Amity Campbell, Dr Corrine

Walmsley, Dr Meg Harrold, Dr Susan Morris, Dr Welber Marinovic, Dr Dale Edwick, Dr Dave Parson, and Ms Eliza Becker. The support provided by these collaborators allowed me to navigate health science aspects of the project and helped me get an understanding of clinical practices for rehabilitation. Their help and support are greatly appreciated.

I also got the opportunity to collaborate with researchers in area of computer science and data science. I would like to thank these collaborators for their support and feedback throughout my research namely, Professor Wanquan Liu, Professor Ling Li, Dr David McMeekin, Dr Hanness Herrmann, Ms Huizhu Pan, and Ms Sana Sabah.

I would also like to thank the staff of Curtin University's HIVE (Hub for Immersive Visualisation and eResearch), namely Associate Professor Andrew Woods as the Manager and Mr Wesley Lamont as the Supervisor at HIVE. They both supported my research by providing guidance on visualisation aspects of my research and aiding in setting up the Unity game engine.

I would also like to take this opportunity to thank my engineering honour students who contributed to early prototyping and software development based on my designs. These students were Mr Yuqing Shi, Ms Stefanie Pieters, Mr Luke Checker, Mr Joshua Hyde, Mr James Backer, Ms Eve Cooper, Mr Harrison Booker, Mr Adil Khokhar, Ms Rebecca Viney, Mr Daniel Friedlander, Mr Austin Parry, and Mr Mark Nguyen.

I also want to acknowledge the admin team at Curtin School of Electrical Engineering, Computing and Mathematical Sciences for their support in arranging my internal research presentations, admin documentation, and room bookings related to my research. These staff were: Ms Michelle Cutinha, Ms Mahlet Habtemariam, Ms Tamara Andreson, Ms Tasneem Dawood, Ms Lois Paffett, and Ms Simret Habtemariam.

On a personal note, I want to give my warmest gratitude to my parents, Dr Hamid Khaksar and Dr Roya Mahshid Falahi for their constant love and support throughout my studies. They believed in me every step of the way and always encouraged me to aim for higher goals. They are both medical doctors and I grew up learning from them how to be compassionate when interacting with patients. This helped me greatly during

the clinical trials for this research. Their passion for helping people is what inspired me to choose a research topic that utilises technology to help people.

Finally, I would like to express my deepest appreciation to my beloved partner and collaborator Ms Bita Borazjani for her love, support, and patience. She always believed in me to push past the barriers I faced in my research and never left my side. I am forever grateful to have her support in my life as a partner and I greatly respect her as a research collaborator.

Dedications

I dedicate this thesis to the clinicians working with patients in area of rehabilitation and occupational therapy, in acknowledgement of their commitment to helping people improve their day-to-day lives. I continue to be in awe of the level of patience and compassion it takes for people to dedicate themselves to this area of work.

I would also like to dedicate this thesis to my parents Dr Hamid Khaksar and Dr Roya Mahshid, my brother Sepehr Khaksar as well as my beloved partner Ms Bitra Borazjani. My family loved and supported me through my studies and their constant encouragement help me tremendously in completing my research.

Copyright Statement

I have obtained permission from the copyright owners to use any third-party copyright material reproduced in the thesis (e.g. questionnaires, artwork, unpublished letters), or to use any of my own published work (e.g. journal articles) in which the copyright is held by another party (e.g. publisher, co-author).

Publications

1.1 JOURNAL PUBLICATIONS

Please note that author attributions have been provided in Appendix A.

- 1) Khaksar, S., H. Pan, B. Borazjani, I. Murray, H. Himanshu, W. Liu, C. Elliott, C. Imms, A. Campbell, and C. Walmsley. 2021. "Application of Inertial Measurement Units and Machine Learning Classification in Cerebral Palsy: Randomized Controlled Trial." *JMIR Rehabilitation and Assistive Technologies* 8 (4)
- 2) Sabah Al-azzawi, S., S. Khaksar, E. Khdhair Hadi, H. Himanshu, and I. Murray. 2021. "HeadUp: A Low-Cost Solution for Tracking Head Movement of Children with Cerebral Palsy Using IMU." *MDPI Sensors* 21 (23)
- 3) Khaksar, S., S. Pieters, B. Borazjani, J. Hyde, H. Booker, A. Khokhar, I. Murray, and A. Campbell. 2022. "Posture Monitoring and Correction Exercises for Workers in Hostile Environments Utilizing Non-Invasive Sensors: Algorithm Development and Validation." *MDPI Sensors* 22 (24,9618)
- 4) Khaksar, S. L. Checker, B. Borazjani, I. Murray, 2023 "Design and Evaluation of an Alternative Control for a Quad-Rotor Drone using Hand Gesture Recognition" *MDPI Sensors* 2023

1.2 INVITED CONFERENCE PAPERS

- 1) Khaksar, S, I. Murray, N. Abhayasinghe, B. Borazjani, N. Mohammadi, S. Sharif, N. Fernando. "Engineering Applications in Rehabilitation and Assistive Technology" SLIIT International Conference on Engineering and Technology in Sri Lanka Institute of Information Technology 10th of February 2022
- 2) Khaksar, S. "A framework for gamification of remote wrist rehabilitation exercises utilising non-invasive sensors" Indo-Asia Digital Health & Telemedicine Summit 22nd March 2023

Table of Contents

Declaration	i
Acknowledgment of Country	ii
Abstract	iii
Acknowledgements	iv
Dedications	vii
Copyright Statement	viii
Publications	ix
1.1 Journal Publications	ix
1.2 Invited Conference papers	ix
Table of Contents	x
List of Figures	xiii
List of Tables	xvii
List of Abbreviations	xviii
Chapter 1: Introduction	1
1.1 Background	1
1.2 Scope and Objectives	2
1.3 Thesis Outline	3
Chapter 2: Literature Review	5
2.1 Limitation and Scope of Literature Review	5
2.2 Summary of Selected Human Motion Capture Technologies	5
2.2.1 Human Motion Capture Technologies in Clinical Environments	5
2.2.2 Sensor-Based Motion Capture Technologies	6
2.2.3 Optical Motion Capture Technologies	9
2.3 Review of Key Elements in Analysis of Human Motion Capture	10
2.3.1 Use of Coordinate Systems in Human Motion Capture	10
2.3.2 Calculation of Smoothness Measurements in Human Motion Capture	13
2.3.3 Gesture Recognition Approaches	16
2.3.4 Data Acquisition Methods	17
2.3.5 Gesture Description Overview	20
2.3.6 Application of Machine Learning in Gesture Identification	21
2.3.7 Gesture Classification	25
2.4 Current Practices In Human Joint Rehabilitation Utilising Traditional Methods and Gamified Approaches	26
2.4.1 Current Clinical Approaches for Human Joint Rehabilitation	26
2.4.2 Review of Existing Methods for Gamification of Rehabilitation	31
2.4.3 Fundamentals of Game Design Principles for Rehabilitation	33
2.4.4 Review of Key Components for a Gamification Framework	34
2.5 Summary and Implications	39
Chapter 3: Experiment Design	40

3.1	Overview of the Methodology for Experimental Design	40
3.2	Defining Hardware Requirements for the Framework	40
3.2.1	Hardware Requirements and Sensor Selection.....	40
3.2.2	Filtering and Sensor Fusion Techniques	43
3.2.3	Real time Joint Angle Measurement Techniques Utilising the Sensor Data.....	44
3.2.4	Representation of Smoothness of Movement Utilising Sensor Data.....	47
3.3	Defining the Machine Learning Requirements for the Framework.....	51
3.3.1	Use of Optical Motion Capture Technologies for The Framework.....	51
3.3.2	Defining Simplifications and Governing Criteria of the HGR.....	52
3.3.3	Stage One: Selection of Gesture-Description Model	54
3.3.4	Stage Two: Selection of Data-Acquisition Method.....	56
3.3.5	Stage Three: Selection of Gesture-Identification Algorithm.....	56
3.3.6	Stage Four: Validation of Selected Gesture-Identification Algorithms	57
3.3.7	Stage Five: Selection of Gesture-Classification Algorithm and Finger Joint Measurements.....	63
3.3.8	Stage Six: Gesture Mapping and Tuning.....	65
3.4	Design and Implementation of the Human Computer Interface.....	65
3.4.1	Use of Unity Game Engine for Visualization.....	65
3.4.2	Connectivity of Sensor and ML Elements to Unity	66
3.4.3	Importing Avatars in Unity for Visualisation of Movement	70
3.4.4	Animation Set up in Unity for Joint Movements	72
3.4.5	Canvas Setup to Streamline Navigation in Unity.....	73
3.4.6	Setting Thresholds for Movement Exercises in Unity.....	74
3.4.7	Implementing Rehabilitation Exercises in Unity.....	77
3.4.8	Implementation of Rehabilitation Goals via Dashboard Design	81
3.4.9	Designing Scene Transitions for In Unity	83
3.4.10	Implementation of Audio Cues to Enhance Engagement.....	83
3.4.11	Overview of the Implemented Framework Interactivity	84
3.4.12	Overview of the Generated Movement Data Reports	86
3.5	Development of Games for The Framework	88
3.5.1	Design Ideology for the Game Designs.....	88
3.5.2	Design and Development of Mini Games in Unity	89
3.5.3	Design and Development of Larger Games in Unity	103
3.6	Summary of the Experiment Design Methods.....	113
Chapter 4: Use Case Examples.....		115
4.1	Selection Criteria for Use Cases.....	115
4.2	Use Case in Classification of Movement Associated with Cerebral Palsy Case 1	117
4.3	Use Case in Classification of Movement Associated with Cerebral Palsy Case 2	135
4.4	Use Case in Posture Monitoring and Correction Exercises for Workers in Hostile Environments.....	153
4.5	Use Case in Alternative Control Systems for Flying Drones as a Physical Gamified Exercise.....	176
4.6	Summary of Use case Examples and Their Implementation	211
Chapter 5: Analysis of Results and System Testing		212
5.1	Introduction of System Testing and Verification	212
5.2	Analysis and System Testing – Engineering Specifications.....	212
5.2.1	Evaluation of Sensors Against Vicon Video Capture System.....	212
5.2.2	Evaluation of Sensor Joint Measurements Against Goniometers	214

5.2.3	Evaluation of Sensor Smoothness Measurements	220
5.2.4	Evaluation of HGR Elements	226
5.3	Analysis and System Testing - Clinical Perspectives	228
5.3.1	Use of Focus Group for Qualitative Feedback	228
5.3.2	Research Questions Used in the Focus Group Discussion	229
5.3.3	Setting, Population, and Participants of the Focus Group	232
5.3.4	Summary of Responses to the Research Questions	234
5.4	Summary of The Analysis of Results and System Testing	239
5.4.1	Summary of the Evaluation of Engineering Aspects	239
5.4.2	Summary of the Evaluation of Clinical Aspects.....	240
Chapter 6:	Conclusions.....	242
Bibliography	245
Appendices	253
Appendix A:	Author Attributions.....	253
Contribution Statement for Paper 1	253
Contribution Statement for Paper 2	255
Contribution Statement for Paper 3	256
Contribution Statement for Paper 4	257
Contribution Statement for Invited Conference 1:	258
Contribution Statement for Invited Conference 2:	259
Appendix B:	Sample Data Recordings from Developed Games.....	260
Appendix C:	Examples Results of Linear LDLJ and Angular SPARC for Smoothness Measurements.....	268
Appendix D:	Full Question List for Focus Group Discussion	271

List of Figures

Figure 1: Euler angles using an aircraft [36].....	11
Figure 2: Example of gimbal Lock [37].....	12
Figure 3: Layout of the orientation frame [37]	12
Figure 4: HGR Data Acquisition Categories [43].....	18
Figure 5: Summary of Hand-gesture modelling methods [43]	21
Figure 6: Active Wrist Exercises Examples (Provided by Royal Perth Hospital in Western Australia)	29
Figure 7: Wrist Strengthening Exercises Examples (Provided by Royal Perth Hospital in Western Australia).....	30
Figure 8: Process Diagram for Hardware Requirements [16].....	41
Figure 9: Comparison Graph for IMU selection [16]	42
Figure 10: Method 1 for Joint Angle Measurement using IMUs [16].	45
Figure 11: Method 2 for Joint Angle Measurement using IMUs [16].	46
Figure 12: Method 3 for Joint Angle Measurement using IMUs [16].	46
Figure 13: Main Code for Implementing SPARC.....	48
Figure 14: Main Code for Implementing DLJ	49
Figure 15: Main Code for Implementing LDLJ	50
Figure 16: The Main Code for Implementing LDLJ using IMU Data.....	51
Figure 17: Overview of 6 stages for implementing the HGR [43]	52
Figure 18: Example of sign language used for the HGR [43].....	55
Figure 19: Python code for calculating three-dimensional joint angles [43].	59
Figure 20: Python code for calculating two-dimensional joint angles [43]	59
Figure 21: High-level overview of the data stream elements of the framework.....	66
Figure 22: Sample code for streaming sensor data to a windows-based computer	67
Figure 23: Python code for implementing MediaPipe Hand for the data stream.....	67
Figure 24: Localisation of 21 hand landmarks within MediaPipe [100]	68
Figure 25: get_label function for processing HGR	68
Figure 26: GUI developed for scanning IMUs and transfer to Unity	69
Figure 27: Code for capturing and storing the movement data to Unity.	70
Figure 28: Base of Avatar used for replicating user movements.	71
Figure 29: Mapping capabilities of the implemented avatar.....	71
Figure 30: T-pose view of the avatar with Hierarchy panel.....	72

Figure 31: Settings priorities and Boolean triggers for camera transitions	73
Figure 32: List of designed canvases	73
Figure 33: Slider menu for setting thresholds	74
Figure 34: Script for saving the set threshold values obtained the sliders.	75
Figure 35: Script to store values obtained from the sliders.....	75
Figure 36: The LoadValues() script used for reading values.	76
Figure 37: LoadCsv() function for loading pre-defined exercise thresholds	77
Figure 38: Wrist movement and finger movement exercises used as example implementation.....	78
Figure 39: Script for setting thresholds limits in exercises	79
Figure 40: Script for thumb touching exercise.....	80
Figure 41: Script for calibration finger movement exercises	81
Figure 42: Scrip for checking the streak	82
Figure 43: Badge system for rewarding the pare after keeping streaks	82
Figure 44: Script for loading different scenes.....	83
Figure 45: Special class created for the audio elements.....	84
Figure 46: Audio manager used for adjusting different audio elements.....	84
Figure 47: Main page view of the Unity application	85
Figure 48: Example side view for wrist and hand movement exercises	85
Figure 49: Example of a selected hand movement exercises (Flexion/Extension).....	86
Figure 50: Example of a selected finger movement exercises (Thumb Touch).....	86
Figure 51: Example of collected data after engaging with a rehab exercise through Unity.	87
Figure 52: Example of collecting data after engaging with a mini game.	87
Figure 53: IMU placement for wrist rehabilitation exercises	89
Figure 54: Script for modifying the zombie's speed	90
Figure 55: Scrip that allows the camera to follow the characters.	90
Figure 56: Screenshot of the user view in the Monster Chase game	91
Figure 57: Screenshot of the Flap and Avoid Game	91
Figure 58: Script for mapping IMU data to the player movement.....	92
Figure 59: Script for implementing the random pipe height element.	92
Figure 60: Script for interaction between the pipe and the bird.....	93
Figure 61: Script for collision detection between the bird and the pipe	93
Figure 62: (a) Menu Screen; (b) Patient out of threshold region UFO movement flatlined (c) Flexion Threshold reached and UFO moves Down; (d)Extension Threshold reached and UFO moves UP.	94

Figure 63: (a) Menu Screen; (b) Settings (c) Level 0 game environment; (d) Level 1 game Environment with Panel moving left.	95
Figure 64: Scrip for managing the small ball.....	96
Figure 65: Script for managing the collision for the Brick-Busting game.....	97
Figure 66: Screenshot of the Dodge the spike game.....	97
Figure 67: Tilting Maze Game: (a) Initiation for Right hand (b) Pronation tilts maze right (c) Supination rotates Maze left (d) Level complete	98
Figure 68: Feedback panel for the Tilting Maze Game	98
Figure 69: The Glasses Wiping game.	99
Figure 70: Piano Tiles Mini Game.....	100
Figure 71: Script for creating black tiles for the Piano Tiles game.	101
Figure 72: Part 1 of the script for implementing the tiles game mechanism.	102
Figure 73: Part 2 of the script for implementing the tiles game mechanism.	102
Figure 74: Threshold mode for setting baselines for the Skiing game	103
Figure 75: Main view mode of the Skiing game.....	104
Figure 76: Character animation for the Skiing game	104
Figure 77: Gameplay mode 2 of the Skiing game encouraging to hold a stretch. ...	105
Figure 78: Feedback provided to the user while engaging in gameplay mode 2.....	105
Figure 79: Threshold setting for the Aeroplane flying game.....	106
Figure 80: Main feedback window for the Aeroplane flying game	106
Figure 81: Aeroplane flying game in Pitch mode	107
Figure 82: Aeroplane flying game in Yaw mode.....	107
Figure 83: Aeroplane flying game in Roll mode.	107
Figure 84: Script used for the main flying control of Aeroplane Flying game.....	108
Figure 85: Script for testing the IMU input for Aeroplane flying game.....	109
Figure 86: Implementation of the ControlPlane() function within IMUController.cs	110
Figure 87: The script for saveThresholds() function.....	111
Figure 88: Script for setting starting positions for the Aeroplane Flying game.....	111
Figure 89: Implementing the scoring system for the Aeroplane Flying Game.....	111
Figure 90: The GameManagerScript for Aeroplane flying game	112
Figure 91: Game over screen implementation for the Aeroplane flying game.....	112
Figure 92: Placing reflectors on a user for Vicon Validation [16].....	213
Figure 93: IMU vs Vicon for flexion and extension exercise	214
Figure 94: IMU vs Vicon for object pick up and stop sign exercises.	214
Figure 95: IMUs connected directly to a goniometer	215

Figure 96: Comparison of Different IMUs with a goniometer	216
Figure 97: Use of goniometer for calculating wrist joints (Baseline measurement)	217
Figure 98: Use of goniometer for calculating wrist joints (Extension measurement)	217
Figure 99: Use of goniometer for calculating wrist joints (Flexion measurement)	218
Figure 100: Linear SPARC for User A	225
Figure 101: Linear SPARC for User B	226
Figure 102: Linear SPARC for User C	226
Figure 103: Sample extension/flexion data for the Monster Chase game.....	260
Figure 104: Sample radial/ulnar deviation data for the Monster Chase game	260
Figure 105: Sample pronation/supination data for the Monster Chase game	261
Figure 106: Sample flexion/extension data for the Flap and Avoid game.....	261
Figure 107: Sample flexion/extension data for the Hovercraft game	262
Figure 108: Sample flexion/extension data for the Hovercraft game with larger obstacles	262
Figure 109: Sample radial/ulnar deviation data for the Brick-Busting game	263
Figure 110: More radial/ulnar deviation data samples for the Brick-Busting game; (a) Small Panel, (b) Medium Panel,	263
Figure 111: Sample pronation/supination data for the Dodge the Spike game.....	264
Figure 112: Sample pronation/supination data for the Tilting Maze game.....	264
Figure 113: Sample pronation/supination data for the Skiing game	265
Figure 114: Sample flexion/extension data for the Skiing game	265
Figure 115: Sample data from Aeroplane flying game representing movement in pitch angles.	266
Figure 116: Sample data from Aeroplane flying game representing movement in Yaw angles.....	266
Figure 117: Sample data from Aeroplane flying game representing movement in Roll angles	267
Figure 118: Linear LDLJ for User A	268
Figure 119: Linear LDLJ for User B.....	268
Figure 120: Linear LDLJ for User C.....	269
Figure 121: Angular SPARC for User A	269
Figure 122: Angular SPARC for User B.....	270
Figure 123: Angular SPARC for User C.....	270

List of Tables

Table 1: Comparing off the shelf IMUs [16]	42
Table 2: Specification of Xsens Dot [92].....	43
Table 3: Viewpoint and poser examples for stage four for HGR [43].....	60
Table 4: Dataset used for classification and analysis [43]	60
Table 5: Table to determining panel size based on angle thresholds.....	96
Table 6 : Joint angle measurement against goniometer for wrist flexion	218
Table 7: Joint angle measurement against goniometer for wrist extension	218
Table 8: Error calculation for flexion angles	219
Table 9: Error calculation for extension angles	219
Table 10: Error calculation for radial deviation angles.....	219
Table 11: Error calculation for ulnar deviation angles.....	219
Table 12: Smoothness measurement comparing normal vs jerky movements.	220
Table 13: Information regarding participants of the trial.....	221
Table 14: Smoothness measurement comparing healthy user vs user with minor spinal cord injury. (User A)	222
Table 15: Smoothness measurement comparing healthy user vs user with major spinal cord injury (User B trial 1)	223
Table 16: Smoothness measurement comparing healthy user vs user with major spinal cord injury (User B trial 2)	224
Table 17: Goniometer measurements for open palm [43]	227
Table 18: MPH measurements for open palm (front view) [43].....	227
Table 19: Accuracy of measurements from 3D data by finger [43]	228
Table 20: Accuracy of measurements from 3D data by viewpoint [43].....	228
Table 21: Information regarding discussion group participants	233
Table 22: Author Signatures for Paper 1.....	254
Table 23: Author Signatures for Paper 2.....	255
Table 24: Author Signatures for Paper 3.....	256
Table 25: Author Signatures for Paper 4.....	257
Table 26: Author Signatures for Invited Conference 1	258
Table 27: Author Signatures for Invited Conference 2.....	259

List of Abbreviations

- **AROM:** Active Range of Motion
- **AUC:** Area Under the Curve
- **CP:** Cerebral Palsy
- **HCA:** Head Control Ability
- **HCI:** Human Computer Interface
- **HGR:** Hand Gesture Recognition
- **I²C:** Inter-Integrated Circuit
- **IMU:** Inertial Measurement Unit
- **IR:** infrared
- **iWHOT:** Infant Wrist Hand Orthosis Trial
- **LDLJ:** Log Dimensionless Jerk
- **MIT:** Minimising Impairment Trial
- **ML:** Machine Learning
- **MPH:** MediaPipe Hands
- **PPE:** Personal Protective Equipment
- **PROM:** Passive Range of Motion
- **RCT:** Randomized Controlled Trial
- **RGB:** Red Green Blue
- **RF:** Radio Frequency
- **ROC:** receiver operating characteristic
- **ROM:** Range of Motion
- **SPARC:** Spectral Arc Length
- **SPI:** Serial Peripheral Interface
- **WEKA:** Waikato Environment for Knowledge Analysis

Chapter 1: Introduction

1.1 BACKGROUND

Within rehabilitation programs, patients are required to engage in a sequence of repetitive exercises as part of a timely intervention strategy. This approach aids them in recovering, enhancing, or maintaining their usual functional capabilities [1]. Patients in rehabilitation performing repetitive movements often find them tedious and increasingly frustrating. Ultimately, making the rehabilitation less effective as the patient becomes more un-motivated and discouraged throughout their exercise program [2]. Increasingly, home-based rehabilitation therapy for patients focused on performance through a goal-orientated task to maximise results and minimising long-term disability is becoming more relevant in today's circumstances [3]. Due to increasing demands for rehabilitation services, medical staff reduction, and rising costs, clinicians have limited availability to monitor and provide high levels of care to patients, resulting in longer recovery times [4]. Available clinical tools such as goniometers are the most common method to measure the baseline limitations of the range of motion for joint angles [5]. However, clinical supervision is required for using goniometers. The problem here is that as a patient progresses through their exercise program, they will need to visit their therapist to validate and monitor their progress. With this method, feedback to patients is limited by the therapist's availability rather than meeting the needs of the patient's requirements. The lack of supervision and limited feedback may lead to a reduction in patient motivation for long conditioning programs and has a negative impact of the effectiveness of the rehabilitation exercises. However, the introduction of non-invasive sensors and movement tracking devices can provide a new method to capture home-based rehabilitation exercises performed by the patient in real-time. Devices such as Microsoft Kinect and Leap Motion controller utilise infrared sensors and structured light technology, whereas video game controllers contain Inertial Measurement units (IMU) that can assist in extrapolating data for clinicians. This allows for opportunities to analyse patient movement data, while incorporating an environment for video games in which the patient can perform exercises movements in an engaging and enjoyable way [6]. Rehabilitation exercise

programs already utilise physical games and activities throughout an exercise program so use of technology in the space is a worthwhile effort[7].

1.2 SCOPE AND OBJECTIVES

This thesis provides a framework for gamification of human joint remote rehabilitation exercises utilising non-invasive sensors. This framework provides guidelines on how a given rehabilitation program can be enhanced with inclusion of gamification and how it can be adapted such that the data provided by the framework is clinically reliable. The main aim of this framework is to provide an environment where both clinicians and patients undertaking a rehabilitation exercise program can benefit from the inclusion of technology and gamified concepts as a means of generating greater motivation and engagement with the program. To Achieve this, non-invasive sensors and machine learning algorithms were utilised to capture human joint movement. This movement data facilitated engagement with the designed gamified rehabilitation exercises where information regarding the joint movements and correctness of the exercises is documented.

The developed framework provides live feedback to the user on how accurately they are engaging in the rehabilitation exercises and keeps a record of number of repetitions, range of joint movements, smoothness of movements, and correctness of exercises for each session. Clinicians can utilise this data as a metric to gauge the user's progress and engagement with the rehabilitation program, The data can also provide insight into why an exercise program is not working since movement over time is measured and logged through the framework.

Different elements of the developed framework were validated through clinical trials and pilot studies to ensure its clinical usability. Additionally, several use cases were considered and published in peer review journals to demonstrate the utility of the framework in different areas. It is important to highlight that due to inclusion of human participants in this research, human ethics approvals were attained, and participants provided informed consent when engaging with this research. Since the research included multiple use cases, where relevant, separate ethics approvals were attained.

The gamified elements of the framework will promote further engagement with the rehabilitation program and encourages the user to maintain participation in the required exercises. Numerous exercise templates and adaptable mini games have been

supplied as examples of aligning rehabilitation exercises with gamified encounters, all while preserving the integrity of the exercises. Additionally, guidelines have been provided on customising the gamified elements, so they meet rehabilitation requirements on a user-to-user basis.

1.3 THESIS OUTLINE

This thesis is presented into six main chapters that explain key aspects of this research. The thesis structure and a small summary of the content for each chapter has been explained below:

- **Chapter 2, Literature Review:**

This chapter provides an in-depth analysis of the literature by analysing different methods of capturing human movements that is relevant to rehabilitation. Both clinical and on clinical environments are reviewed as part of this chapter. After reviewing the technology, key elements in analysis of human movement technologies are reviewed. Finally, this chapter provides a review on current practices in human movement rehabilitation both using traditional methods and gamified approaches.

- **Chapter 3, Experiment Design:**

This chapter provides an overview on the methodology for design of different technologies required for defining the framework. In this chapter, hardware and machine learning requirements for development of the framework have been detailed followed by design of human computer interface components of the framework through visualisation in the Unity game engine. This chapter also highlights how joint and smoothness measurements are conducted. Finally, this chapter provides design templates for developing gamified exercises in such a way that the games are replicating gamification exercises and the data collected during the exercises is clinically valid and is relevant to the user and the clinician.

- **Chapter 4, Use Case Examples:**

This chapter contains four different use cases as published peer assessed journal articles that evaluate different elements related to the development of the framework. The evaluation techniques of both sensor and machine

learning elements of the paper can also be read in this chapter. All the published papers in this chapter are available in open access journals so the reader may choose to read the article on the journals website if they wish.

- **Chapter 5, Analysis of Results and System Testing:**

This chapter provides an in-depth details on analysis of results and system testing for this research. The evaluation of results is provided from both engineering perspectives as well as clinical perspectives. The engineering perspectives include result of different quantitative analysis methods for evaluating the data provided by the framework and details the details of relevant verification procedures. The second part of this chapter contains the qualitative analysis of a focus group discussion held with experts in field of rehabilitation.

- **Chapter 6, Conclusions:**

This chapter contains a summary of contributions of this thesis and provides guidelines for future development and directions for implementation of gamified concepts in rehabilitation settings.

Chapter 2: Literature Review

2.1 LIMITATION AND SCOPE OF LITERATURE REVIEW

This section provides an in-depth literature review on the topic of gamification of rehabilitation. The review will cover a summary of selected human motion capture technologies use in the context of rehabilitation. The technologies are categorized primarily into sensor-based and optical motion capture technologies, serving as the primary approaches for capturing human movement intended for integration into a gamified environment. As mentioned in Chapter 1, one of the objectives of the developed framework is to provide a low-cost solution that would be viable for most people so the most accurate solution may not be the best solution as the mechanism for capturing human movement. The requirement of low cost and accessibility of the solution exclude the study or technologies such as Virtual Reality (VR) or Augment Reality (AR).

After reviewing the technology, key elements in analysis of the human movement data are explored. This section will also provide an in depth look at one of the more complex aspects of human motion capture which is gesture control.

Finally, the review will focus on current practices in human joint rehabilitation both in clinical settings and gamified settings. This section will also provide a look at game design theories and how it has been applied to rehabilitation in the past.

2.2 SUMMARY OF SELECTED HUMAN MOTION CAPTURE TECHNOLOGIES

2.2.1 Human Motion Capture Technologies in Clinical Environments

Goniometers and inclinometers are the main tools used in clinical research to measure joint angles [8]. A goniometer is an instrument that can be used for measuring joint angles and the available range of motion at a joint and monitoring changes in joint angles in clinical settings, [9]. Inclinometers are specific types of goniometers dependent on gravity and are used to measure motion in the spine by placing the device on the neck or spine and reading the angle at different positions. This use of goniometers requires precision for an accurate reading which is achieved through skilful observation and practice, with human error being a major factor that leads to

them being inaccurate and unreliable, according to [10] and [11]. The accuracy of range of motion measurement is a crucial part of clinical assessment since this information is used as a guide for treatment plans, evaluating the effectiveness of treatments, and monitoring an individual's response to the treatments [12]. The reliance on the ability of the clinicians to accurately palpate bony landmarks and visually estimate the alignment of the axis of the body part and goniometer can cause issues. Goniometers are reliable, versatile, and widely used, even with measurement errors of up to 15 degrees. The issue lies when the dynamic range of movement needs to be measured, especially when dealing with a younger age group [13] [14].

In some instances, other clinical tools need to be utilised when there is a requirement to get a better understanding of overall posture rather than joint measurements. X-rays are used to determine spinal health and to evaluate the alignment of bones, and magnetic resonance imaging (MRI) or CT scans are used to determine any issues with nerves, muscles, tissues, tendons, bones, ligaments, etc [15]. Furthermore, nerve studies utilising electromyography (EMG) are used to measure electrical impulses produced by the nerve in response to muscles. This is done with the aim of determining nerve compressions caused by a herniated disk or diagnosing spinal stenosis [15].

2.2.2 Sensor-Based Motion Capture Technologies

There are several sensor-based approaches available such as use of Infrared (IR) LEDs, Fiber-Optic Sensors, E-Textile Sensors, and Inertial Measurement Units (IMU). These technologies are explored in the following subchapters [14] [16].

Infrared and Near Infrared Based Hand Tracking Technologies

Infrared cameras (IR) than detect light emitted from a surface within set temperature ranges can be utilised for capturing hand movements [17]. IR cameras allow segmentation techniques to be run on items within a similar temperature to the human body. This reduction in areas required for segmentation, the computational requirements are greatly reduced compared to methods that use colour-based hand tracking techniques [17] [14].

A study conducted in [17] demonstrates the ease of implementing segmentation in body temperature range of 30°C-37°C where the regions in this range provide a higher pixel value compared to the remaining pixels. The algorithm removes the other

objects detected within this range by omitting the smallest regions first followed by selecting 2 of the largest regions. The algorithm assumes a single arm if only one region has been detected [17]. The operator's hand and arms are extracted from an image followed by a search window to determine the orientation of each arm. The search window looks for fingertips in smaller windows which lowers the computational requirements. Finger approximation of a rectangle with a semicircle tip is then utilised for each finger. This method utilises the approximation mentioned before to search for semicircles in the segmented image. The study in [17] shows that 20 candidates with the highest ratings are utilised before removal of false positives from the sample. False positives on multiple matches around a candidate are removed, which means the neighbours with lower scores around the highest score candidate will be removed. Another approach removes false candidates by reviewing pixels surrounding the centre of a matching template. Multiple pixels are checked in a diagonal direction inside the hand region and the candidate is removed if they are found. After removal of the false positives, the system is able to correctly identify the fingertips in the captured image [17].

One of the uses of IR techniques can be seen in Leap Motion Controller (LMC) which is a small optical hand tracking module. LMC can track hands within 3D areas up to 110cm from the device in a field view of 160x160° [18]. LMC uses two near infrared cameras with 115 frames per second. Early testing results of LMC can be seen in [19] where static and dynamic hand tracking can be measured. However, the joint angles calculated via this device have not been clinically validated.

Active Sonar Finger Tracking

Sonar methods are commonly used to detect and determine the distance and direction of objects underwater utilising acoustics. An interesting use of sonar can be seen in FingerIO a project taking place at Washington University. This project aims to use active sonar for finger tracking. The study conducted by [20] utilises mobile devices such as smart phones and smart watches as active sonar tracking beacons. The system provides inaudible soundwaves in 18 to 20 kHz sample rates coupled with speed of sound in the air which provides an error of 2.1-2.8cm in finger position. FingerIO uses Orthogonal Frequency Division Multiplexing (OFDM) to compute cyclic suffix of S samples and uses it to calculate sample error. The sample error is then utilised to correct incoming signals to achieve fine grain finger tracking with 8mm accuracy [20].

The system is currently under development and there is relatively high errors in finger position which makes it not ideal for clinical applications.

Fiber-Optic Sensing for Posture Monitoring

Fiber-optic sensors utilise measurements of light traveling through an optical fibre system which can be in form of light intensity, phase, and polarization [21]. The advantage of this technique is the immunity to electromagnetic interference, radio frequency interference, and lack of effects from corrosive environments [22]. Fiber-optic sensors can withstand high temperatures, provide a wide dynamic range, and contain large bandwidths [22]. Study conducted by Roehampton University [23] provide a dynamic method in measuring lumbar curvature via Fiber-optic sensors. This study, 8 Fiber-optic sensors are paired in series and attached to a ribbon of sprung steel in an elastic housing which allowed the ribbon to slide freely during spinal movement. The measurements were compared to optical motion capture technologies for validation with the conclusion that these sensors can be used for sagittal lumbar curvature measurements across time [23]. This method is not suitable in clinical setting due to the high cost of the sensor implementation and lack of clinical validation of the sensor data [16].

E-Textile Sensors for Posture Monitoring

E-textile sensors commonly refers to electronic textiles which are fabrics that incorporate electronics woven within them [24]. The sensors within the fabrics are interconnected which will lead into a less invasive design, tangle free, and cannot be snagged by other objects. A review conducted by [25] mentions the human skin and the clothes as the inspiration behind the invention of textile sensors. Similar to human skin's reaction to stimuli, electronic textiles will react to the environment through transfer of energy through material [16].

As a part of the study conducted by [26], a wireless wearable T-shirt for posture monitoring of rehabilitation exercises has been developed. This solution is made of stretchable fabric containing wireless sensors that operate as textile substrate. The inductive sensors were sewn in a zigzag pattern to front and back of the shirt leading to a lightweight design. The sensors measured deformation as the shirt lengthens and shortens in the sagittal plane of the body utilising inductive impedance measurements. The results were validated against optical motion capture technologies to show the reliability of the techniques. The limitation of this design is the notion that the subject

seating is minimal, and it will not interfere with parasitic capacitance. This limitation as well as relaxation of the tight stretchable shirt worn our due to use and washes makes this design unreliable in clinical settings.

Inertial Measurement Unit (IMU)

Inertial measurement units (IMU) are among the most popular methods for capturing movement and position of objects. IMUs include an accelerometer, gyroscope, and magnetometer usually connected to microcontroller module to transfer orientation information[16]. IMUs are well-developed, non-invasive, affordable with long battery life [25]. IMUs require minimal computational power and have been implemented in wide range of application as reviewed by [27]. In recent years there has been a number of IMU based motion capture research studies such as studies of gait modulation in patients with foot drop problems [28] and human activity recognition using thigh angle derived from single thigh mounted IMU data [29]. The use of IMUs for hand movement in free space is currently underdeveloped primarily due to the lack of a clear calibration reset point compared to gait analysis. Later in Chapter 4 of this thesis, use of IMUs in classification of movement associated with Cerebral Palsy will be discussed [16] [14].

Depending on the application of IMUs, a relevant signal-processing pipeline needs to be used so that the data can be sent to an external computer accurately without any loss. It is also important to calibrate the device to find an initial position for the sensor [30]. Kalman filter, complementary filter and sensor fusion techniques are common filtering approaches when working with IMUs. IMUs provide a full orientation frame and position for the wrist which can be used to for the gamification of the rehabilitation exercises. The issue with using IMUs in free space is the lack of initial starting position for initialization which needs to be determined as a part of this research. The use of IMUs within the designed framework will further be discussed in Chapter 3.

2.2.3 Optical Motion Capture Technologies

The use of digital technology is another appropriate method for human motion capture. Motion capture (also known as mo-cap) refers to a group of technology where the movement of people is recorded digitally [31]. The history of motion capture dates to the 1960s, when Lee Harrison III, who was an American animator, used the

recording of real-time human movement for animation. This recording was done via a series of adjustable resistors, cathode ray tubes, and analogue circuits [32] [14].

Motion capture is used in sports, medical applications, entertainment, ergonomics, and robotics. When used in filmmaking and game development, it is combined with the recording of actions of actors for animations and visual effects. Additionally, full body movement, face tracking, facial expression, and finger movements are combined together to create performance capture [31]. In health care, motion capture is used for gait analysis which is to analyse an individual's walking pattern or for kinematic modelling in biomechanics.

Optical passive motion capture technologies use retro-reflective markers attached to the body parts of the individual that reflects light onto a nearby camera lens. From this reflection, the position of the marker is calculated within three-dimensional space and recorded. Optical active motion capture uses the same technique, but rather than reflecting light; the light is emitted [32].

The equipment required for motion capture is extremely costly and is not commonly available in a typical hospital; for example, according to Thewlis et al. [33], a simple Vicon system [34] cost approximately AUD \$250,000 (US \$268,605.52) in 2011 [33]. Even if the equipment is available, it would be difficult to utilise this technology outside research contexts as participants will need to be moved to these motion analysis laboratories to conduct measurements. Another limitation is the need for additional expert staff to run the laboratories for the motion analysis [14].

2.3 REVIEW OF KEY ELEMENTS IN ANALYSIS OF HUMAN MOTION CAPTURE

2.3.1 Use of Coordinate Systems in Human Motion Capture

The orientation of a rigid body with respect to a fixed coordinate system can be described using three angles, referred to as the Euler angles. These angles which have been illustrated in Figure 1 are defined as the roll, pitch, and yaw (ϕ, θ, ψ). It is not important what order these angles are represented; however, the order of rotation is crucial. The rotation matrix for each individual Euler angle is shown in Eq 1 to Eq 3. The ZYX order of rotation evaluated in this study was used to generate the equivalent rotation matrix in Eq 4 describing the three consecutive rotations in Matlab. Note that the individual rotations are applied in reverse order. The order of rotations is critical,

and thus, there exist twelve possible sequences of rotation axis that can be divided into two groups:

- 1) Proper Euler angles: zxz , xyx , zyz , zyz , xzx , yxy .
- 2) Tait- Bryan Angles: xyz , yzx , zxy , xzy , zyx , yxz .

The most common combination used to re-orientate the body from an initial frame at XYZ is using Tait-Bryan angles, namely combination zyx , which consist of a rotation the previously mentioned Pitch, Roll, and Yaw values [35].

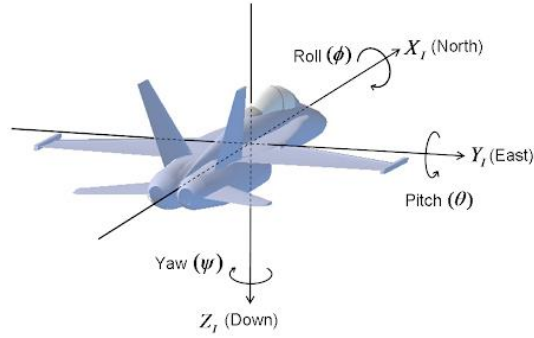


Figure 1: Euler angles using an aircraft [36]

$$\text{Eq 1} \quad R(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix}$$

$$\text{Eq 2} \quad R(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$\text{Eq 3} \quad R(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Eq 4} \quad R(\phi, \theta, \psi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix}$$

Advantages to using Euler angles are that it is easier to visualize and can describe rotation and orientation in a precise manner, but a significant disadvantage is that of the occurrence of gimbal lock which results in the loss of a degree of freedom. Gimbal lock occurs when two out of three gimbals are aligned, this can be translated to a pitch

of ± 90 degrees in the ZYX order of rotation. An example of gimbal lock is shown in Figure 2.

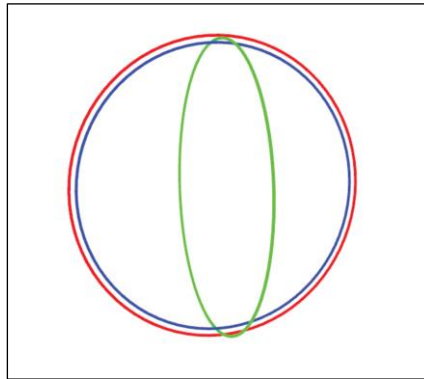


Figure 2: Example of gimbal Lock [37]

Setting $\theta = 90$ and applying basic trigonometric identities to the rotation matrix of Eq 3 results in Eq 4. From this equation it can be observed that changing the roll and yaw values will yield the same rotation matrix. To interpret the raw numerical data, a 3D animation has been developed by [37] which uses a unit sphere to represent the orientation of the wearable device. The orientation of the sphere is constantly updated using consecutive Euler angle measurements and synchronised using their corresponding timestamps. Layout of the orientation frame can be seen in Figure 3.

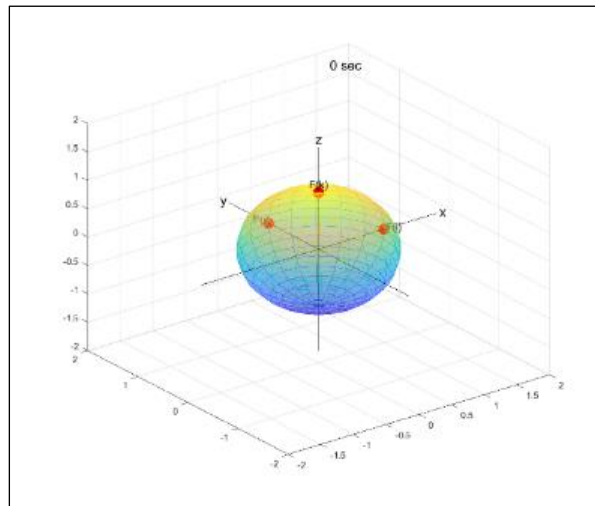


Figure 3: Layout of the orientation frame [37]

The fixed coordinate system is represented by the black lines originating from the centre of the sphere labelled x, y and z which point to north, east and down respectively. The sphere representing the orientation of the wearable device contains

three red markers labelled f(i), f(j) and f(k) which point in the positive x-axis, y-axis and z-axis of an example sensor as illustrated in Figure below. The time corresponding to the orientation is displayed in the top corner of the plot. The basic operating principal behind the Matlab code is application of the rotation matrix described in Eq 5 to a set of coordinates that define a unit sphere [37].

$$\text{Eq 5} \quad R(\phi, \theta, \psi) = \begin{bmatrix} 0 & -\sin(\phi - \psi) & \cos(\phi - \psi) \\ 0 & \cos(\phi - \psi) & \sin(\phi - \psi) \\ -1 & 0 & 0 \end{bmatrix}$$

Quaternions can represent the orientation of a rigid body with respect to a fixed coordinate frame without the added limitation of gimbal lock observed using Euler angles. Quaternions consist of four values, one real component q_0 and three imaginary components q_1, q_2, q_3 as shown in Eq 6. A unit quaternion is defined as an ordinary quaternion with a magnitude of one.

$$\text{Eq 6} \quad \hat{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$$

The symbols i, j, k in the quaternion expression are unit vectors pointing along the three perpendicular spatial axes. The fundamental formula describing quaternion algebra is shown in Eq 7.

$$\text{Eq 7} \quad \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

The quaternion notation used to represent a rotation of θ degrees about an axis defined by the vector $\hat{u} = (u_x, u_y, u_z)$ is shown in Eq 8.

$$\text{Eq 8} \quad \hat{q} = \cos\left(\frac{\theta}{2}\right) + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}) \sin\left(\frac{\theta}{2}\right)$$

2.3.2 Calculation of Smoothness Measurements in Human Motion Capture

In addition to capturing orientation of an object, the captured IMU data can be utilised to provide a smoothness measurement of human movement. Research presented in [38] advises that measuring and quantifying smoothness provides a great merit to rehabilitation as it represents how much control the participants have had on

their movement. According to [39], smoothness of movement in each rehabilitation exercises demonstrates how uninterrupted the movement was given that the following attributes are taken place when considering smoothness:

- Being dimensionless
- Being consistent
- Being sensitive to change in movement.
- Being practical

Smoothness of movement is typically represented by a bell-shape speed provide where movements that lack this kinematic pattern are considered as less smooth [38]. Less smooth movement tends to have multiple peaks so a single observed peak would demonstrate smooth movement. These peaks are referred to as intermittencies and can occur due capability of the performer and nature of the tasks. Intermittency is created by impairments such as a deficiency in motor control or result of an injury. The exercises them can also form intermittencies for instance in point to point reaching where the smoothness is highly dependent on the activity. This means the smoothness of different tasks cannot be compared to each other [38]. Smoothness can be calculated utilising Log Dimension Less Jerk (LDLJ) and spectral Arc Length (SPARC) as stated in [39].

There are several methods for capturing how much jerk exists in a movement as represented by [38] and [40]. Jerk-based calculations are usually not dimensionless, however [40] defines dimensionless jerk (DLJ) and log dimensionless jerk (LDLJ) which both meant the validation criteria mentioned above. Eq 9 shows LDLJ-V as a velocity-based equation derived from [39] which itself is derived from the equation to determine LDJ. The main different here is the fact that LDLJ-V utilises the negative natural log of the absolute value that is provided by LDJ. LDLJ-V uses a normalisation factor captured from the peak velocity to make results dimensionless. In Eq 9, t_1 and t_2 represent the time period of the movement, $v(t)$ is the velocity of the movement in the time domain, and v_{peak} is the max velocity.

$$\lambda_L^v(\mathbf{v}) \triangleq -\ln\left(\frac{(t_2 - t_1)^3}{v_{peak}^2} \int_{t_1}^{t_2} \left\| \frac{d^2}{dt^2} \mathbf{v}(t) \right\|^2 dt\right)$$

$$v_{peak} \triangleq \max\|\mathbf{v}(t)\|, t \in [t_1, t_2]$$

Eq 9

SPARC is another method for smoothness measurement which is dependent on the movement arrest periods (MAP). MAP is defined at the period with no movement and a period with movement is one where some velocity is experienced. This period of movement is the loss in measurements of higher derivatives such as acceleration and jerk as stated in [39]. SPARC uses sub movements across a period to model the movement which can be seen in Eq 10 [39].

$$\lambda_S^v(\mathbf{v}) \triangleq -\int_0^{\omega_c} \left[\left(\frac{1}{\omega_c}\right)^2 + \left(\frac{d\hat{V}(\omega)}{d\omega}\right)^2 \right]^{\frac{1}{2}} d\omega$$

$$\hat{V}(\omega) = \frac{V(\omega)}{V(0)}; V(\omega) = |F(\|\mathbf{v}(t)\|)|$$

Eq 10

In Eq 10, $V(\omega)$ is the Fourier magnitude of $\mathbf{v}(t)$, $\hat{V}(\omega)$ is the normalised magnitude spectrum. ω_c is the adaptive cut off frequency which differentiates Spectral Arc Length (SAL) from SPARC. SAL utilises 40π , but SPARC uses an adaptive range as its cut of point. This is to reduce the sensitivity to temporal scaling up to the determined cut-off frequency as seen in Eq 11.

$$\omega_c = \min\left\{\omega_c^{max}, \min\{\omega, \hat{V}(r) < \bar{V} \forall r > \omega\}\right\}$$

Eq 11

Research found in [38] recommends \bar{V} to be 0.05 and ω_c^{max} to be 20π or 40π as the range keeps segments small enough to be sensitive to intermittency which leads to reliability. Setting 40π as the frequency cut off provides a band that is used to cover the full range of motion from a person's normal and abnormal movement and corresponds to 20Hz.

2.3.3 Gesture Recognition Approaches

Gesture recognition is defined as the mechanism for a system to recognise a physical and predefined action [41]. The predefined actions are known as gestures that can be used to the classly expressive and purposeful motion of the human body [41, 42]. This literature review focused on hand gesture recognition (HGR) which restricts the description above to physical movement of the fingers and hand with the aim of conveying information [41]. This means the definition describes the sole observation of the movement of the human hand irrespective of the human body [42]. HGR has become a staple of the Human Computer Interface (HCI) development environment and has been a highly researched topic over the past 40 years [43] [42].

The primary motivation driving the development of HGR algorithms has been its applicability as a fast, natural, and accurate source of HCI [44]. As HGR is a branch of the general study of human activity recognition [45], it has had applications ranging from simple alternative control use cases to complex human-robot collaboration [46]. Development of hand gesture recognition algorithms began in the 1980s and has continued to be refined and diversified into a range of unique approaches. Modern approaches utilise machine learning to aid in the recognition process [43] [47].

Machine Learning Pipeline

Machine learning is defined as an algorism capable of making decisions outside of the literal definition by an adoption process called training [48]. In the case of classification, machine learning algorithms consist of the following components: data acquisition method, data pre-processing, feature extraction and object classification. These components form the machine learning pipeline, which is the basis for most modern HGR algorithms [43] [48]. The purpose of each individual stage has been explained below:

1. *Data Acquisition Method.* Data acquisition is the first step of the machine learning pipeline and refers to the collection of raw data from a source external to the algorithm [46, 48].
2. *Data Pre-processing.* Typically, the data provided by the data acquisition method is rife with noise and low-value data. To perform feature extraction on such a data sample would be inefficient, and as such, data pre-processing is

used to remove ‘low value’ data from the provided raw data, refining it such that it can be more efficiently used by future algorithm components [48].

3. *Feature Extraction.* Feature extraction further refines the inputted data such that it becomes consistent in shape, irrespective of the supplied image. The purpose of this method is to further reduce the inputted data into a single constant form ‘feature’ where a feature can be described as a meaningful template of data that will be used in the classification process [48].

Object Classification

Classification is the process by which the algorithm generates an educated prediction based upon the inputted data and a pre-trained model. The aim of this prediction is to classify the data as a single element from a pre-defined set. The methods in which a model is trained and from which a model makes its prediction vary based on the application [43] [48].

Gesture Recognition Pipeline

The machine learning pipeline is a general description of how machine learning can be applied to a classification problem. Review documents [46, 47, 49] redefine the above components to make them more specific to the structure of an HGR algorithm. This redefined HGR-specific component list is as follows [43].

1. *Gesture Acquisition.* This is simply the data acquisition method used by a HGR algorithm.
2. *Gesture Description.* This defines the type of gestures being recognised and the primary method in which gestures will be distinguished from one another.
3. *Gesture Identification.* This defines the pre-processing data algorithm, the feature extraction algorithms and gesture tracking components that are used with an HGR algorithm.

Gesture Classification. This defines the object classification algorithm that will transform the gesture data returned by the gesture identification component into a selected gesture.

2.3.4 Data Acquisition Methods

The data acquisition source utilised by HGR algorithms can be defined into two governing categories, these being image and non-image based [46]. These two categories can be further subdivided into the sub-categories illustrated in Figure 4. The image-based category has the following subcategories, marker, depth camera, stereo

camera, and single camera. The non-image-based category has the following subcategories, glove, band, and non-wearable. Non-wearable technologies have been omitted from this investigation as they are an emergent technology with limited implementations available[43] [46] [50].

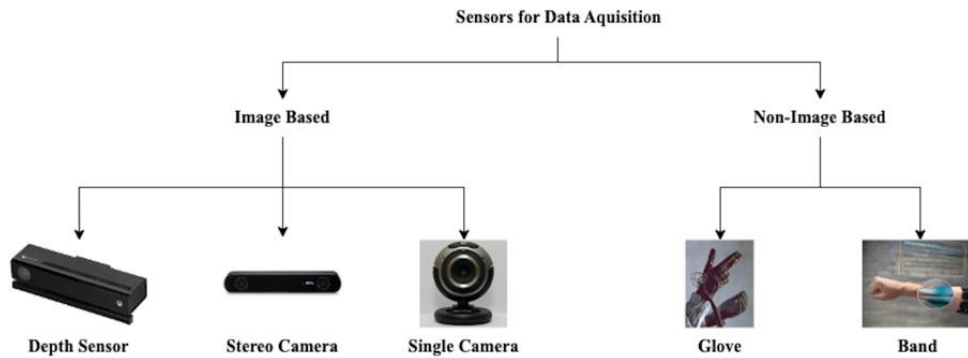


Figure 4: HGR Data Acquisition Categories [43]

Use of Single Camera for Data Acquisition

Single-camera approaches use a single viewpoint RGP camera to monitor the human hand. HGR algorithms that utilise single camera inputs are a mature technology [47] and have been implemented since the early 1990s [46]. The primary drawback for this method is the single viewpoint restriction of the camera which reduces the robustness of the system. This reduction is due to the fact that algorithm cannot directly observe components of the hand that are obscured from the camera’s single viewpoint. This source of error is commonly referred to as occlusion and self-occlusion [46]. Furthermore, single-camera approaches cannot directly observe the depth of hand components making 3D modelling more difficult. The advantages of single camera method is the high observational speed of the camera which requires limited processing to get observational data, natural operation, readily available components, and high-resolution data which reduces the restrictions on range [43] [47] [42, 47].

Use of Stereo Camera for Data Acquisition

Stereo camera HGR refers to any solution that utilises two or more receivers with the aim of producing stereoscopic vision for generating a three dimensional module of the environment [46, 47] [46, 47]. There are different operational wavelengths for these receivers depending on their implementation where the key attribute that separates the approaches is the use of multiple optical receivers. For

example, leap motion is one of the example of the input medium which utilises infrared receivers to capture skeletonised hand models [43] [44].

Advantages of stereo camera approaches are high accuracy, fidelity, and robustness due to use of multiple viewpoint modelling. It is important to highlight that self-occlusion cannot be totally avoided and will still affect the accuracy of the modelling in different positions. This effect is less prevalent compared to single camera approaches. Other restrictions of stereo camera systems are their high computational requires for triangulation of the 3D environment, limited range due to need of focal point of observation, and calibration difficulties [43] [46, 47] [46, 47].

Use of Depth Camera for Data Acquisition

Depth camera HGR approaches are described as any method that utilises non-stereoscopic vision for direct observation of the depth of an environment [46]. The 3D environment can be directly quantified by utilising depth information from a single sensor rather than RGB colour data [51]. Use of depth information is considered as an emergent sector of HGR development with common methodologies such as Time of Flight precepts (ToF) and light coding where light travel time is utilised to ascertain depth information [51]. As example use of this technology can be seen in Microsoft Kinect V2 [43] [44].

Advantages of depth camera approaches is in the removal of lighting, shading, and colour contrast in the data acquisition process which removes the common sources of inaccuracy [51]. The disadvantages of depth camera approaches are in their high cost and range restrictions which reduces the fidelity of available data at extended range. Depth cameras remain applicable for full body gesture recognition at range of 0.5m but they cannot be used in HGR [46] [51]. Occlusion and self-occlusion remain the source of error in depth cameras.

Glove-Based Data Acquisition

Glove based HGR approaches are defined as methods that require the use of sensors directly to the user's hand or a glove for data acquisition [52]. Sensors are used to measure the flexion of the human hand and fingers directly which were the original form of HGR first appearing in 1983 [52]. This method is more matured compared to the others and have an array of implementation scaled for a range of applications [42]. The advantage of glove-based approaches include high accuracy, high data rates, and

fast computational speeds [50]. The disadvantage of the glove-based approach includes being cumbersome, restrictive of the human hand, and relatively long pre-test tuning times [43] [46].

Band Based Data Acquisition

Band based approaches are defined as HGR systems that requires mounts the sensors to the forearm of the user for data collection [46]. This approach utilises electrical and optical sensor to observe the movement of human man using in direct methods [50]. These approaches are a modern implementation of the glove-based approach due to the advancement of surface electromyography [46]. There are several advantages in band based methods such as fast response times, low computational requirements, and not being tied to the hand which makes them less restrictive compared to glove-based approaches [50]. There are still some disadvantages in band based methods such as the need for directly mounting to the user which makes it more restrictive than the visual methods, lower resolution and weaker input signals compared to other HGR approaches, and they are affected by differing location of the sensor attachment [43] [50].

2.3.5 Gesture Description Overview

Gesture description refers to the information represented by gesture and the method for modulating within the HGR algorithm [53]. There are three categories in gesture description which are physiological scope, the information interpreted from the gesture, and the model used for representing the gesture [54]. There are different considerations to be made on each of factors which has been explored below [43].

Physiological scope refers to the pre-set taxonomy used to determine the physical nature of the gestures [54]. Upon review of the literature, the use of static or dynamic gesture sets, the inclusion of wrist motion, and the number of hands used to form the single gesture has been noted as the main distinction factor.

The information interpretation based on HGR algorism can be divided into spatial information, temporal or pathic information, and symbolic information [41]. Spatial information consists of the position of the gestures within the environment. Temporal or pathic information is interpreted from velocity and the path an observed gesture takes within the environment. It is important to highlight that spatial information is typically observed utilising the world coordinates of the observed

gesture similar to spatial information. Symbolic information refers to the shape of the observed gesture and is usually interpreted utilising joint angle calculations or shape matching methods [41].

The model used to represent an observed hand varies depending on the desired scope of input gestures [54]. Depending on the increase of number, complexity, and information density of the gesture, the model becomes more complex, and a relevant modelling method will also need to be implemented. The complexity of the mode is proportional to the to the number of classifiable landmarks that has been provided within the model. HGR modelling can be categorized as 3D based modelling and appearance-based modelling as scene in Figure 5 [43] [54].

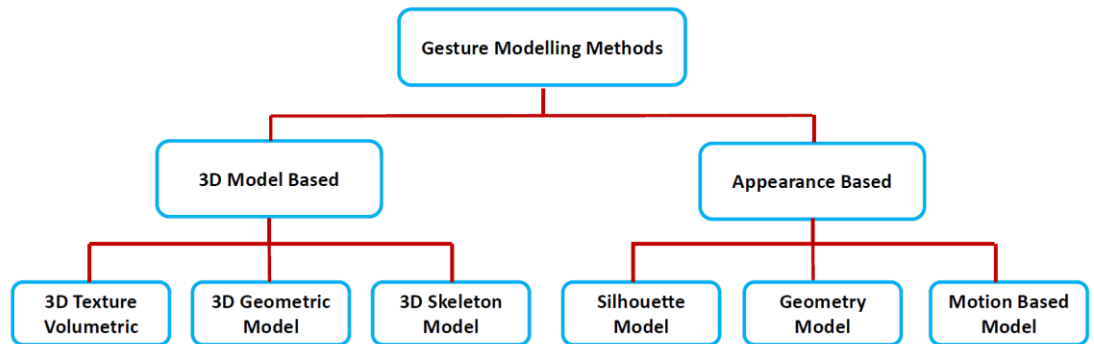


Figure 5: Summary of Hand-gesture modelling methods [43]

There are models with less complexity such as the silhouette geometry model, which is used for simple HGR applications, but they offer few classifiable landmarks. This model is appearance based which lowers the computational requirements and can be extracted directly from the image with little intermittent computation. These models are better suited for low response time algorithms that focus on being lightweight with fast operating applications [55, 56] [55, 56]. More complex models such as 3D skeleton model could offer up to 21 landmarks for classification and require more computational power to be accurately generated. Due this this higher computational power, they are usually utilised in control applications where accuracy and reliance on more expansive data sets are required [55, 56] [55, 56].

2.3.6 Application of Machine Learning in Gesture Identification

Gesture identification refers to methods for detecting human hand apart from its background with the aim of generating a computer model for classification also known

as feature extraction [42, 46, 48] [42, 46, 48] [42, 46, 48]. The following sections focuses on single camera visual observation methods, 3D skeleton representation, and 2D silhouette models [43] [44].

Use of OpenCV in Gesture Identification

Colour segmentation and shape or colour feature extraction are among the most basic form of feature extraction for single camera RGB data [54]. Colour segmentation method utilises the unique consistency of the human hand colour to differentiate the hand from the background. The advantage of segmentation is the low computational requirements, however, only 2D models can be produced such as silhouette geometry models. Additionally, the accuracy is negatively affected due to self-occlusion and variances in the background, skin tone, and lighting conditions [57]. The unique shape of human hand is used in shape analysis to aid in the detection and is usually achieve by contour or edge detection methods such as Fourier Descriptors, pr histograms of oriented gradients. Similar to segmentation, shape analysis can only be used for calculating a 2D model however, it is more resilient against background, operator, and lighting variances [57, 58]. Shape analysis methods require more computational power compared to colour segmentation and still have self-occlusion and robustness issues [43].

As mentioned above, these methods cannot be utilised for generation of 3D models, however, their low computational requirements make them still viable in modern HGR algorithms. For example, colour segmentation can be used to locate areas of interest within the image, followed by more robust algorithms for pre-processing which can lead to a reduction in the amount of data sent to the more complex identifiers [57, 58] [57, 58]. The existence of opensource libraries such as OpenCV has also helped in widespread application of these algorithms. OpenCV contains python modules capable of performing the aforementioned feature extraction methods efficiently and can be used as the building blocks for many other approaches as seen in [44].

Use of MediaPipe Hands in Gesture Identification

Media Pipe Hands (MPH) is an on device real time hand identification solution designed to operate using data from a single RGB camera [55]. The output of MPH is a list of 2.5D, 21 landmark skeleton model for each observed hand within the frame. MPH contains a 2-stage pipeline with the first stage being a palm detector, followed

by the second stage which is the hand landmark extraction model. This pipeline minimises the computational load of 3D skeleton identification process in two key methods. First method uses a light computational algorithm to locate areas of interest in the image and then applies the landmark model on the located areas. The second method uses the tracking of identified hands between frames which reduces the computational power needed to identify subsequent frames [43].

BlazePalm detector is the first stage of the MPH pipeline which detects the hand locations and deals with challenges such as: variability of hand physical appearance (colour, size, pose), large scale span, occlusion issues due to use of single perspective, and low contrast patterns within the hand. To solve these issues, the BlazePalm detector utilises 3 unique strategies. The first strategy is the utilisation of palm and fist detection over whole-hand detection which addresses the issue of self-occlusion and pose variation. It is important to know that whilst fingers are commonly self-occluded, fist and palms cannot be. The second strategy is the use of an encoder-decoder extraction method such as FPN which addresses the problem of low contrast patterns and colour variation. The final strategy is the minimisation of focal loss during training with the aim of combatting the high variance in scale and hand size. The combination of these strategies provides an accuracy of 95.7% for palm detection followed by placing a bounding box around the wider 2D silhouette feature surrounding the palm [59, 60] [59, 60].

The second stage of MPH pipeline applies regression algorithms on the image encapsulated within the bounding box to locate the previously mentioned 21, 2.5D joint landmarks. 2.D here refers to the use of x and y coordinate taken relative to the image's orthogonal frame and calculation of the z coordinate relative to depth for the landmarks. The relative depth values are acquired based on perspective angle of the camera, perceived distance of the new landmark and the palm of the hand. The 21 landmarks provide the estimated joint calculation of the hand. MDP can provide outputs representing probability values for the confidence of the algorithm on the 21 landmark prediction and binary classification values to the hand being left or right. The regression model is used via real world data and synthetically generated images to improve accuracy in different environments and relative depth calculations [59, 60].

Reports provided by [60] indicate average precision of 93.33% for the MPH pipeline which is very promising considering the straight forward installation of MPH

via pip or official webpage. MPH has some key issues such as, decreased accuracy when palms are placed together and rotational inconsistencies [59, 60].

Use of InterHand2.6M in Gesture Identification

InterHand2.6M (IHM) is a relatively new algorithm that uses single RGP camera alongside a pre-trained convolution neural network (CNN) which has been labelled as ResNet and provides highly accurate feature extraction [56]. The output of IHM is a normalised 3D, 21 landmark skeleton model for 2 hands which is tuned to detect left and right hand a single operator [43] [56].

The previously mentioned ResNet can be trained using an application specific data set or using the extensive data set labelled InterHand2.6M provided by IHM. This data set contains annotated data which can be utilised for training ResNET. The trained model has provided high accuracy when observing gestures involving two interconnected hands [56]. The InterHand2.6M dataset was collected in a multi camera studio consisting of 140 cameras and 450 bidirectional LEDs. 26 unique set of human hands were observed while performing 53 total gestures per person. This data was then annotated with the 3D landmarks using a semi-autonomous approach. The extensive nature of this data collection process is one of the reasons contributing to the high observational accuracy of the mode [56].

A ResNet model is placed within an InterNet wrapper after being trained. InterNet utilises a single frame of RGB as its input and sends it to ResNet for feature extraction and outputs the initial hand features. InterNet then uses the features to generate a 2D coordinate set and the relative depth of each of the 21 landmarks that have been previously mentioned. Camera back projections and an inverse affine transformation which is a transformation that preserves lines and parallelism are then applied to the coordinates to create the final normalised 3D list of landmarks. The 3D coordinates are transposed onto an image window so visual validation can be conducted [56].

Both InterNet and InterHand2.6M datasets are open source and readily available on PyTorch and github. There is some fine tuning required as the initial download of the data bases are tuned for static image recognition. To be able to utilise InterNet for real-time video up to 30 frames per second, ResNet will need to be re-trained. InterNet has shown 99% accuracy across several indoor and outdoor environments for gesture

sets which includes 2 hand interactions [56]. Due to high computational requirements for InterNet, there is limited literature available covering its use in applications.

2.3.7 Gesture Classification

Gesture classification is the process where features are extracted by gesture-identification algorithms and classified within a pre-defined list of gestures [42, 61]. To classify input-gesture models typical machine learning techniques can be utilised. Among the widespread approaches are decision trees, K-nearest neighbours (KNN), Hidden Markov model (HMM), Artificial Neural Networks (ANN), Naïve Bayes (NB), Linear Regression bounds, Support Vector Machines (SVMs), and Convolutional Neural Networks (CNN). The use of classifiers that can hand high dimensionality features spaces and classification of element into distinct non-linear classes are more desirable [61]. These algorithms are further explored in the following paragraphs [43].

Use of Decision Trees in Gesture Classification

Decision trees operate via a tree like structure which consists of decision nodes that represent points of classification and leaf nodes that represent classifiable classes. Decision trees are typically implemented via intuitive, light weight, white box methods where simple, high-level comparisons are utilised at each node level to classify the given data. Decision trees are best used for simple classification problems and small gestures such as the ones mentioned in [62]. More complex variants of decisions trees which are known as random forest classifiers can be utilised for classifying multidimensional landmark models without overfitting [43] [44, 62] [44, 62]. Overfitting happens when ML algorithms produce accurate results for training data set but not for new data sets.

Use of Decision SVM in Gesture Classification

SVM methods use a trained hyperplane as a binary classifier to separate the classifiable classes. The hyperplanes can be utilised to separate large number of classifiable classes across multi-dimensional space even in non-linearly separable cases [62]. The hyperplanes are generated while training the model by utilising the closest values between neighbouring classes and defining the hyperplane to separate the values [42]. The dimensionality of the data is increased until the linear separation of the points is possible [42]. SVM classifiers have high classification speeds

considering their utility for multiple-dimensional data classification. It is important to note that SVM methods are prone to overfitting so the training needs to be done with care [43] [42].

Use of CNN in Gesture Classification

CNN methods are becoming more popular even outside HGR applications [62]. CNN methods operate on neural network models where a given feature is passed through a various interconnected hidden layers, which uses confidence value calculations to build up confidence value for the final classification [42]. CNN algorithms can recognise subcomponents of the gesture and use them to provide accurate predictions for the gesture. As mentioned in [62] and [44], carefully trained CNN can achieve classification accuracies close to 100%. The issue with CNN is the reliance on large training data sets and hyper parameter training [46, 57] [46, 57]. The Hawks Harris algorithm detailed in [57] can nullify the hyper parameter tuning in HGR applications [43].

2.4 CURRENT PRACTICES IN HUMAN JOINT REHABILITATION UTILISING TRADITIONAL METHODS AND GAMIFIED APPROACHES

2.4.1 Current Clinical Approaches for Human Joint Rehabilitation

Before exploring gamification in rehabilitation, current human joint rehabilitation methods need to be investigated. Rehabilitation is defined as set of exercises that reduces disability in individuals with respect to their muscles to allow them to interact with their environment effectively in their day-to-day activities [1]. There are multiple reasons where individuals may require rehabilitation such as pre-existing conditions, surgery, old age, after injury. It is estimated that 2.4 billion people worldwide have conditions that would benefit from rehabilitation based on articles by World Health Organization [1]. This large number highlights the importance improvements to medical processes for rehabilitation, which is a difficult challenge due to varying nature of rehabilitation requirement going from person to person [1]. The person going through rehabilitation may need to receive assistance in moving a body part with the presence of a qualified clinicians and often there is a requirement to complete exercises outside of clinical environments [1]. To better understand different requirements for rehabilitation, wrist rehabilitation for people recovering

from injuries and people with chronic conditions have been reviewed in the literature below.

Rehabilitation for Paediatric Wrist Impairments and Injuries

Paediatric patients in Australia are classified as age range of 4 months to 17 years old [63]. Rehabilitation for chronic or short-term wrist injuries or impairments are a common issue for children. For example, a 10-year study in Sweden [64] defines upper extremity fractures are among 68% for the age of 0 to 18 years old. These fractures are often treated in a short time frame but are followed by physical therapy to improve strength, flexibility, achieve the required range of movement, and to prevent further re-injury [64].

The issue becomes more complex with longer term and chronic patients that suffer from neurological conditions such as Cerebral Palsy (CP) or joint inflammation such as Juvenile Arthritis. These conditions can affect mobility in the upper limbs and would require longer term rehabilitations. Some of these conditions such as CP are either lifelong or can become chronic such as Arthritis which means the long-term conditioning programs need to be utilised to maintain and improve ROM (range of motion). Approximately 1 in every 1000 children develop a type of chronic condition in case of Arthritis [65]. CP is one of the most common motor disabilities in children and it affects 1 in 1000 births [64]. It is reported that spastic limb movement accounts for 82.9% of all CP cases which affects the range of motion in joints and very common for wrist joint of many patients [66]. The limited range of motion caused by muscle stiffness and tightness of the wrist joint will lead to other problems such as weakness in grip strength [66]. These statistics highlight the importance of maintaining rehabilitation and therapy intervention where the symptoms need to be maintained and managed by the therapist, leading to ease of participation in day-to-day activities.

Physical Therapy and Hand Rehabilitation

As earlier in this chapter, rehabilitation is needed to maintain function, flexibility, strength and range of motion after an injury or in presence of a movement disorder such as CP [1]. Conditioning programs in rehabilitation usually contain a set of exercises for patient to repeat over a period to improve or maintain their functionality. The exercises related to improving range of motion, two categories are defined which are passive and active [67].

Passive exercises are assisted by another person, for example a wrist maybe held to its maximum extension angle with the clinician physically pushing the patient's wrist and holding it at this point. These exercises will improve blood flow and sensory stimulation and prevent muscle stiffness [66].

Active exercises involve the patient performing the exercise without the clinicians physically participating. The benefits of these exercises are stimulating neuroplasticity which helps with rewiring of the brain more compared to passive exercises, strengthens the muscle, and improves ROM [67].

Both passive and active exercises are required for full recovery, and they need to be performed consistently. The exercises will need to take place at home outside the clinical settings and the repetitive nature of the exercises sometimes leads to lack of motivation by the engagement. The lack of feedback received outside clinical settings are also a factor leading to abandonment of exercises [68, 69]. That is why the idea of gamification has risen to popularity in the past couple of years as it aims to utilise games to increase engagement in rehabilitation exercises. Gamification is further explored in Chapter 2.4.2.

As example of rehabilitation exercises can be seen in hand exercises for wrist movement given below:

- Flexion and Extension (moving the hand down and up)
- Ulnar deviation and Radial deviation (tilting the wrist side to side)
- Supination and Pronation (rotating the palm to be positioned up and down)

The therapist may provide a specific type and length of exercises depending to the impairment of injury to the wrist joint. For example, Figure 6 and Figure 7, show example set of exercises to improve ROM and strengthen wrist muscles. In case of injuries, medical professionals need to ensure the tissue is healing correctly and the bones are healed from the fracture before offering rehabilitation exercises.

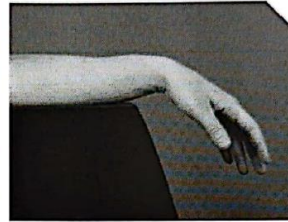


» Wrist Exercises (Active)

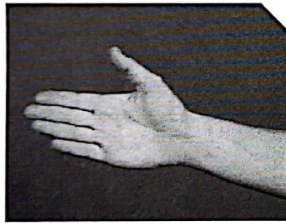
Lift wrist up fingers
fold in



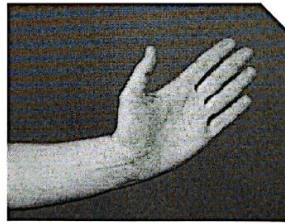
Drop wrist down
fingers open



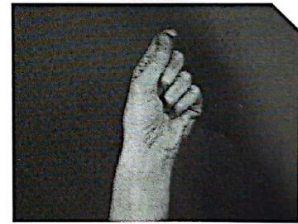
Hand shaking forward



Hand shaking back



Wrist circles,
painting the ceiling



Palm up



Palm down



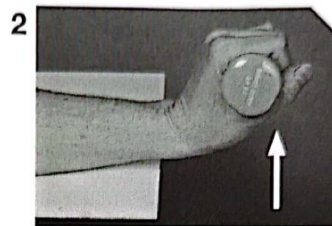
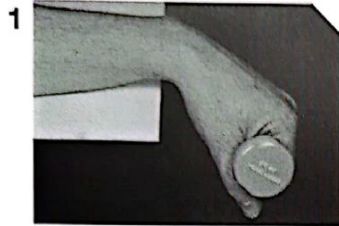
Figure 6: Active Wrist Exercises Examples (Provided by Royal Perth Hospital in Western Australia)



Wrist Strengthening

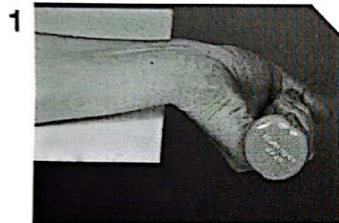
Perform each exercise 5 times, 3 times per day

Wrist Extension



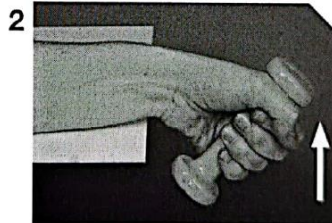
Lift wrist up and down
Weighted stretch
Hold Position 1
For 20-30 seconds

Wrist flexion



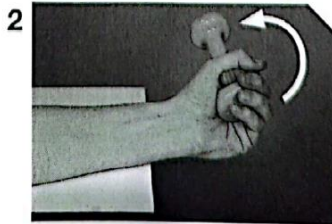
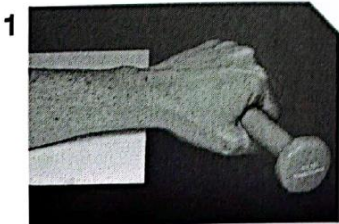
Roll palm up and down
Weighted stretch
Hold Position 1
For 20-30 seconds

Ulna and radial deviation



Lift wrist up and down
Weighted stretch
Hold Position 1
For 20-30 seconds

Pronation and supination



Roll palm up and down
Weighted stretch
Hold Position 1
For 20-30 seconds

Figure 7: Wrist Strengthening Exercises Examples (Provided by Royal Perth Hospital in Western Australia)

2.4.2 Review of Existing Methods for Gamification of Rehabilitation

Gamification in rehabilitation has become popular in the past decade with advancement of technology [70]. It is becoming more and more important for develop and facilitate rehabilitation in home-based therapy system and provide hands off approaches for example people in remote areas [68]. Games developed for upper-limb rehabilitation can produce greater functional outcomes compared to the typical home-based therapy sessions as represented reported in [71]. Gamification is capable of providing an engaging experience by providing challenges to be solved by the patient, leading to an increase in the motivation and more successful completion of the rehabilitation program at home [65].

Study by [72] provides a patient-centred serious game utilising a leap motion controller incorporating hand movement exercises. In this implementation, the game was developed for patients who had physical impairments which took effect in personalization of the experience. The game it was consisted of a flying style game where wrist exercises were mapped to fly high, low, left, and right on a given pathway. Predefined routs would be followed using hand gestures while receiving visual and auditory feedback depending on capabilities of their participant. Patients real time movement parameters would be displayed such as ROM and joint angles with capability of playing back the session data. The session included leap motion data, player data, and generated path data. The study had five female patients in age range of 18 to 30 years. One of the participants had CP with others recovering from finger or wrist trauma. Patients found the game to lack engagement due to a single navigation style and existence of a single game. The game was also designed for wide range of ages, so the younger patients did not find the game entertaining. This study showed movement data in all direction as X, Y, and Z orientation. The clinicians suggested hiding unused directions so more focus can be given to the active axis of movement relevant to the current exercise. [72].

Fruit Catcher is another rehabilitation game with the objective to get fruits in a basket developed by [73]. This game was designed for people who have recently been discharged from the hospital and require home based rehabilitation. The game utilises a Nintendo Wii Balance Board as the sensor for capturing movement. Difficulty parameters were designed for the game such as size, number, and fall frequency of the fruit. Number of repetition and time could be set by the therapists which provided

useful for adaptability of the game. This solution could detect if the patient was compensating for the lack of ROM and adjust the gameplay accordingly. The game provided instant feedback if it detected improper movements, but it still needed to be closely monitored by the clinicians. The issue with the Wii board is not being capable of capturing accurate data from the wrist joint application and enjoyment was limited to a single game [73].

Another gamified example can be seen in the work done by [69] for Juvenile Idiopathic Arthritis with wrist inflammation. A series of mini games were developed using leap motion sensors as the input measure. The developed solution provides input parameters, track actions, and data recording to the clinicians. The patient sessions could be replayed, and the levels were divided into two categories:

- Random game levels with specified constraints for the player.
- Creation of set game levels via clinician input

The games targeted wrist exercises such as flexion, extension, radial and ulnar deviation. The game consisted of rhythm games, flabby bird clone, skiing game, and plane simulator [69]. The mini-game aspect was found to be intuitive since patients required very few trials to interact and perform exercises to match the gameplay. The rhythm game and skiing game were the most difficult requiring more tuning such as widening the flags in the skiing game depending on the patient's ROM [69].

These solutions highlight some of the benefits and challenges of designing games for rehabilitation purposes. As it was observed, all solutions point to increase in engagement with exercises given the correct targeting of the age group. Also, the importance of more simple mini game collections has been demonstrated so the exercises can be done in quick sessions. The issue with the current solutions is the lack of clinical validity of the data provided during or after the play sessions. Even though improvement trends can be monitored via the solution provided by [72] and [69], data captured from motion leap devices have not been clinically validated meaning the joint angles reported with these solutions are not representative of the actual wrist or finger joint angles. The framework developed for this thesis will provide guidelines of designing exercises specific games focusing on a target rehabilitation goal with the aim of clinically validated movement information to be provided to the clinicians.

2.4.3 Fundamentals of Game Design Principles for Rehabilitation

Gamification is defined as the application of game principles and game design elements in non-game situations [74]. The framework of the game design has many popular theories and fundamentals however one of the most recognized on is the Elemental Tetrad as shown in [75]. As the name suggests, there are 4 elements in Elemental Tetrad that are crucial for all games as defined below:

- **Mechanics:** This is the procedure and rule of the game such as space, objects, rules, actions, chance, and goals.
- **Technology:** This is the materials and methods of interactions for delivery the game play such as computer or sensors
- **Aesthetics:** This demonstrates how the game looks, sounds, smells, tastes, and feels
- **Story:** This is the sequence of events that unfold in the game and according to [75] falling blocks in Tetris can be considered the story element.

There are numerous studies in recent years that integrate the four elements into rehabilitation applications. Applying the Elemental Tetrad becomes more challenging in rehabilitation and home based solutions as games will need to be developed and modified based on patient's capabilities [76]. The exercises within the games should help the recovery process in a safe environment without any harm or damage to the patient. Another study suggests that games in rehabilitation must be intuitive and have a sense of achievement to encourage and motivate participation in the rehabilitation program [77].

A study carried by [2] provides the following features as important elements to be included in to consider when game theory is applied to rehabilitation settings:

- Precise data recording
- Feedback for both clinicians and patients
- Positive and negative feedback with the aim of motivating and engaging the patient.

- Provide challenge without being frustrating.

2.4.4 Review of Key Components for a Gamification Framework

This section provides a review of key components of game design which are relevant to rehabilitation. It is important to note that the following elements will have varying importance depending on type of rehabilitation exercises currently taking place. As previously mentioned, rehabilitation will need to be fine-tuned on a person-to-person basis due to reliability on location, time, and treatment of patients after injury or nature of underlying conditions such as CP or Arthritis. This is why customizability of the games is one of the most crucial aspects of gamification since they should be adapted to different needs.

Patient as the Players

Patients are the target demographic for gamification of rehabilitation and the age of patients can influence their engagement and level of interest [75]. The story of the game needs to be clear and simple, and it will capture the initial appeal for the player. For example, [78] provides a cow milking mini game where the mood of the cow changed when on the farmer (who is the stand in for the player) makes the correct movements. The mood of the cow here creates an emotional link to the player and encourages them to engage with the game more while performing actions that feel more meaningful.

Game Mechanics

The patient going through rehabilitation, will need to perform actions or exercises in a 3D space. For example, [79] provides a hen house mini game where the patients perform wrist movement exercises to collect rolling eggs in a basket. The rules of the game encourage the actions which make player actions impact the outcome of the game as required by the concept of game mechanics [75]. The application of this design can promote autonomy and self-driven motivation from the player as their movements have a direct impact on the game.

As the challenge increases and the game progresses, there are ways to motivate the player by various levels, achievements, streaks, and unlocking new features. In rehabilitation, choosing the difficulty level is important to make sure the patient is performing their exercises at the appropriate level. The research done by [80] demonstrates a method for dynamic change of difficulty. This approach utilised

success and failure rates of patients to adapt the difficulty and keep the patient motivated. For example, upon multiple failures the difficulty would be lowered so the patient does not become discouraged from continuing the exercises. The study showed an increase in the number of correct tasks that were completed in a given period of time compared to randomized difficulties. The main problem with this approach was the lack of control given to the clinicians in adapting the difficulty since all the decisions in regard to the difficulty were taken by the game.

Game Components

There are patient specific design elements such as points, avatars, tasks, and messages which are known as game components. These elements can be designed and finetuned based on different patient requirements. A study by [7] highlights the importance of correct engagement and completion of rehabilitation exercises. A point system can then be utilised as the gold of the game to encourage the patient's actions as they do the exercises correctly. It is important that this point system provides a safe and adjustable experience for the patient as different patients may have different points required for their exercises.

The game dynamic refers to the way each game component interacts with the game itself [75]. The mobility and patient condition will need to be considered when designing how a patient is going to interact with game components. There is a fine line between making sure exercises are being done correctly while avoiding exhaustion. A study by [78] recommends the following dynamics for different types of rehabilitation: free movement, touch the target, catch the target, follow the path, move the target, and point & shoot.

Game Technology

One of the crucial components of gamified rehabilitation is the input devices used for capturing movement of the patient. It is important for the technology to be non-invasive, Low cost, and accessible for home-based use. Previously in chapter 2.2 multiple human motion capture technologies were explored. The developed framework in this thesis utilises a combination of optical motion capture technologies and inertial measurement units as the game technology. The implementation of this technology is explored in Chapter 3.

Providing Feedback

It is essential to monitor a patient's progress and provide feedback on how correctly the exercises are being performed. This feedback should be provided to both the patient going through the exercises and the clinician monitoring the exercise. The feedback can be either positive or negative to help the patient correct their movements and be encouraged to continue the exercises.

This feedback will help the clinicians adapt the exercises based on how a patient is improving during their rehabilitation program and as a result the games will need to be modified accordingly. The system developed by [77] allowed the clinician to select the game parameters depending on the patient requirements. This customization provided a less frustrating experience for the patients as it encouraged continued engagement with the game and exercises. In some instances a camera was utilised so the clinician could remotely monitor the movement of the patients outside the game environment as seen in [2, 77] [2, 77].

Researchers such as [81] suggest providing feedback in a graded form or absolute form to increase the efforts of motor learning. Negative feedback is also important to improve the patients' skills while interacting with the games since the patients will need to know how accurately they are doing the exercise [82]. Feedback can be provided via reward-based systems or through simple messages provided to the patients.

Achieving Rewards

Game developers sometimes construct game design principles via dopamine responses as a means to increase engagement. Dopamine is a neurotransmitter by human body's nervous system to send message between cells and plays a role in how we feel pleasure [83]. The game rewards can take the form of virtual rewards such as power ups or achievements in the game [84]. Rewards could extend to real world benefits such as monetary rewards in a casino style game. In rehabilitation there the increase in range of movement after engagement with the game could be considered one of these rewards which could be followed by virtual badges and achievements in the gamified environment.

Optimal Challenge

As previously mentioned, the challenge level experience by the player needs to be considered while designing levels. Ideally, the games will start with low difficulty levels so the user is motivated to engage with the rehab program. This methodology is to counter proficiency of the control by the player and promoting the increased range of movements as selected by the clinicians [85]. The challenge here is that from the player's point of view lack of failures could create an uninteresting gameplay experience leading to abandonment of the rehabilitation program. Study done by [86] describes this feeling as 'Fiero!' which is the Italian word for proud which occurs somewhere between frustration and relief.

Clear Goals and Instructions

There are multiple ways clear goals can be perceived. For example, the players path should be clear which is implemented by utilising light and contrast to guide the player in a certain path. This implementation can be defined as a goal-oriented game design where the player needs to identify long term and short-term goals of the game. Feedback provided via the user interface, animation, and visual effect can inform the player regarding how close they are to these goals [87]. Clear goals can inform external goals such as the goal of performing well in rehabilitation [4]. Unclear goals could frustrate the patients leading to low motivation as it has been mentioned multiple times so far.

Motivation in Games

Motivation, as it has been seen so far, connects all aspects of the game design. A well designed game, will utilise different game components to encourage a dopamine response via gameplay leading to further play [83]. Motivation can be considered a psychological aspect as a result of a well design goal-orientated action [83]. The motivation in context of rehabilitation can be extended to extrinsic motivation such as self-motivation by engaging with the rehabilitation process [84].

Achieving Flow

The concept of flow has been put forward by a psychologist as seen in [88] where the flow is defined as "state of concentration or complete absorption with a given activity such that nothing else seems to matter". The paper suggests that this concept

can be applied to art, work, sports, and gaming. This state of “Flow” can be achieved by the following points:

- Defining clear goals and rewards
- Loss of sense of time
- Immediate feedback on successes and failures
- Adaptable challenge level
- Providing a sense of control

These tenants directly relate to game design philosophies presented before so clear goals, rewards, feedback, and challenge makes games one of the best candidates for achieving the “Flow” state.

Use of Lense in Game Design

One of the popular books in game design written by [75] provides a detailed look at the art of game design. This book suggests that good game design requires the developer to keep different perspectives in mind. These perspectives are defined as “Lenses” to consider when designing games. This concept can be applied to gamified rehabilitation such as Lense of patient, guardians, and clinician as the main crucial perspectives in gamified rehabilitation.

Emotions in Games

The article by [86] discusses the concept of emotions in games which provides five emotions to consider which are enjoyment, focus, decision, performance, and learning. Enjoyment contributes to creating an emotional connection between the player and the game leading to increased motivation. Focus facilitates active engagement with the game play achieved by feedback and reward systems that were previously discussed. Decision in this context refers to making decisions driven by emotional responses. Performance is demonstrated as different methods for engaging with the gameplay. Learning is defined where the game facilitates motivation to repeat and master an action. These topics build on the philosophies of game design and can be utilised to evaluate what a user experiences. There are further definitions provided by [86] that discuss different levels of enjoyment as hard fun, easy fun, serious fun, and people fun set on a sliding scale between goal-orientation and open ended design.

2.5 SUMMARY AND IMPLICATIONS

This chapter provided a review of several technologies used for capturing human movement, it provided a review of key elements in analysing human movement data, and an assessment of current practices in human joint rehabilitation in clinical and gamified settings.

The review of technology showed optical motion capture technologies such as Vicon to be the most accurate representation of human movement. However, this technology is very expensive and required specialised facilities for capturing human movement data so it would be suitable for rehabilitation at home. The review of sensor technology demonstrated that Inertial Measurement Units (IMU) provide the best balance between cost, performance requirement, and accuracy for human movement. IMUs have already been utilised in several clinical applications as seen in the literature which makes them suitable for this application.

The review of key elements in analysis of human movement data showed that machine learning should be utilised to get a clinically accurate representation of human movement. It was found that in order to avoid the cost and processing power requirements associated with machine learning approaches, trained algorithms such as MediaPipe should be utilised. Since the literature did not provide a significant number of clinical validations of this technology, an evaluation of this method was done as part of this thesis and has been reported in Chapter 5.

The review of current practices in rehabilitation and gamified approaches showed multiple sources on the applicability of rehabilitation. It was found that engagement and motivation can be improved utilising gamified elements. The review of the literature did not provide too many instances on clinically valid data in the designed games. It is true that most solutions provide data and trends to clinicians, but most did not provide a clinical validation on accuracy or reliability of the information. This thesis provides a framework for gamification where the provided data can be utilised directly by the clinicians both during exercises sessions and for long term remote monitoring. One of the common themes in the literature was the need for customizability of the games as rehabilitation exercises will need to be adjusted depending on different patient needs.

Chapter 3: Experiment Design

3.1 OVERVIEW OF THE METHODOLOGY FOR EXPERIMENTAL DESIGN

This chapter provides a detail overview of the experimental design for the gamification framework including the hardware requirements and specification, machine learning components, visualisation elements utilising the Unity game engine, as well as development of the gamified features. From a design standpoint, this will elucidate how affordable technology was employed to capture human movement in a manner that maintains clinical validity. The selection process investigated technologies that can be used outside clinical settings so that the user can receive feedback on how accurately exercises are being performed. This aspect additionally offered a way for clinicians to gauge progress, as the framework facilitates the assessment of the rehabilitation program's effectiveness for the user. This chapter also provides guidelines for developing games for rehabilitation purposes as well as design templates for several game archetypes.

3.2 DEFINING HARDWARE REQUIREMENTS FOR THE FRAMEWORK

3.2.1 Hardware Requirements and Sensor Selection

The developed framework provides the means of monitoring human movement whilst participating in rehabilitation exercises. This framework provides feedback to the user, so they get an understanding on how correctly the exercises are being performed. To facilitate the capabilities of the framework, several requirements were defined such as capability for recording live data, providing visual representation of the movement, capability to customize both the exercises and the games, and providing measures of joint and smoothness. The diagram in Figure 8 illustrates the hardware requirements [16].

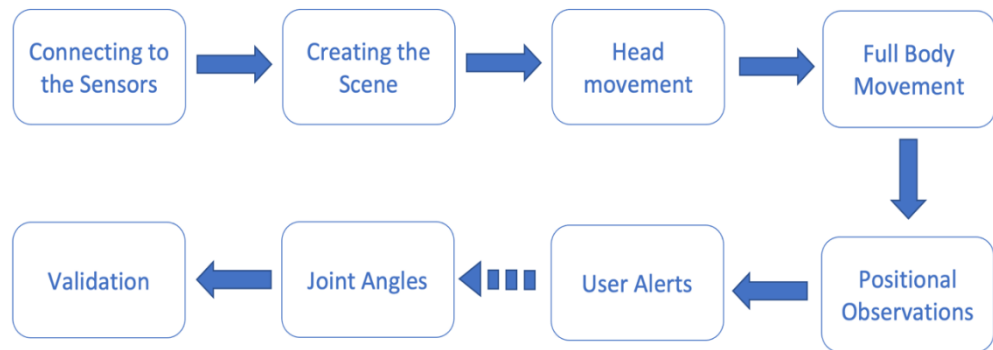


Figure 8: Process Diagram for Hardware Requirements [16]

The sensor selected for this project needed to be small, lightweight, have long battery life, utilise BLE 5.0 (Bluetooth Low Energy), have high sampling rates, provide continuous measurement, and have low cost. The reason for BLE 5.0 was due to robustness and having 8 times more data transmission speed compared to BLE 4.3 and BLE 2.1 [16] [89]. Other communication protocols such as Wi-Fi and Zigbee were considered but eventually ruled out due to availability of BLE in most commercial sensor devices as well as smart phones and laptops, which helps in ease of adaptability. The sampling rate requirement was defined based on the research by [90] and [91] which state the sampling rate requirement for accurately capturing human movement needs to be at least 15Hz for slow movements at least 60Hz for fast movement.

Table 1 and Figure 9 provide a comparison of several off the shelf IMUs that were considered based on their sampling rate, connectivity, batter life, weight, size, and price. Xsens Dot (known as Movella as of 2022) was selected as most suitable since it provided the best compromise of battery life, sampling rate, cost, and performance compared to others [16].

Table 1: Comparing off the shelf IMUs [16]

IMU	Sampling Rate	Connectivity	Battery Life	Weight	Size	Price
Xsens Dot [92]	120 Hz	BLE 5.0	9 h	11.2 g	36.3 × 30 × 10.8 mm	€495.00 (~\$798.05 AUD) for 5 pack
Vicon Blue Trident [93]	100 Hz	BLE 5.0	12 h	9.5 g	42 × 27 × 11 mm	\$1600.00 USD (~\$2184.36 AUD) each
Shimmer IMU [94]	128 Hz	BLE 2.1	14 h	23.6 g	51 × 34 × 11 mm	€359.00 (~\$578.79 AUD) each
Bonsai IMU [95]	50 Hz	BLE 4.3	16 h	15 g	36.5 × 32 × 13.5 mm	€2490.00 (~\$4014.44 AUD) for 15 pack

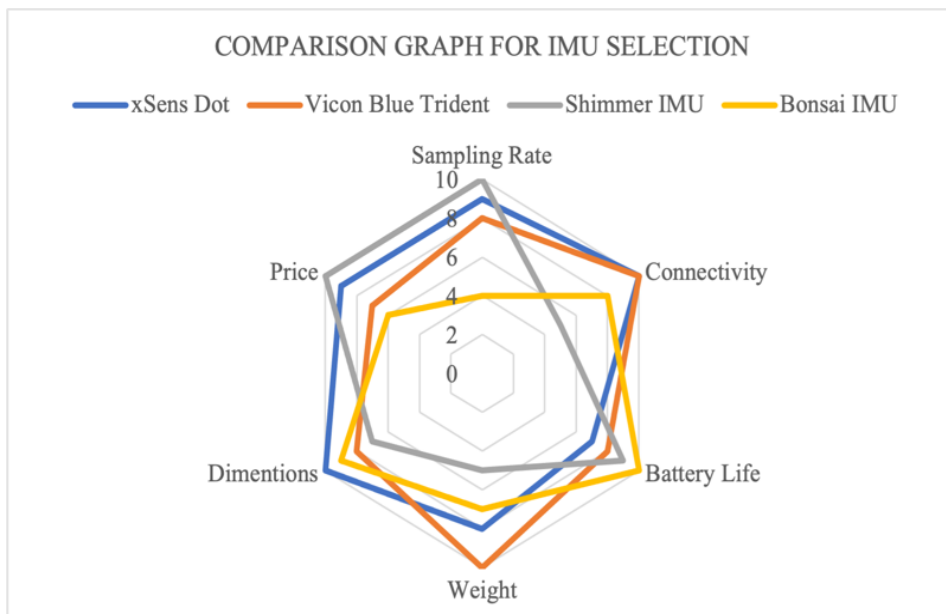


Figure 9: Comparison Graph for IMU selection [16]

Orientation data and free acceleration data are captured from Xsens Dot which utilises a built in sensor fusion algorithm called as XKFCore [92] and customised Kalman filter. Xsens Dot specification can be seen in Table 2 and in as previously mentioned covers all the requirements that were previously defined [16].

Table 2: Specification of Xsens Dot [92]

Physical Properties	Specifications
Weight	11.2g
Dimensions	36.3 × 30.4 × 10.8 mm (l × w × h)
Latency	30 ms
Battery	LIR2032H rechargeable coin battery
Communication Method	Bluetooth 5.0
Internal Storage	64 MB
Sample Rate	800 Hz
Output Rate	1Hz, 60Hz, and 120Hz
Electrical Current Consumption	68mA
Operating Temperature	0 to 50° Celsius
Water resistant rating	IP68
Communication platforms	- Android OS 8.0 or above - iOS 11.0 or above - Windows, MacOS, Raspberry Pi

3.2.2 Filtering and Sensor Fusion Techniques

Sensor fusion and filtering techniques need to be implemented when working with data obtained from inertial measurement units. The popular sensor fusion techniques for IMU data are Kalman filters, Complementary filters, and Particle filters. As previously mentioned, Xsens Dot utilises an internal Kalman filter where the filtering happens on the hardware which leads to a reduction in pre-processing time. Kalman filter plays an essential role in sensor fusion as stated by [96]. The Kalman filter was initially developed for navigation and control systems and its attributes address the filtering requirements for IMU data. The extended Kalman filter implemented in Xsens Dot addresses non-linearities by performing local linearization with the Taylor approximation of the non-linear model to work around this problem. This method is used to turn it into a linear model based on linearization points that need to be updated for each prediction of the recursive estimation. There are two main steps in successfully utilising Kalman filters which are prediction and correction [16].

The prediction step utilises control commands determine when the dynamic system will be in the next point in time. The correction step then utilises IMUs data to correct mistakes and predict error compared to the previous step. The prediction and correction steps which are known as recursive estimation, continuously repeat to provide accurate result [16] [96].

The limitation of Kalman filter is in its use in nonlinear values since the assumptions made during the filtering process would cause some issues. As stated in [97], most human movement is linear so this issue will not affect the use of Kalman filter in rehabilitation. In case of non-linear movement, extended Kalman filter solves the issue by performing a Taylor approximation of non-linear section of movement which creates locally linear sections in the data. [16] [96].

The Xsens Dot utilised a hardware motion processor capable of both sensor fusion and filtering, so additional filtering was not required to be implemented in the data processing pipeline of the framework. . However, to demonstrate the use of these filtering algorithms in clinical setting, sensor fusion of raw accelerometer and gyroscope data was implemented in the classification of movement associated with cerebral palsy. The details of this experiment can be found in Chapter 4 of this thesis.

3.2.3 Real time Joint Angle Measurement Techniques Utilising the Sensor Data

Three joint angle measurement techniques were tested to find the optimal approach to determine the wrist, knee, and elbow joint measurements given the constraint of maintaining clinically valid data. It is important to highlight that shoulder joint angles have been omitted in the calculations since these joint angles are complex and cannot be captured utilising only IMU data. IMUs can be utilised in calculating finger joints but due to the requirement of several IMUs per finger and the cumbersome nature of this approach, HGR was used as the method of measuring finger joints and detecting gestures as an example of a complex joint[16]. Hand gesture recognition methods also require a machine learning algorithm to be implemented which have been detailed in chapter 3.3.

To evaluate the validity of the selected methods, calculated angles were validated against goniometer measurement as goniometers are still the ones most commonly used in clinical settings and rehabilitation. These joint measurements also needed to be represented in real time so the user would be able to get feedback on the

accuracy of the exercising that are being performed. The sensor data would be streamed in real time via Bluetooth to a visualiser developed utilising the Unity game engine. Chapter 3.4 provides an in-depth overview of how sensor data is transferred to the human computer interface part of the framework. [16].

Method 1: Two quaternion angles provided by the IMU are utilised to create a rotate vector in reference to a unit vector on a given axis. This means each calculated measurement will provide the joint angle with reference to each axis. A dot product is then taken between the rotate vectors followed by an arc cosine to provide the joint angle which is later converted to degrees [16]. Figure 10 demonstrate a flow chart of method 1 using wrist rotation as an example.

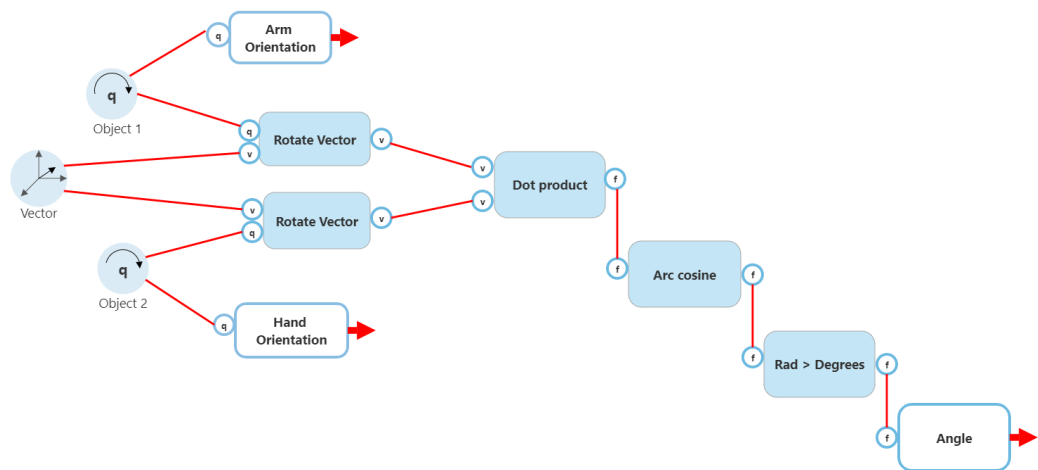


Figure 10: Method 1 for Joint Angle Measurement using IMUs [16].

Method 2: In this method, quaternions angles are captured from the IMU followed by measuring the difference between the quaternion of the game object and the child body part in the visualizer. To calculate this difference, the inverse quaternion of each parent and child object is calculated and multiplied together [16]. The new quaternion value is converted to Euler and presented to the user. Figure 11 demonstrate a flow chart of method 2 using wrist rotation as an example.

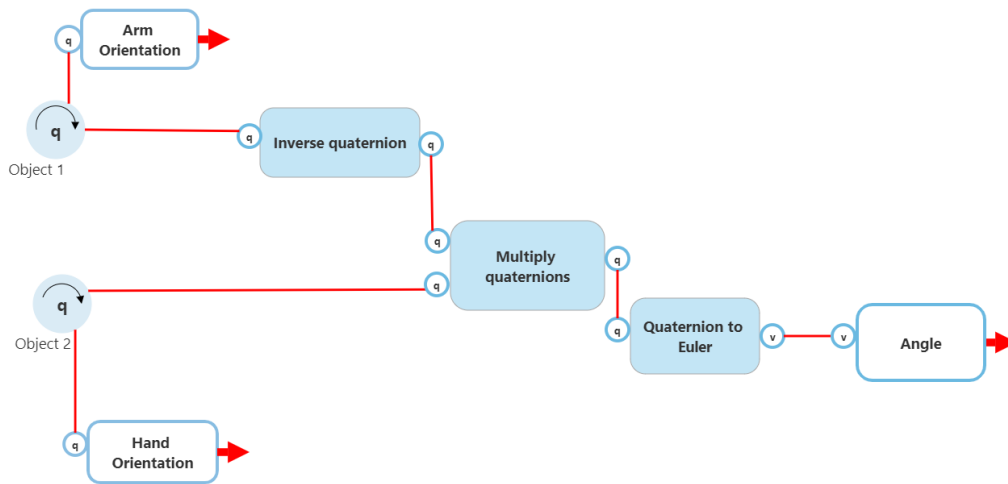


Figure 11: Method 2 for Joint Angle Measurement using IMUs [16].

Method 3: This method utilises one of the Unity game engine built in function “gameobject.localRotation.eulerAngles”. This function provides the angle of the child game object with reference to the object immediately attached to it known as the parent. This function will calculate the joint angle utilising quaternion data attached directly to the game objects [16]. Figure 12 demonstrate a flow chart of method 3 using wrist rotation as an example.

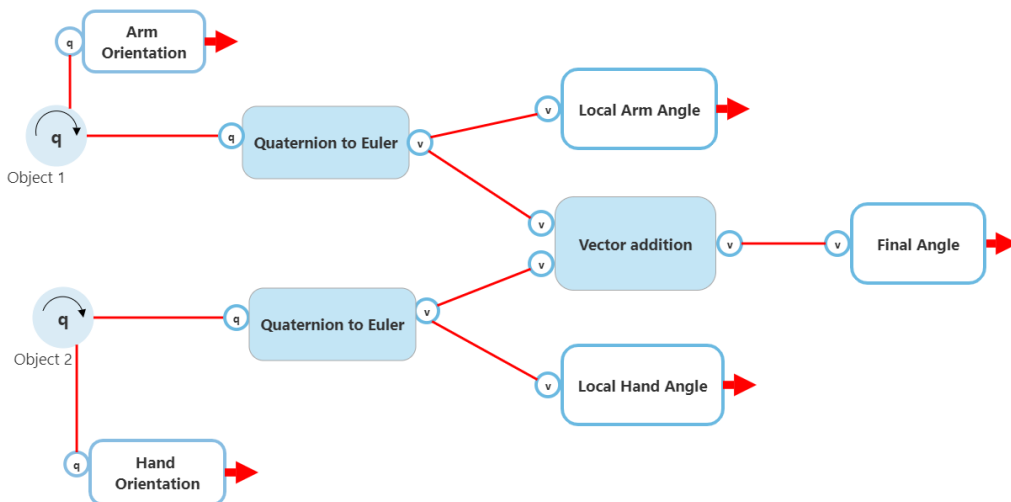


Figure 12: Method 3 for Joint Angle Measurement using IMUs [16].

These methods were validated against goniometers to measure the accuracy of the calculations. Additionally, the accuracy of the IMUs were validated in Curtin University’s Motion Analysis lab against Vicon motion capture technology [16]. The details of this validation results are available in Chapter 5.

3.2.4 Representation of Smoothness of Movement Utilising Sensor Data

As presented in section 2.3.2, data captured from IMUs can be utilised for measuring smoothness of human movement. To calculate smoothness measures using SPARC, DLJ, and LDLJ, the general python code provided by [40] have been modified. The code utilises function calls as wrappers, so the code segments can compute different requirements of the smoothness measures. This code is then applied to the data captured during an exercise session to provide a measurement of how smooth the exercises took place.

Implementation of SPARC Using Sensor Data

Figure 13 demonstrates the `sparc()` function which calculates the smoothness of a given speed profile using the modified spectral arc length metric. It takes as input the movement array, which represents the movement speed profile, the `fs` parameter indicating the sampling frequency of the data, and optional parameters such as `padlevel`, `fc`, and `amp_th`. The function returns the spectral arc length estimate of the smoothness of the movement, along with the frequency and magnitude spectrum of the movement data, as well as the selected portion of the spectrum used for calculating the spectral arc length. The function starts by determining the number of zeros to pad the movement data for estimating the spectral arc length. It then computes the frequency array and the normalized magnitude spectrum using the Fast Fourier Transform (FFT). Next, it applies low pass filtering to select the portion of the spectrum within the given cut-off frequency, `fc`. It further applies an amplitude threshold to determine the cut-off frequency up to which the spectral arc length is to be estimated. Finally, the function calculates the arc length by summing the differences between adjacent points in the selected frequency and magnitude arrays, considering the scaling factors.

The resulting arc length, representing the smoothness of the movement, is returned. If any error occurs during the computation, the function returns NaN (Not A Number) values for all output parameters. It is important to note that this function can only take a one-dimensional array and as such all spectral arc length values will be based on the magnitude of their respective types, gathered by computing the square root of the x, y, z values each being individually squared and added together.

```

def sparcc(movement, fs, padlevel=4, fc=10.0, amp_th=0.05):
    try:
        # Number of zeros to be padded.
        nfft = int(pow(2, np.ceil(np.log2(len(movement)))) + padlevel))

        # Frequency
        f = np.arange(0, fs, fs / nfft)
        # Normalized magnitude spectrum
        Mf = abs(np.fft.fft(movement, nfft))
        Mf = Mf / max(Mf)

        # Indices to choose only the spectrum within the given cut off
        # frequency Fc.
        # NOTE: This is a low pass filtering operation to get rid of high
        # frequency noise from affecting the next step (amplitude threshold
        # based cut off for arc length calculation).
        fc_inx = ((f <= fc) * 1).nonzero()
        f_sel = f[fc_inx]
        Mf_sel = Mf[fc_inx]

        # Choose the amplitude threshold based cut off frequency.
        # Index of the last point on the magnitude spectrum that is greater
        # than or equal to the amplitude threshold.
        inx = ((Mf_sel >= amp_th) * 1).nonzero()[0]
        fc_inx = range(inx[0], inx[-1] + 1)
        f_sel = f_sel[fc_inx]
        Mf_sel = Mf_sel[fc_inx]

        # Calculate arc length
        new_sal = -sum(
            np.sqrt(pow(np.diff(f_sel) / (f_sel[-1] - f_sel[0]), 2) +
                    pow(np.diff(Mf_sel), 2)))
        return new_sal, (f, Mf), (f_sel, Mf_sel)
    except:
        return np.NaN, np.NaN, np.NaN

```

Figure 13: Main Code for Implementing SPARC

Implementation of Dimensionless Jerk Using Sensor Data

Figure 14 shows the main function for calculating individual factors of the dimensionless jerk metric for a given movement profile. It takes as input the movement array, containing the velocity, acceleration, or jerk profile. The fs parameter indicates the sampling frequency of the data. The data_type parameter specifies the type of movement data provided (either 'vel' for velocity or 'accl' for acceleration). The rem_mean parameter is only applicable for acceleration or jerk data, indicates whether the mean of the movement data should be removed before computing the jerk. The function returns three factors: T^N representing the duration scaling factor, A^M representing the amplitude scaling factor, and J representing the jerk cost. The function first defines parameters for different data types, specifying the number of different dimensions and the scaling factors. It checks if the data_type parameter is valid and raises an exception if not. The input movement is converted into a NumPy array, and

the dimensions are checked to ensure that the data has at least three samples, otherwise the dataset is invalid. The time interval `dt` is calculated based on the sampling frequency. If the `data_type` is acceleration and `rem_mean` is `True`, the mean of the movement data is subtracted from the array. This step removes the mean acceleration if specified. Next, the jerk is computed by taking the norm of the difference of consecutive rows in the movement array (`n` times). The jerk is then divided by the appropriate power of the time interval `dt` to obtain the modified jerk. The modified jerk is the squared sum of the jerk values multiplied by `dt`. The factors `mdur`, `mamp`, and `mjerk` are calculated as the duration, amplitude, and jerk components of the dimensionless jerk metric, respectively. Finally, these factors are returned by the function.

```
def dimensionless_jerk_factors(movement, fs, data_type='vel', rem_mean=False):
    # Parameter definition for different data types
    param = {'vel': {'n': 2, 'N': 3},
            'accl': {'n': 1, 'N': 1}}

    n, N = (param[data_type]['n'], param[data_type]['N'])

    # make sure data_type makes sense.
    if data_type not in ('vel', 'accl'):
        _str = '\n'.join(("data_type has to be ('vel', 'accl')!",
                        "{0} provided is not valid".format(data_type)))
        raise Exception(_str)
    return

    # first enforce data into an numpy array.
    movement = np.array(movement)
    r, c = np.shape(movement)
    if r < 3:
        _str = '\n'.join(
            ("Data is too short to calculate jerk! Data must",
            "have at least 3 samples ({0} given)".format(r)))
        raise Exception(_str)
    return

    dt = 1. / fs

    # Remove the mean if the movement dat is acceleration?
    if data_type == 'accl' and rem_mean == True:
        movement = movement - np.mean(movement, axis=0)

    # jerk
    jerk = np.linalg.norm(np.diff(movement, axis=0, n=n), axis=1)
    jerk /= np.power(dt, n)
    mjerk = np.sum(np.power(jerk, 2)) * dt

    # time.
    _N = len(movement)
    mdur = np.power(_N * dt, N)

    # amplitude.
    mamp = np.power(np.max(np.linalg.norm(movement, axis=1)), 2)

    # dlj factors
    return mdur, mamp, mjerk
```

Figure 14: Main Code for Implementing DLJ

Implementation of Log Dimensionless Jerk Using Sensor Data

Figure 15 shows the main code for Log dimensionless jerk which is a basic translation of the dimensionless jerk value, as described previously. The code takes in the same factors, feeds them back into the dimensionless jerk function and then modifies the result before returning the value.

```
dljfac = dimensionless_jerk_factors(movement, fs, data_type, rem_mean)
return - np.log(dljfac[0]), np.log(dljfac[1]), - np.log(dljfac[2])
```

Figure 15: Main Code for Implementing LDLJ

Figure 16 lists the `log_dimensionless_jerk_imu_factors()` function which calculates the individual factors of the log dimensionless jerk metric used for IMU (Inertial Measurement Unit) data analysis. It takes several input parameters: `accls` representing the accelerometer profile, `gyros` representing the gyroscope profile, `grav` representing the gravity vector, and `fs` representing the sampling frequency of the data. The gravity vector can be represented by a simple 2d array with the values `[0, 0, -9.81]`, which will be present in the main body.

The function returns three factors: $-\ln(T)$ representing the duration scaling factor, $+\ln(A)$ representing the amplitude scaling factor, and $-\ln(J)$ representing the jerk cost. First, the function computes the movement duration (`mdur`) by multiplying the number of samples (`_N`) with the sampling interval (`dt`). Next, it calculates the gravity-subtracted mean square amplitude (`mamp`) by taking the norm of the accelerometer data and dividing it by the number of samples (`_N`). If gyroscope data (`gyros`) is provided, the square of the norm of the gravity vector (`grav`) is subtracted from `mamp`. Then, the function calculates the derivative of the accelerometer signal (`_daccls`) by taking the differences between consecutive accelerometer measurements and scaling it by the sampling frequency (`fs`).

If gyroscope data is available, the function computes the cross product (`_awcross`) of each accelerometer measurement and its corresponding gyroscope measurement using a loop. Otherwise, `_awcross` is set to an array of zeros with the same shape as `_daccls`. Next, the corrected jerk (`_jsc`) is computed by subtracting `_awcross` from `_daccls`. The norm of `_jsc` is then taken, and its squares are summed over the time axis (`axis=0`). The resulting value is multiplied by the sampling interval (`dt`) to obtain the jerk cost (`m jerk`). Finally, the function returns the negative natural

logarithm of the duration scaling factor ($-\ln(T)$), the natural logarithm of the amplitude scaling factor ($+\ln(A)$), and the negative natural logarithm of the jerk cost ($-\ln(J)$).

```
def log_dimensionless_jerk_imu_factors(accls, gyros, grav, fs):
    # Sample time
    dt = 1. / fs
    _N = len(accls)

    # Movement duration.
    mdur = _N * dt

    # Gravity subtracted mean square amplitude
    mamp = np.power(np.linalg.norm(accls), 2) / _N
    if gyros is not None:
        mamp = mamp - np.power(np.linalg.norm(grav), 2)

    # Derivative of the accelerometer signal
    _daccls = np.vstack((np.zeros((1, 3)), np.diff(accls, axis=0) * fs)).T

    # Get corrected jerk if gyroscope data is available.
    if gyros is not None:
        _awcross = np.array([np.cross(_as, _ws)
                             for _as, _ws in zip(accls, gyros)]).T
    else:
        _awcross = np.zeros(np.shape(_daccls))

    # Corrected jerk
    _jsc = _daccls - _awcross
    mjerk = np.sum(np.power(np.linalg.norm(_jsc, axis=0), 2)) * dt

    return - np.log(mdur), np.log(mamp), - np.log(mjerk)
```

Figure 16: The Main Code for Implementing LDLJ using IMU Data

The functions and code explained in this sub chapter went through a validation process, the result of which is presented in Chapter 5.

3.3 DEFINING THE MACHINE LEARNING REQUIREMENTS FOR THE FRAMEWORK

3.3.1 Use of Optical Motion Capture Technologies for The Framework

The developed framework has been designed to human movement to be monitored whilst engaging in rehabilitation exercises. The developed framework uses IMU sensors as the principal method of capturing human joint movements (wrist, elbow, and Knee; however, it also utilises optical motion capture technology and machine learning for capturing finger movement for modelling a complex joint. As there are several joints within a human hand, utilising several IMUs would not be feasible and that is the justification for utilising the gesture recognition and machine learning approaches that have been used chosen [43].

An appropriate algorithm needed to be developed, validated, and implemented that facilitates hand gesture required for capturing finger movements required for the framework. HGR requires a human control interface component and a command mapping component, so it can be utilised as an alternative control method of interacting with visual elements within the gamified exercises. As outlined in Chapter 2, HGR algorithms consist of a gesture description method, data acquisition method, gesture identifying method, and a gesture classifying component. To cover all these requirements, a six-stage approach was defined which will be covered in the following sub chapters [43]. Figure 17 provides a high-level overview of these six stages.

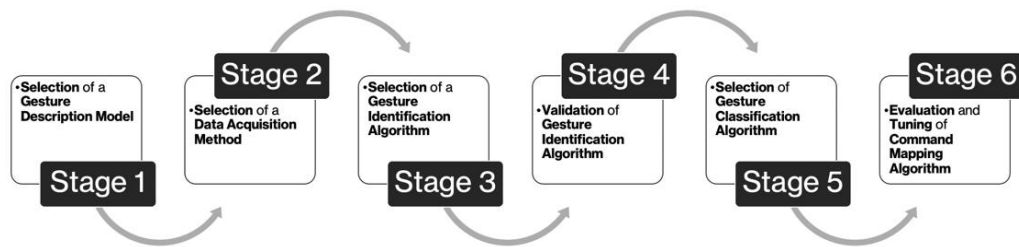


Figure 17: Overview of 6 stages for implementing the HGR [43]

3.3.2 Defining Simplifications and Governing Criteria of the HGR

There are several unique approaches for different subcomponents of HGR algorithms. The following simplifications were considered to help with selection of relevant HGR components [43].

- 1) There cannot be any changes to components that have been selected before each stage of investigation. This meant that selected component would not change to accommodate for needs of new developed approaches. For example, data acquisition techniques would not change to the HGR requirements of the following stages.
- 2) Each component was selected at its own stage without focusing on its effect on future stages. This selection took place in accordance with the applicable governing criteria, successful selection of the previous component, and relevant validation results.

- 3) Gesture description and data acquisition component were selected based on the outcome of the literature review without any new quantitative or qualitative analysis.

These simplifications provide a linear investigation structure which means the number of applicable implementations were exponentially reduced for each stage of the HGR implementation. This simplification also removed the requirement of time-consumed in the experimental analysis for the first two investigation stages. This led to a reduction of the workload for analysing different implementation and allowed for investigating a wider range of techniques. The validity of the overall investigation was not compromised since these components cannot be defined by metrics that can be analysed. This means, they needed to be derived from attributed to the result of the literature review [43].

A set of governing criteria were defined to make sure a cohesive and effective solution has been developed. These criteria augment the dependent and independent variables in each investigation stage such that an effective HGR can be implemented [43]. The governing criteria was defined as seen below:

- 1) Reliability of the commands: This refers to both number of unit commands issues by the algorithm and the ability to distinguish between unique commands.
- 2) Reproducibility of the commands: This refers to the ability to robustly produce the same action given the same user input.
- 3) Being physically non-restrictive by equipment: This refers to any physical restriction the equipment provides to the user's body as well as any imposed environment requirements.
- 4) Ease of operation: This criterion refers to the complexity in learning the interface for the HGR and how quickly a new user would be able to start interacting with the system.
- 5) Low computational requirements: This criterion was defined to make sure the HGR could run on readily available devices like low-cost smart phones while keeping an optimal operational speed.

- 6) Low monetary cost to the user: low monetary investment. Minimise economic burden on clinician and patient.

3.3.3 Stage One: Selection of Gesture-Description Model

The first stage of the HGR selections was to choose a gesture description model according to the defined governing criteria. As previously mentioned, gesture description includes a gesture type, gesture information, and a gesture model. Based on the simplification criteria, single hand static gesture was chosen for the gesture type, symbolic information was chosen for the gesture information, and a three-dimensional, 21 landmark skeleton model was selected as the gesture model [43].

Gesture-Type Selection Scope and Justification

Three main considerations existed in the gesture type that needed to be analysed. The first was the motion of observed gestures, which could be static or dynamic gesture sets. The second was the inclusivity of wrist motion which defined the scope of observation. Thirdly, the number of observed gestures were also considered. Static single-hand gestures were chosen to ensure future components are simple and easy to understand for the user. Single hand static gestures also helped with lowering the computational requirement for the detection algorithm [43].

A wide range of gestures were developed to avoid biases in the classification and identification algorithms. To help with the versatility of the detection algorithm, a requirement was set to ensure the gesture set contains gestures defined by recognised sign-language systems [43]. An example of the sign language gesture set used in the detection algorithm can be seen in Figure 18.

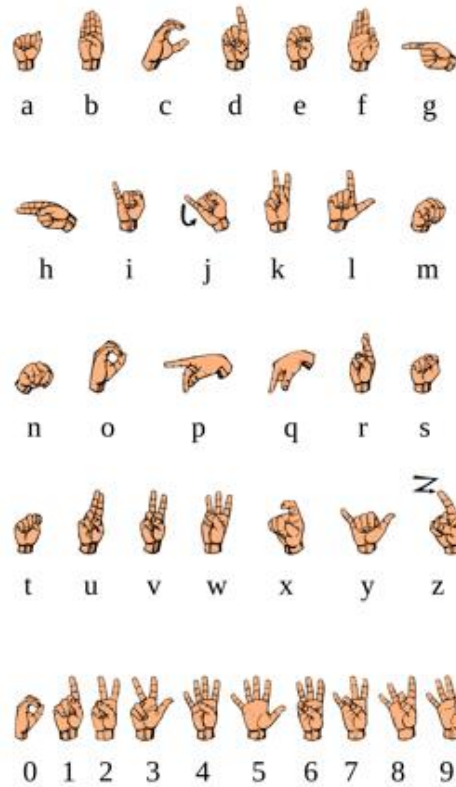


Figure 18: Example of sign language used for the HGR [43]

Gesture Model Selection Scope and Justification

The computational requirements to generate each model and the number of classifiable landmarks affect the gesture model selection. To balance these factors, a modelling method was selected that allowed for the required classifiable landmarks depending on the gesture set. Based on the models reviewed in Chapter 2, a 3D skeleton model was selected since appearance-based models had low number of classifiable landmarks. 3D geometric and 3D texture volumetric models were not selected either due to their complexity leading to high computational requirements [43].

Gesture Information Scope and Justification

Four considerations had to be made regarding gesture information which were being spatial, pathic, symbolic, or affective. The selection of the source of this information was not mutually exclusive, meaning one or all could be selected. Symbolic information was chosen as the primary source of information based on the gesture type selection mentioned above. Spatial information for the three-dimensional

skeleton landmark model was used to calculate joint angles for the 15 observed joints [43]. Details of the joint angle calculations are provided in section 3.3.7.

3.3.4 Stage Two: Selection of Data-Acquisition Method

Stage two was to select a data acquisition approach for the HGR capable of efficiently and non-restrictively observe a human hand based on the selected gesture models. The result of the literature review and governing criteria 1,2,3,5, and 6 were utilised for the selection process by evaluating each solution against the given criteria. The selection led to choosing single RGB cameras as the main data acquisition method for the HGR as it satisfied applicable criteria. Depth cameras did not meet Criterion 3 requirement as the range restrictions could restrict users. Stereo cameras were omitted to their high computational requirements and focal point restrictions which did not satisfy Criterion 3 and 5. Band and glove methods were also omitted as they did not meet Criterion 3 [43].

There are some drawbacks to use of single RGB cameras that were addressed by other components of the HGR. These drawbacks are robustness issues associated with background and operator hand, variability, and single viewpoint errors such as self-occlusions and transform inconsistency. However, this form of HGR is very mature that address most of these issues and the technology is readily accessible to end use via smart phones, tablets, and laptops [43].

3.3.5 Stage Three: Selection of Gesture-Identification Algorithm

Stage 3 was to select a gesture identification algorithm to extract hand features from the data captured from a single RGB camera. The extracted features needed to be in form of a three-dimensional skeleton model following Governing Criteria 1,2, and 5. The selected method also needed to address drawbacks of single RGB camera. To find the best applicable gesture identification algorithm, a qualitative analysis was conducted that focused on computational requirement and the observable localization accuracy of the selected algorithms [43].

There are several feature-extraction methods that are applicable for HGR gesture identification, since it was not possible to directly test all of them directly, the design simplifications mentioned in stage one was used to reduce the scope. Selection of gesture-description and data-acquisition component reduced the scope by the removal of any methods not developed for three-dimensional skeletal models; the removal of

any method not compatible with single RGB cameras; and considering only existing open-source implementations. This led the selection to MediaPipe hands, InterHands2.6M, and OpenCV as the possible approaches as discussed in Chapter 2, each with their own pre-processing and feature extraction techniques leading to different computational requirements. The localisation accuracy of the skeleton model and computational requirements for feature extractions were the primary focus on selecting the gesture identification approach [43]. The following tests were designs to address each of the mentioned requirements:

1. Identifier Implementation: This test was design to implement a baseline variant of the three algorithms capable of observing a single human hand, calculating the angle of its 15 primary joints while displaying the 3D-skeleton model. This stage assesses the operational readiness and the computational requirements of implementing each algorithm. It also acquires an operational version of the algorithms where future testing would be performed. All the algorithms were initially tested on a 2017 Mac Book Pro with a 3.5GHz Dual Core Intel Core i7 CPU, and Intel Iris Plus Graphics card. The system also had a 16 HB LPDDR3 RAM and 256 GB of storage.
2. Qualitative Analysis: This test observed was conducted to observe the localisation accuracy of the algorithms to make sure Governing Criteria 1 and 2 are addressed. This approach reduced the single camera data-acquisition method. Accuracy of the algorithms were observed for self-occlusion, rotation, and translation using operational version of the first sub test. To achieve this, a user's hand was held in a constant position in front of the camera and the displayed 3D model was then recorded. The hand was then rotated and translated around the camera's view point while observing the display model to make sure localisation accuracy is being maintained despite the movement. The final stage was to turn the hand so some parts are occluded from the point of view of the camera to determine if hand features could still be produced.

3.3.6 Stage Four: Validation of Selected Gesture-Identification Algorithms

Stage four was to validate the gesture identification algorithm based on the accuracy and robustness of the produced model. This stage utilised a quantitative

approach based on the advice received by clinical collaborators of this research by comparing joint angles calculated from the models to the joint angles measured by a finger goniometer. The result of this method showed the if a selected algorithm met Governing Criterion 2. This method was applied to the final selected identification algorithm. As mentioned in stage 1, a 3D skeleton model was selected as the gesture description model. Percentage variances between joint angles of the observed hand measured by the goniometer and the ones measured by the model were calculated to determine the accuracy of the model. As discussed in Chapter 2 goniometers are the preferred clinical method for capturing human joint and are used at the beginning of each rehabilitation exercise session to for understanding the base line of participants. Due to their widespread use in clinical settings, goniometers were used to evaluate the accuracy of the calculated angles. The measured joint angles were the metacarpophalangeal, proximal interphalangeal, and the distal interphalangeal joints of all fingers. The static arm of the finger goniometer was stabilized against the proximal side of the joint and the hinge of the goniometer was place directly above the observed joint. In the event of a bulbous knuckle, the goniometer was moved to the side of the finger so that the hinge sat directly in front of the observed joint. After securing the position, the free arm of the goniometer would be lightly pressed against the distal side of the observed joint. This measurement was done without any force applied as it would alter the pose of the observed hand. After the free arm of the goniometer was in contact with the distal side of the joint, the joint angle was recorded to the nearest 5° [43].

Joint Angle Calculation Methods Using MediaPipe

Two methods were applied for joint angle calculation derived from the 3D model. The first method was vector angle calculation where two vectors are created: each traveling to the joints associated with the current measurement. After defining the vectors, their dot product was calculated to measure angles between two joins. This method is a similar implementation to what was previously discussed in chapter 3.2.3. Another method for joint angle measurement was to discard the depth component of the model and perform the same calculation as previous method. This method was implemented to measure how much of an effect the 3D nature of the model had on performance [43]. Figure 19 and Figure 20 show the python code used for implementing the calculations.

```

def calcJointAngles_3D(landmarkArray):

    jointAngles = [0]*15
    jointAngleIndex = 0

    for i in range(5):

        for j in range(3):

            pInit = landmarkArray.landmark[0]
            #print(landmarkArray.landmark[0])
            if (j != 0 ):
                pInit = landmarkArray.landmark[i*4+j]
            pMid = landmarkArray.landmark[i*4 + j + 1]
            pFinal = landmarkArray.landmark[i*4 + j + 2]

            v1 = [pMid.x - pInit.x, pMid.y - pInit.y, pMid.z - pInit.z]
            v2 = [pFinal.x - pMid.x, pFinal.y - pMid.y, pFinal.z - pMid.z]

            theta_num = v1[0]*v2[0] + v1[1]*v2[1] + v1[2]*v2[2]
            theta_dom = calcMagnitude(v1) * calcMagnitude(v2)

            theta = math.acos(theta_num / theta_dom)

            jointAngle = (math.pi - theta) * 180/math.pi
            jointAngles[jointAngleIndex] = jointAngle
            jointAngleIndex += 1

    return jointAngles

```

Figure 19: Python code for calculating three-dimensional join angles [43].

```

def calcJointAngles_2D(landmarkArray):

    jointAngles = [0]*15
    jointAngleIndex = 0

    for i in range(5):

        for j in range(3):

            pInit = landmarkArray.landmark[0]
            #print(landmarkArray.landmark[0])
            if (j != 0 ):
                pInit = landmarkArray.landmark[i*4+j]
            pMid = landmarkArray.landmark[i*4 + j + 1]
            pFinal = landmarkArray.landmark[i*4 + j + 2]

            a = np.array([pInit.x, pInit.y]) # First coord
            b = np.array([pMid.x, pMid.y]) # Second coord
            c = np.array([pFinal.x, pFinal.y]) # Third coord

            radians = np.arctan2(c[1] - b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
            angle = np.abs(radians*180.0/np.pi)

            if angle > 180.0:
                angle = 360-angle

            jointAngles[jointAngleIndex] = round(angle)
            jointAngleIndex += 1

    return jointAngles

```

Figure 20: Python code for calculating two-dimensional join angles [43]

To test the robustness and accuracy of the algorithm, two independent variables, hand pose and hand orientation to camera, were modified through the following steps [43]:

1. Hand pose: Three positions were considered: Fully closed position, partially closed position, and fully open position. These positions were

selected as they are easy to hold, stable, repeatable, and were suggested by clinical collaborators of this research. These positions simulate a full range of motion of a human hand and the example of these positions can be seen in Table 3 and Table 4.

Table 3: Viewpoint and poser examples for stage four for HGR [43]














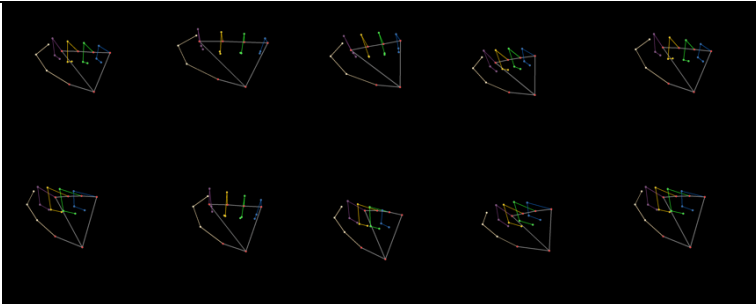

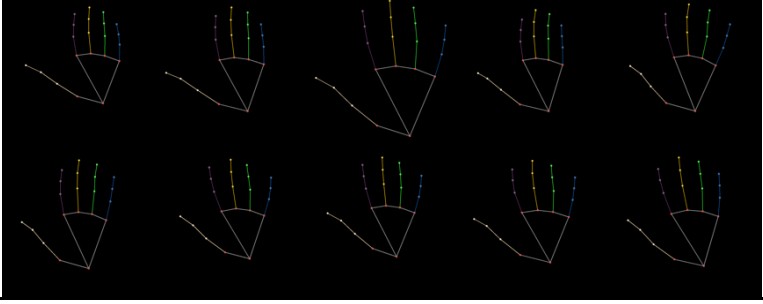
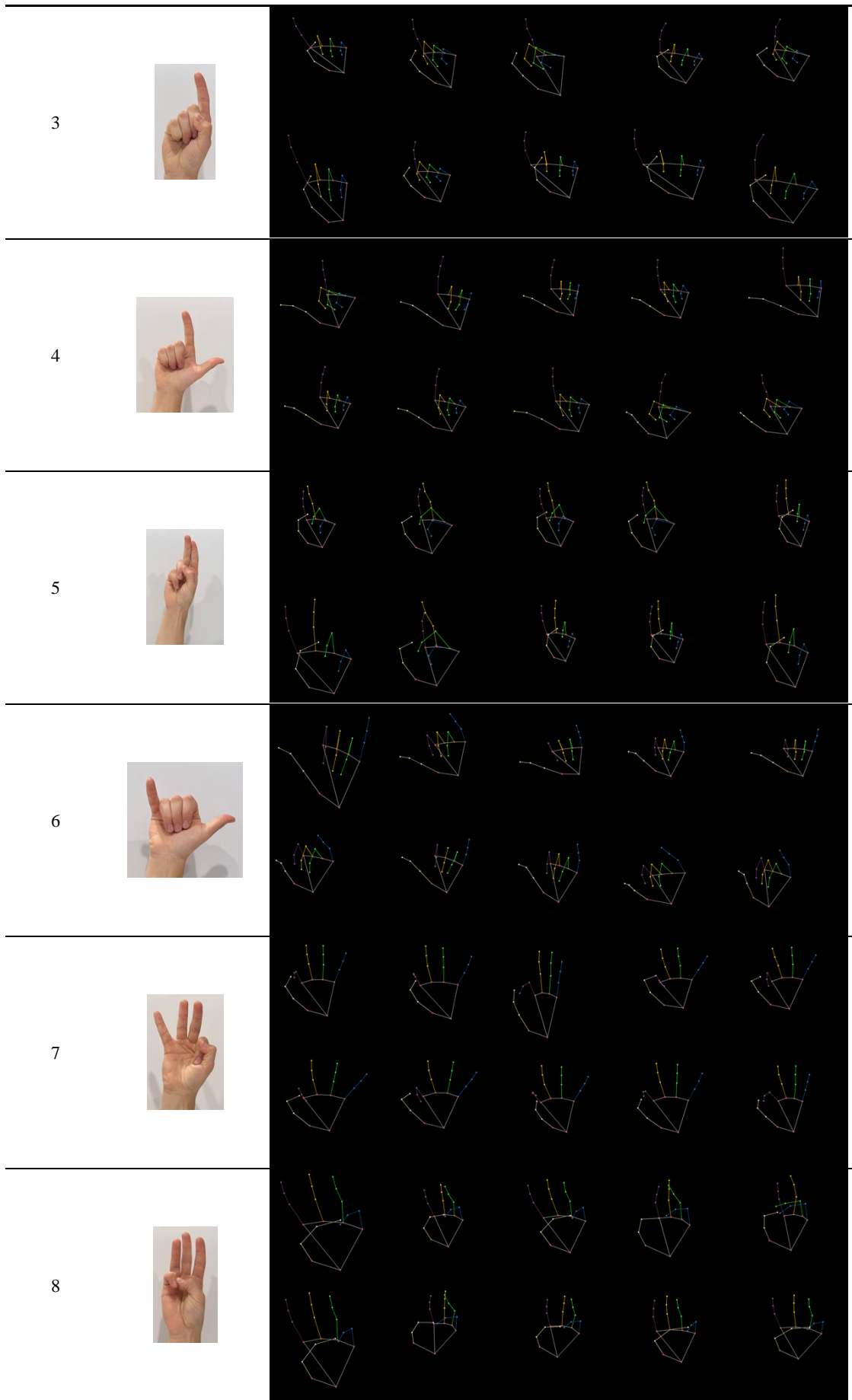
Pose	Viewpoint Offset			
	0° (Font)	45° (Forty)	90° (Side)	180° (Back)
Open				
Partial				
Closed				

Table 4: Dataset used for classification and analysis [43]

Gesture Identifier	Gesture Reference Image	Models
1		
2		



2. Hand orientation with respect to camera: The second independent variable was the incident angle of the camera's POV and the observed hand. The orientation angle would change so the robustness of the algorithm can be measure against rotation and self-occlusions. Four photos were taken for each of the hand poses above which included: the front of the hand, 45° offset, 90° offset, and a 180° offset as seen in Table 3 and Table 4.

Maintaining high accuracy in MediaPipe Measurements via Control Inputs

Sufficient accuracy of the MediaPipe measurements was defined as being within goniometer error ranges which was maintained using the following control inputs while keeping the results repeatable.

1. Lighting: Defined to avoid variance in lighting so all tests were conducted in a well illuminated environment without any shadowing on the observed hand. This was maintained by conducting the evaluation in the same lighting conditions.
2. Background: Considered to lower the impact on the MPH modelling process. White backgrounds were used to maintain a high level of contrast between the hand and the surrounding environment and help with feature extraction.
3. Body position: defined to ensure viewpoint angles and positions for each participant is the same. The control variables were for the participants to kneel in a comfortable position, with their forearm braced against a test bench. The test bench included a set of markers for appropriate positions for each background and participant.
4. Euler angles of the camera: Defined so the viewpoint orientation was the same for all participants and varied by desired amounts between tests. This practice was derived from research done in [98].
5. General hand size and distance from camera: A fixed camera distant was used for each participant to limit the variability to the participant hands. This was achieved by a fixed camera mounting location for each captured pose.

Final Testing Procedures for Stage Four of HGR

This stage required the following five steps to be able to test the final components required for stage four of the HGR.

1. Establishing the previously method controls
2. Forming the required hand pose by the participant
3. Recording the joint angle utilising a goniometer
4. Taking photos of the hand from the required viewpoint
5. Measuring the joint again via a goniometer

The sets of joint angle measures mentioned above were then compared and if they did not match, the test image would be discarded, and the process would be repeated. The aim of this approach was to ensure the participants hand pose had remained stable during the test. The joint angle calculation and gesture-identification algorithms would then utilise the valid photos to generate a set of observed joint angles. The python code for this stage was previously provided in Figure 19 and Figure 20. The final set of accuracies were then generated by comparing the results against the goniometer. The outcome of these validation tests is provided in Chapter 5.

3.3.7 Stage Five: Selection of Gesture-Classification Algorithm and Finger Joint Measurements

Stage five utilised quantitative methods to select the gesture-classification algorithm that best complemented the gesture-identification method. The Governing Criteria 1,2, and 5 effected this stage from which two quantitative metrics were calculated to make the final decision. These metrics were the classification accuracy of the algorithms in form of a confusion matrix and classification speed to evaluate the computational requirements of the algorithms. [43]. After applying the simplification methods described in section 3.3.2, the following two possible scenarios were defined:

1. If stage 4 demonstrated that the selected gesture-identification algorithm can produce the model accurately and robustly a low dimensionality classifier base on 15 single dimension joint angles would be utilised. This led to investigating decision trees, KNNs, and linear regression algorithms.

2. If the stage 4 lead to a non-ideal model accuracy, a higher input dimensionality classifier with 21 three-dimensional coordinate system (total of 63 dimensions) would be utilised. This led to investigation of ANNs, SVM, linear regression, and non-machine learning bounds-based approaches.

Dependent variables for stage five remained as classification accuracy and time. The independent factors were the style and implementation of the classifiers [43]. The following classifiers were kept as constant to insure a fair investigation:

1. Test data set: A common set of test gestures were identified between algorithm which included 10 gestures. Then images were then created for each gesture and converted to the 3D model using MPH. This resulted in 100 models that formed the test data. Criteria 1 and 2 were considered for this selection where the hands varied in the ten selected images in scale, orientation, and pose. This avoid testing the accuracy algorithm only on “best case scenarios”.
2. Computational requirements for each algorithm: The hardware specification of used for each algorithm kept the same to ensure it would not affect the selection process. During the classification time testing, the time taken for each algorithm was only used at the prediction stage of the classifier. This excluded the time required for initializing and training the classifier, loading the MPH model, and any other time associated with generation of confusion matrices.

The procedure for stage five utilised a basic algorithm to sequentially test each of the prospective algorithm against the common data set. After each test, the predictions of the classifiers were recorded in their respective confusion matrices followed by recording the time taken to perform the classification in a CSV file. The final confusion matrices were developed after all test images have been tested, and the time performance data was recorded for evaluation [43].

3.3.8 Stage Six: Gesture Mapping and Tuning

Stage six involved developing the gesture mapping component capable of translating classifiable gesture into exercise activities. As an example, for the use case, a physical quad-rotor drone was utilised, and the exercises would fly the drone in different directions. This use case is further explained in Chapter 4. One to one command mapping was utilised as the mapping method to align with the previously mentioned design decisions. This served as the initial solution that demonstrated the HCI algorithm as an alternative means of control. The initial commands were further tuned to allow smooth interactions with the game objects and the physical drone following governing criteria 4 [43].

3.4 DESIGN AND IMPLEMENTATION OF THE HUMAN COMPUTER INTERFACE

3.4.1 Use of Unity Game Engine for Visualization

The Unity game engine was selected as the main component for containing the games designed for rehabilitation exercises. Unity is a real-time development platform game engine that was developed by the Unity Technologies and released in 2005. The primary language used for the developed games is C sharp (C#) and editing of the source code is performed through Microsoft Visual Studio. The Unity Game engine has the ability to be supported by multiple platforms such as a PC or smart Phones, making it an ideal platform for game development and deployment as they can be accessed via common consumer devices.

The Unity game engine lets the developer design both 2D and 3D games with applied physics, graphics, and visual effects. It also provides a range of free assets and tutorials through its website from other developers, providing a wide range of guides for developers to apply in their games. Additionally, 3D models developed in blender which is a free and open-source 3D creation suit, can be added to the unity application to make unique game objects [99]. Furthermore, the unity engine has a “interaction Engine” which allows for the user to interact through their physical movement with game objects and interfaces. Unity provided a solid foundation for creating immersive and interactive gamified experiences for the framework.

3.4.2 Connectivity of Sensor and ML Elements to Unity

Unity contained the capability for custom plugins which helped in incorporating IMU data streaming and HGR capability for each of the interactive experiences. Figure 21 demonstrates the high-level connectivity of the main visual elements of the framework which consisted of the following structure:

1. Data extraction from the IMU and vision-based software.
2. Transmission of extracted data to the unity platform
3. Creation of a main visualisation interface in unity than can facilitate different exercises.
4. Design of sample mini game with the capability to modify them to fit different exercise.
5. Live recording exercise session with the aim of report orientation data such as active and passive range of movement, smoothness measures, and statistics regarding the engagement with the exercises.

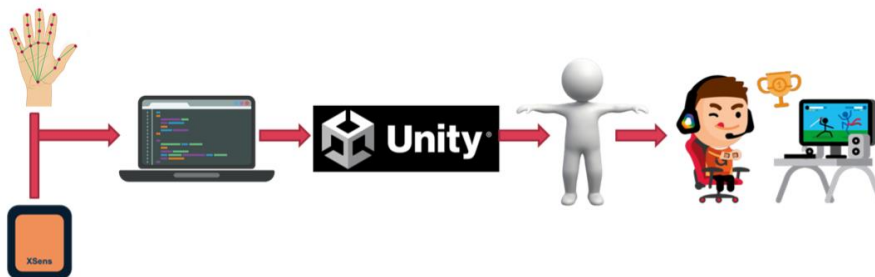


Figure 21: High-level overview of the data stream elements of the framework.

IMU Data Streaming to Unity

The primary source for creating the data stream from the IMUs to unity was the Xsens Dot software development kit (SDK) provided by manufacturers of the sensors. The implemented SDK provides a dependable data stream solution for transmitting the IMU data directly to Unity. To facilitate the transmission, the Xsens Dot PC SDK was also utilised which is provided as an application programming interface (API) which integrates data captured from the IMU into windows-based applications. The SDK provides libraries for interfacing with the sensors, sample codes, and through documentation for implementation. The code provided in Figure 22 Fig demonstrates

the method for real time streaming of Euler angles, device's tag name, address, battery, and connection status to the windows-based computer.

```
Press any key or wait 20 seconds to stop scanning...
Number of connected DOTs: 1. Press any key to start.
Stopped scanning for devices.
Opening DOT with address: @ D4:22:CD:00:2A:67
Found a device with Tag: Qing2 @ address: D4:22:CD:00:2A:67
Available filter profiles:
General
Dynamic
Current profile: General
Successfully set profile to General
Putting device into measurement mode.
Qing2
Roll: 0.39, Pitch: 1.52, Yaw: 68.86|
Resetting heading for device D4:22:CD:00:2A:67: OK
Qing2 BatteryLevel: 51 Charging status: 0
```

Figure 22: Sample code for streaming sensor data to a windows-based computer

Machine Learning Data Streaming to Unity

As previously mentioned, MediaPipe hand was utilised as the main HGR for the framework which meant the data captured from MediaPipe needed to be transferred to Unity. To facilitate the transfer, the Python code in Figure 23 was first created as the first step of implementation.

```
22 def draw_finger_angles(image, results, joint_list):
23     store_angle = []
24     # Loop through hands
25     for hand in results.multi_hand_landmarks:
26         #Loop through joint sets
27         for joint in joint_list:
28             a = np.array([hand.landmark[joint[0]].x, hand.landmark[joint[0]].y]) # First coord
29             b = np.array([hand.landmark[joint[1]].x, hand.landmark[joint[1]].y]) # Second coord
30             c = np.array([hand.landmark[joint[2]].x, hand.landmark[joint[2]].y]) # Third coord
31
32             radians = np.arctan2(c[1] - b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
33             angle = np.abs(radians*180.0/np.pi)
34
35             if angle > 180.0:
36                 angle = 360-angle
37
38             store_angle.append(angle)
39
40
41             cv2.putText(image, str(round(angle, 2)), tuple(np.multiply(b, [640, 480]).astype(int)),
42                 cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2, cv2.LINE_AA)
43
44     return image
```

Figure 23: Python code for implementing MediaPipe Hand for the data stream.

Aliases were created after importing the MediaPipe library to facilitate access to the drawing utilities and hand tracking modules within MediaPipe. Figure 24 was then utilised as a reference for the hand land marks so an array was created called “jont_list”. This was followed by implementing an iterative process in the

“draw_finger_angles” function to ensure each detected hand in the “results” is traversed, and the joint set in the “joint_list” can be sequentially examined. The angles are then calculated as previously discussed in stage four of the HGR.

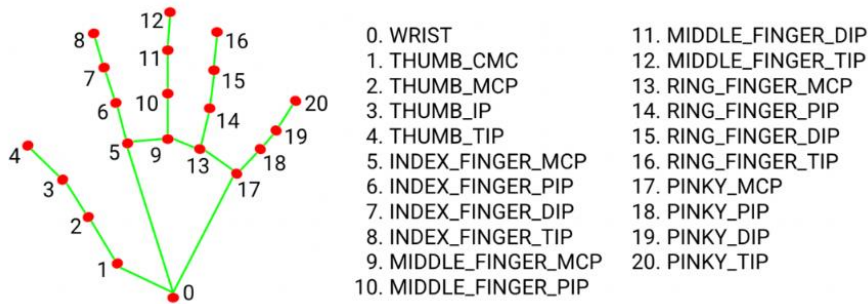


Figure 24: Localisation of 21 hand landmarks within MediaPipe [100]

Figure 25 provides a look at the “get_label” function that processes the hand classification results, extracts the score and label required for the HGR, and calculates the coordinates for the wrist landmarks. The main part of the implementation a video capture is started where each frame needs to be processed by converting BGR to RGB and flipped horizontally.

```

61 def get_label(index, hand, results):
62     output = None
63     for idx, classification in enumerate(results.multi_handedness):
64         if classification.classification[0].index == index:
65
66             # Process results
67             label = classification.classification[0].label
68             score = classification.classification[0].score
69             text = '{} {}'.format(label, round(score, 2))
70
71             # Extract Coordinates
72             coords = tuple(np.multiply(
73                 np.array((hand.landmark[mp_hands.HandLandmark.WRIST].x, hand.landmark[mp_hands.HandLandmark.WRIST].y)),
74                 [640, 480]).astype(int))
75
76             output = text, coords
77
78     return output
79
80 cap = cv2.VideoCapture(0)

```

Figure 25: get_label function for processing HGR

The previously mentioned “MediaPipe Hands” module facilitates the detection and tracking of the hand landmarks within the frame and renders them on the image if detected. The “get_label” function then obtains the hand level and display it on the image followed by the “draw_finger_angles” function for visualising the finger angles on the image.

Connectivity of Final Elements to Unity

The last connectivity stage was the use of Transmission Control Protocol (TCP) which facilitated the transfer for the IMU and MediaPipe to Unity. A Graphical User Interface (GUI) was developed using PySimpleGUI to scan the IMUs and use TCP to transfer the data to Unity. Figure 26 shows a screenshot of the GUI.

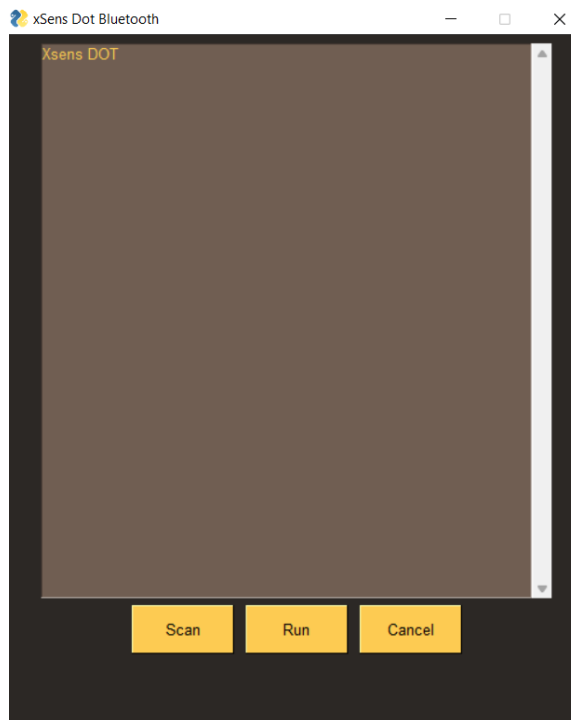


Figure 26: GUI developed for scanning IMUs and transfer to Unity

The TCP transmission required the use of the ZMQ (ZeroMQ) library, which is linked to a set address and port which is typically set as “tcp://localhost:5555” in the set-up stage. The “send_json()” function is utilised to transmit the required data via a socket which transmits the angle data as a JSON object to the TCP end point. Use of JSON allowed for utilising the inherent structure and nesting capabilities of data structures, providing compatibility across different programming languages and platforms. This is why JSON was selected for interactivity of Unity and Python codes. A polling procedure was implemented to make sure the socket connection is robust. The polling object is registered with the socket and timeout duration of 10milliseconds is configured. The “poll()” function can evaluate the status of the buffer for the socket to determine occupancy. If the buffer gets full, the socket is connected and

disconnected to the same endpoint to prevent potential blocking issues. “is_buffer_full” variable is monitored to determine the occupancy status of the buffer. In Unity, a TCP server is created using a “ServerReceiver” C# code. This code utilises the NetMO library to effectively manage TCP socket communication. When the “ServerReceiver” object is initiated, a “PullSocket” is created to listen on the localhost port (in this case 5555) and waits incoming transmission from the TCP client. Figure 27 demonstrate the code for capturing and storing the data into a set data type which goes through the steps mentioned below:

- 1) Instantiation of the first ServerReceiver
- 2) Creation of a new server
- 3) Start the server.
- 4) Creation of a character array to store the JSON packets from a string.
- 5) Removal of non-numeric entries from the string and data storage in different elements inside and array of strings.
- 6) Conversation of data string to float and storage as a 3-D vector.
- 7) Aborting the transmission

```
1  Public static Server server;
2
3  Server = new Server();
4
5  Server.Start();
6
7  _lineread1 = server.GetSensor1();
8
9  storeSplitter1 = _lineread1.Split(_delimiter,StringSplitOptions.RemoveEmptyEntries);
10
11 euler.x = float.Parse(storeSplitter1[0]);
12 euler.y = float.Parse(storeSplitter1[1]);
13 euler.z = float.Parse(storeSplitter1[2]);
14
15 Server.Stop()
```

Figure 27: Code for capturing and storing the movement data to Unity.

3.4.3 Importing Avatars in Unity for Visualisation of Movement

To be able to visualise the exercises and map the movement of the participants an avatar was required to be created in Unity. The created avatar needed to replicate real-time movement of the user. The designed model for the avatar was obtained from [101] due to its compatibility with Unity and can be seen in Figure 28.

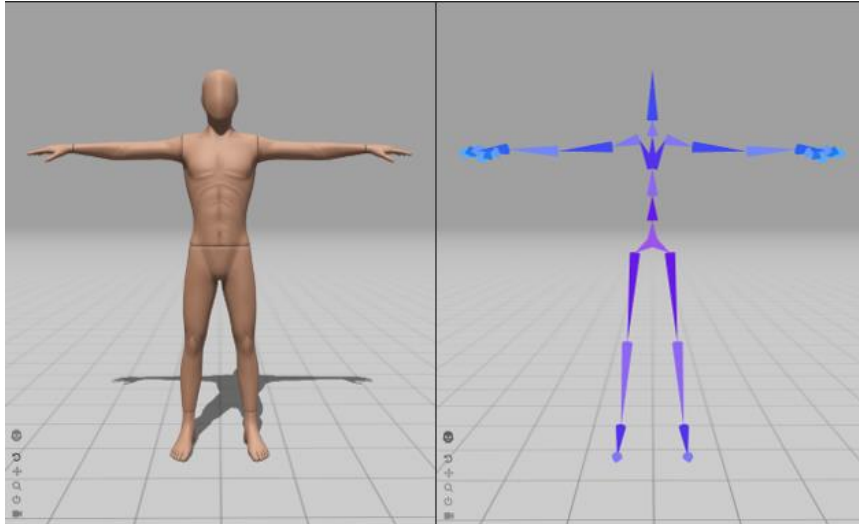


Figure 28: Base of Avatar used for replicating user movements.

The avatar provides flexibility by facilitating various arrays of options for manipulating the joints of the model. After importing the model into Unity, a script was written for controlling joint transformations. The different controllable joints within the model can be seen in Figure 29.

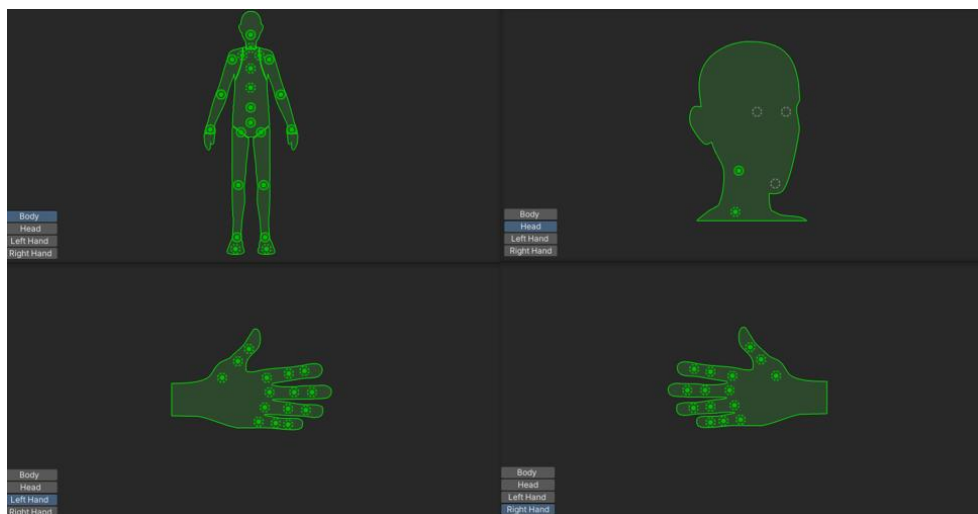


Figure 29: Mapping capabilities of the implemented avatar

Once the avatar was successfully imported into Unity, a default T-pose became visible in the Unity scene. As scene in Figure 30 several joints can be manipulated at this stage as seen in Hierarchy panel to the left-hand side of the model.



Figure 30: T-pose view of the avatar with Hierarchy panel

3.4.4 Animation Set up in Unity for Joint Movements

To provide a more visual experience to the user, animation was set up for the joint movements within Unity. To achieve this, the animation tool provided by Unity was utilised. This tool allowed for dynamic movements to be mapped to the joint transitions and allowed for different poses for the avatar depending on the rehabilitation exercises taking place. This change of pose helped put more focus on the joint that was being affected as part of the rehabilitation exercise. The animations also provided a more pleasing and engaging visual experience for the user since the animations provided the means for the avatar to mimic the user's movement more realistically.

As an example of specific poses required for the avatar, a side view was designed for hand rehabilitation exercises that provided clear visibility for the wrist. This special pose allowed the ROM of the wrist to be more easily visible by the user. A dedicated camera transition was then designed to dynamically transition from the main pose to the hand exercise pose. To initiate the camera transitions, an invisible button was placed on the arm that was used as the trigger of the transition. This button would be activated by the user upon clicking on the arm of the avatar. The script in Figure 31 demonstrates changes in priority of the desired camera and Boolean triggers that facilitate the transition.

```

    animator.enabled = true;
    //change camera priority
    fullBodyCamera.Priority = 0;
    LeftArmCamera.Priority = 1;
    LeftHandCamera.Priority = 0;
    //Animation
    animator.SetBool("isSittingLeftArm", true);

```

Figure 31: Settings priorities and Boolean triggers for camera transitions

3.4.5 Canvas Setup to Streamline Navigation in Unity

The visualisation required to facilitate a streamlined navigation for different exercises was made possible by designing an accessible interface by integrating multiple canvases in Unity. Each designed canvas facilitated a set of exercises and settings by providing buttons and instructions for a particular exercise. This provides an easy to navigate layout for the user to be able to select desired exercises or modify relevant settings. Figure 32 provides a list the canvases that were designed.

```

▶ mainMenuCanvas
▶ RewardCanvas
▶ setThreshHoldCanvas
▶ SummaryCanvas
▶ headCanvas
▶ leftArmCanvas
▶ LeftHandCanvas
▶ LeftHandSelectGameCanvas
▶ leftExtensionCanvas
▶ leftRadialCanvas
▶ leftPronationCanvas
▶ leftExtensionSelectGameCanvas
▶ leftRadialSelectGameCanvas
▶ leftPronationSelectGameCanvas
▶ rightArmCanvas

```

Figure 32: List of designed canvases

A user interface manager script was developed to maintain smooth navigation between different canvases. This script manages visibility of each canvas through different button clicks. To achieve this, each respective canvas needed to be associated with a GameObject and specific functions were created to trigger the button clicks. The SetActive() function was then utilised to hid or reveal the required canvas.

3.4.6 Setting Thresholds for Movement Exercises in Unity

As demonstrated in Chapter 2, customizability of the gamified exercises is a crucial component of rehabilitation. To address this requirement, threshold settings needed to be implemented so each exercise could be customized depending on users' requirements such as initial ROM/AROM and other attributes that are usually set by the clinicians during a rehabilitation program. The thresholds were also an important aspect of the feedback given to the user during the exercises as they determined when feedback is given regarding their target range of movement or number of daily repetitions of the exercises. Hand movement exercises were selected as an example of different threshold requirements within a rehabilitation program. To adjust the thresholds, slides were implemented as seen in Figure 33

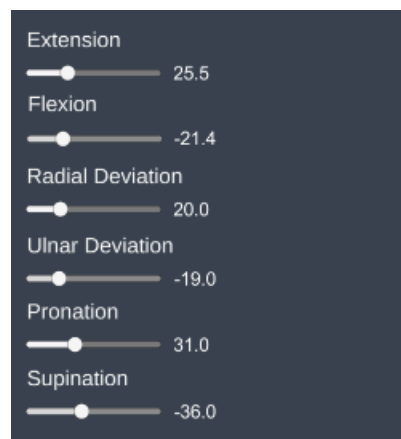


Figure 33: Slider menu for setting thresholds

To facilitate a more user-friendly experience, slide values needed to be saved so they can be loaded on future startup of the application. To achieve this, a script called “SliderScript.cs” as seen in Figure 34 was developed that monitors the values set by the slide and saves them as float data types. This data will eventually be used as the exercise's thresholds defined by the clinicians and set by the user.

```
//Upon starting the program, the slider values will be listened and converted to float
0 Unity Message | 0 references
void Start()
{
    LoadValues();

    pronationSlider.onValueChanged.AddListener((a) =>
    {
        pronationSliderText.text = a.ToString("0.0");
    });

    supinationSlider.onValueChanged.AddListener((b) =>
    {
        supinationSliderText.text = b.ToString("0.0");
        supinationBaseAngle = b;
    });

    radialSlider.onValueChanged.AddListener((c) =>
    {
        radialSliderText.text = c.ToString("0.0");
        radialBaseAngle = c;
    });

    ulnarSlider.onValueChanged.AddListener((d) =>
    {
        ulnarSliderText.text = d.ToString("0.0");
        ulnarBaseAngle = d;
    });

    extensionSlider.onValueChanged.AddListener((e) =>
    {
        extensionSliderText.text = e.ToString("0.0");
        extensionBaseAngle = e;
    });

    flexionSlider.onValueChanged.AddListener((f) =>
    {
        flexionSliderText.text = f.ToString("0.0");
        flexionBaseAngle = f;
    });
}
```

Figure 34: Script for saving the set threshold values obtained the sliders.

The script utilises `PlayerPrefs.SetFloat()` function within Unity to store the data internally in pairs as demonstrated in Figure 35.

```
//Storing the slider values upon save
0 references
public void SaveSettingButton()
{
    savedPronationAngle = pronationSlider.value;
    PlayerPrefs.SetFloat("SavedPronationAngle", savedPronationAngle);

    savedSupinationAngle = supinationSlider.value;
    PlayerPrefs.SetFloat("SavedSupinationAngle", savedSupinationAngle);

    savedRadialAngle = radialSlider.value;
    PlayerPrefs.SetFloat("SavedRadialAngle", savedRadialAngle);

    savedUlnarAngle = ulnarSlider.value;
    PlayerPrefs.SetFloat("SavedUlnarAngle", savedUlnarAngle);

    savedExtensionAngle = extensionSlider.value;
    PlayerPrefs.SetFloat("SavedExtensionAngle", savedExtensionAngle);

    savedFlexionAngle = flexionSlider.value;
    PlayerPrefs.SetFloat("SavedFlexionAngle", savedFlexionAngle);

    LoadValues();
}
```

Figure 35: Script to store values obtained from the sliders.

Finally, LoadValues() function within Unity was utilised to capture the saved values obtained from the sliders. This function uses PlayerPrefs.GetFloat() method to retrieve the float values and assign them to the relevant corresponding float variable. This variable is then used to update the slider value and display a text to the user that the value loading process was successful. The LoadValues() function is called in program startup and the save process to automatically save the latest values. This means the system is using the latest values for optimal functionality. The script used for this stage can be seen in Figure 36.

```

//Load the saved slider values
3 references
void LoadValues()
{
    float savedPronationAngle = PlayerPrefs.GetFloat("SavedPronationAngle");
    pronationSlider.value = savedPronationAngle;
    pronationSliderText.text = savedPronationAngle.ToString("0.0");
    pronationBaseAngle = pronationSlider.value;

    float savedSupinationAngle = PlayerPrefs.GetFloat("SavedSupinationAngle");
    supinationSlider.value = savedSupinationAngle;
    supinationSliderText.text = savedSupinationAngle.ToString("0.0");
    supinationBaseAngle = supinationSlider.value;

    float savedRadialAngle = PlayerPrefs.GetFloat("SavedRadialAngle");
    radialSlider.value = savedRadialAngle;
    radialSliderText.text = savedRadialAngle.ToString("0.0");
    radialBaseAngle = radialSlider.value;

    float savedUlnarAngle = PlayerPrefs.GetFloat("SavedUlnarAngle");
    ulnarSlider.value = savedUlnarAngle;
    ulnarSliderText.text = savedUlnarAngle.ToString("0.0");
    ulnarBaseAngle = ulnarSlider.value;

    float savedExtensionAngle = PlayerPrefs.GetFloat("SavedExtensionAngle");
    extensionSlider.value = savedExtensionAngle;
    extensionSliderText.text = savedExtensionAngle.ToString("0.0");
    extensionBaseAngle = extensionSlider.value;

    float savedFlexionAngle = PlayerPrefs.GetFloat("SavedFlexionAngle");
    flexionsSlider.value = savedFlexionAngle;
    flexionsSliderText.text = savedFlexionAngle.ToString("0.0");
    flexionBaseAngle = flexionsSlider.value;

    gameExtensionBaseAngle = savedExtensionAngle;
    gameFlexionBaseAngle = savedFlexionAngle;
    gameRadialBaseAngle = savedRadialAngle;
    gameUlnarBaseAngle = savedUlnarAngle;
    gamePronationBaseAngle = savedPronationAngle;
    gameSupinationBaseAngle = savedSupinationAngle;
}

```

Figure 36: The LoadValues() script used for reading values.

An alternative approach for loading threshold settings was provided after consulting with clinical collaborators of this research. In certain cases, such as geriatric users, younger children, or rehabilitation after major trauma, adjusting of the threshold setting needs to be closely monitored by a clinician. To facilitate closer monitoring, the recommended settings can be provided via a CSV file created by the clinicians. This is done by a LoadCsv() function as seen in Figure 37.

```

}
//Load the threshold settings from a CSV file provided by healthcare professionals
0 references
public void LoadCsv()
{
    // Open the CSV file
    StreamReader reader = new StreamReader(filePath);

    // Read the CSV file line by line
    string fileContent = reader.ReadToEnd();
    string[] lines = fileContent.Split('\n');

    // Store the values in a two-dimensional array
    csvData = new string[lines.Length][];
    for (int i = 0; i < lines.Length; i++)
    {
        csvData[i] = lines[i].Split(',');
    }

    loadedExtensionAngle = float.Parse(csvData[1][1]);
    loadedFlexionAngle = float.Parse(csvData[1][3]);
    loadedRadialAngle = float.Parse(csvData[2][1]);
    loadedUlnarAngle = float.Parse(csvData[2][3]);
    loadedPronationAngle = float.Parse(csvData[3][1]);
    loadedSupinationAngle = float.Parse(csvData[3][3]);
    extensionGoal = int.Parse(csvData[6][2]);
    radialGoal = int.Parse(csvData[7][2]);
    pronationGoal = int.Parse(csvData[8][2]);
    fingerGoal = int.Parse(csvData[9][2]);

    //get the exercise goal from saved file
    PlayerPrefs.SetInt("ExtensionGoalToday", extensionGoal);
    dg.extensionTarget = PlayerPrefs.GetInt("ExtensionGoalToday");
    PlayerPrefs.SetInt("RadialGoalToday", radialGoal);
    dg.radialTarget = PlayerPrefs.GetInt("RadialGoalToday");
    PlayerPrefs.SetInt("PronationGoalToday", pronationGoal);
    dg.pronationTarget = PlayerPrefs.GetInt("PronationGoalToday");
    PlayerPrefs.SetInt("FingerGoalToday", fingerGoal);
    dg.fingerTarget = PlayerPrefs.GetInt("FingerGoalToday");

    // Close the CSV file
    reader.Close();
}

```

Figure 37: LoadCsv() function for loading pre-defined exercise thresholds

3.4.7 Implementing Rehabilitation Exercises in Unity

To demonstrate the capability of adding exercises to the framework, four hand movement exercises were selected and implemented as a template. These exercises were extension/flexion, radial/ulnar deviations, and pronation/supination, as seen in Figure 38. A finger touching exercises was also implemented as an example of HGR. As previously discussed in Chapter 2, in addition to ROM, the number of repetitions and hold time at pre-defined thresholds needed to be recorded during the exercise sessions. To achieve this, values captured from the threshold settings input field, either through user input or by uploading the CSV file by the clinicians, are utilised as the

limits for the exercises. This functionality was implemented in a script as seen in Figure 39.

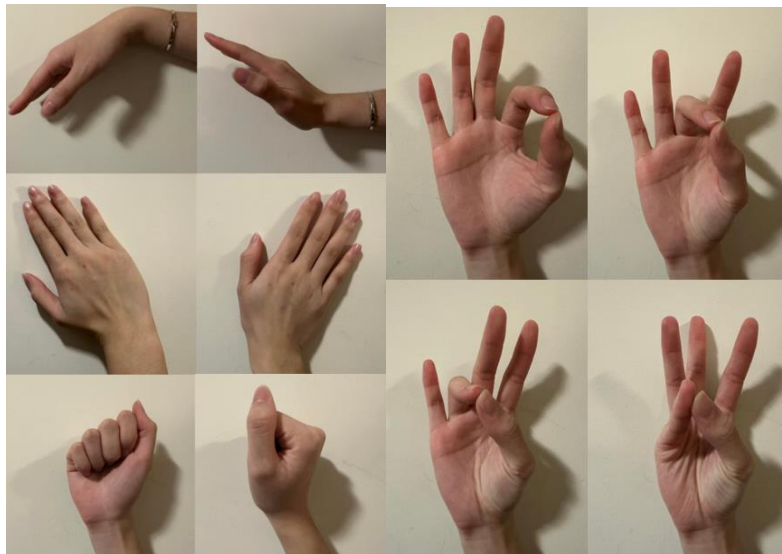


Figure 38: Wrist movement and finger movement exercises used as example implementation.

```

//To get the number of repetition from the input field
0 references
public void GetExtensionRepetitionInput(string input)
{
    if (int.TryParse(input, out int value))
    {
        numReps = int.Parse(input);
        gameNumRepExtension = numReps;
        Debug.Log("the input is: " + input);
    }
    else
    {
        Debug.Log("Please enter integer only.");
    }
}

//to Get the hold time from the input field
0 references
public void GetExtensionHoldTimeInput(string input)
{
    if (int.TryParse(input, out int value))
    {
        holdTime = int.Parse(input);
        gameHoldTimeExtension = holdTime;
        Debug.Log("the input is: " + input);
    }
    else
    {
        Debug.Log("Please enter integer only.");
    }
}

```

Figure 39: Script for setting thresholds limits in exercises

A validation mechanism has been implemented through the TryPars() function. This function makes sure valid inputs are accepted via checking the integer inputs and prompts the user to re-enter if an invalid value has been selected. The wrist rehabilitation exercises example provides some guides to engage with the application by providing prompts to start the exercises, highlight thresholds, and start a counter when a hold position is achieved. Depending on the exercise, different attributes of the IMU was selected for example, extension/flexion relies on Pith angles and radial/ulnar deviation relies on the Yaw angle of the IMU.

```

//Thumb Touching Exercise Implementation
1 reference
public void StartExercise()
{
    isIndex = true;
    if (reps > 0)
    {
        if (isIndex)
        {
            currentExerciseText.text = "Touch Index Finger " + reps + " reps to go.";
            if (leftIndexAngle < 8)
            {
                isIndex = false;
                isMiddle = true;
            }
        }
        if (isMiddle)
        {
            currentExerciseText.text = "Touch Middle Finger";
            if (leftMiddleAngle < 8)
            {
                isMiddle = false;
                isRing = true;
            }
        }
        if (isRing)
        {
            currentExerciseText.text = "Touch Ring Finger";
            if (leftRingAngle < 8)
            {
                isRing = false;
                isPinky = true;
            }
        }
        if (isPinky)
        {
            currentExerciseText.text = "Touch Pinky";
            if (leftPinkyAngle < 8)
            {
                isPinky = false;
                isIndex = true;
                reps--;
            }
        }
    }
    else
    {
        currentExerciseText.text = "Exercise Done";
        isStartExercise = false;
        stopExerciseButton.SetActive(false);
        startExerciseButton.SetActive(true);
        currentFinger++;
        PlayerPrefs.SetInt("CurrentFinger", currentFinger);
    }
}
}

```

Figure 40: Script for thumb touching exercise.

The thumb touching exercise utilises four data points provided from the MediaPipe script that are stored in the Unity environment. Each of these data points relate to the distance between the thumb and other fingers. This distance is calculated by the angle formed by tip of the thumb and the tip of the other finger and the metacarpophalangeal joints between them. The python code for this process was previously presented in Figure 23. For this exercise, patients were required to touch the tip of each finger with the tip of their thumb in a sequential manner. To detect the touches, low angle calculated based on MediaPipe data was utilised with the high angle pointing to absence of contact. Figure 40 provides the script for this implementation where an 8-degree angle threshold was utilised to indicate movement towards the thumb. This angle accommodates any potential inaccuracy in joint measurement while providing a reliable method for finding the direction of fingers. To calibrate the exercise, the script in Figure 41 was implemented as the initialisation method. This

function utilises the space key as an input for method for resetting all calculated joint angles to zero which standardizes the starting point for the required measurements.

```
if (Input.GetKeyDown(KeyCode.Space))
{
    //sensor 1 left wrist
    tempXAngle = euler.x;
    tempYAngle = euler.y;
    tempZAngle = euler.z;

    //sensor 2 left forearm
    tempXAngleLF = euler2.x;
    tempYAngleLF = euler2.y;
    tempZAngleLF = euler2.z;
}

//Calibrated angle
actualXAngleH = (euler.x - tempXAngle);
actualYAngleH = (euler.y - tempYAngle);
actualZAngleH = (euler.z - tempZAngle);
```

Figure 41: Script for calibration finger movement exercises

3.4.8 Implementation of Rehabilitation Goals via Dashboard Design

Rehabilitation programs define goals where the user has to perform a pre-defined number of exercises for a pre-determined number of daily repetitions which needs to be maintained over several weeks. A daily goal dashboard with a continuity system (known in game developed as streaks) was developed to encourage the user to persist with the rehabilitation program. Tracking this information also provides the clinician a view on how often the user participated in the rehabilitation exercises outside the clinical environment. This overview combined with tracking of ROM and smoothness measures for the patients can provide a clear view of how effective the rehabilitation program has been for the user. The daily goals are set by the clinicians and provided to Unity using a CSV file. A script titled “StreakSystem.cs” was developed to check if the user has been engaging the exercises set and if the exercises were done consecutively for a set number of days which determines the streak. The streak would reset if the user were not able to maintain the consecutive days as demonstrated in Figure 42. A badge system was then implemented as seen in Figure 43 to provide visual feedback to the user on how many streaks were maintained.


```

//Implementation of streak system
@ Unity Message | 0 references
private void Start()
{
    // load the streak count and last completed date from PlayerPrefs
    streakCount = PlayerPrefs.GetInt(STREAK_COUNT_KEY, 0);
    long ticks = 0;
    lastCompletedDate = new DateTime(ticks);
    if (long.TryParse(PlayerPrefs.GetString(LAST_COMPLETED_DATE_KEY, "0"), out ticks))
    {
        lastCompletedDate = new DateTime(ticks);
    }
    else
    {
        // Set the last completed date to epoch time if it cannot be parsed
        lastCompletedDate = new DateTime(0);
    }
}

@ Unity Message | 0 references
private void Update()
{
    // Check if tasks were completed today

    if (dg.exerciseComplete)
    {
        // Check if the last completed date was yesterday
        if (lastCompletedDate.Date == DateTime.Today.AddDays(-1))
        {
            if (!hasRun)
            {
                // Increment the streak count
                streakCount++;
                hasRun = true;
            }
        }
        else if (lastCompletedDate.Date < DateTime.Today.AddDays(-1))
        {
            // Reset the streak count to 1 if the last completed date was before yesterday
            streakCount = 1;
        }

        // Save the streak count and last completed date to PlayerPrefs
        PlayerPrefs.SetInt(STREAK_COUNT_KEY, streakCount);
        PlayerPrefs.SetString(LAST_COMPLETED_DATE_KEY, DateTime.Now.Ticks.ToString());
        PlayerPrefs.Save();
    }
    else
    {
        // Reset the streak count to 0 if tasks were not completed for three consecutive days
        if (DateTime.Today.Subtract(lastCompletedDate.Date).Days >= 3)
        {
            streakCount = 0;

            // Save the streak count and last completed date to PlayerPrefs
            PlayerPrefs.SetInt(STREAK_COUNT_KEY, streakCount);
            PlayerPrefs.SetString(LAST_COMPLETED_DATE_KEY, DateTime.Now.Ticks.ToString());
            PlayerPrefs.Save();
        }
    }

    streakText.text = "Streak: " + streakCount;
}

```

Figure 42: Scrip for checking the streak

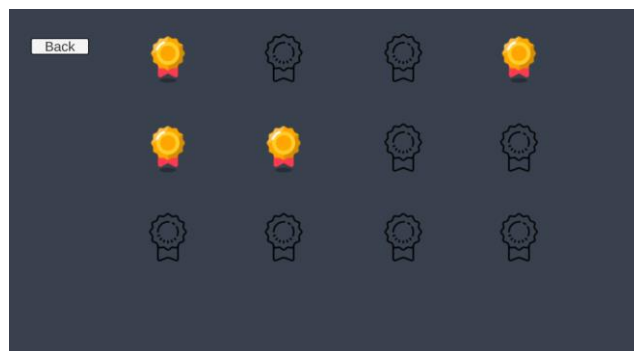


Figure 43: Badge system for rewarding the pare after keeping streaks

3.4.9 Designing Scene Transitions for In Unity

To provide a more engaging gamified experience for the user, several mini games were developed that provided a digitally gamified version of the existing rehabilitation exercises. These mini games needed to be kept in a separate scene in Unity and managed through a scene management system to transition between the avatar view and game view. The design allows for future mini games to be developed separately and added to the Unity program without requiring changing the main scene. To facilitate multiple mini games and their relevant scenes, a server object accessible by all C# scripts of the Unity code was initialised.

After the initiation of the server object, a script called GameLoader.cs was created to manage different transitions between the created scenes. This script provides a routine for loading scenes depending on their scene index as seen in Figure 44. A transition animator was also developed to make the transitions more aesthetically pleasing by providing a fade in/out effect.

```
//Load the scene with index of 6
0 references
public void LoadSixthGame()
{
    Time.timeScale = 1;
    StartCoroutine(LoadGame(6));
}

//Coroutine method to load scenes
7 references
IEnumerator LoadGame(int gameIndex)
{
    transition.SetTrigger("Start");
    yield return new WaitForSeconds(transitionTime);
    SceneManager.LoadScene(gameIndex);
}
```

Figure 44: Script for loading different scenes.

3.4.10 Implementation of Audio Cues to Enhance Engagement

To further enhance the engagement and entertainment aspect of the gamified elements, audio cues such as sound effects and background music were designed and implemented in Unity. For implementation of the audio, a dedicated sound class was developed that included initial parameters for the audio source as seen in Figure 45. The AudioManager.cs script was then developed to be able to manage the audio playback components and characterises as seen in Figure 46. The function “FindObjectOfType<AudioManager>().Play()” was utilised to provide seamless

audio playing of a given source by adjusting characteristics such as volume, pitch and how often the given audio element would loop.

```
public class Sound
{
    public string name;
    public AudioClip clip;
    [Range(0f, 1f)]
    public float volume;
    [Range(.1f, 3f)]
    public float pitch;

    [HideInInspector]
    public AudioSource source;

    public bool loop;
}
```

Figure 45: Special class created for the audio elements.

```
public class AudioManager : MonoBehaviour
{
    public Sound[] sounds;

    void Awake()
    {
        foreach (Sound s in sounds)
        {
            s.source = gameObject.AddComponent<AudioSource>();
            s.source.clip = s.clip;
            s.source.volume = s.volume;
            s.source.pitch = s.pitch;
            s.source.loop = s.loop;
        }
    }

    public void Play(string name)
    {
        Sound s = Array.Find(sounds, sound => sound.name == name);
        s.source.Play();
    }
}
```

Figure 46: Audio manager used for adjusting different audio elements.

3.4.11 Overview of the Implemented Framework Interactivity

All the elements explained so far in this chapter were integrated to create the main page view seen in Figure 47 for the user going through the rehabilitation program. This interface combines these elements to allow the user to input their personal information, access exercise data, set or load thresholds, view daily goals, and view their badges. Navigational buttons are also provided so different menu can be access easily.

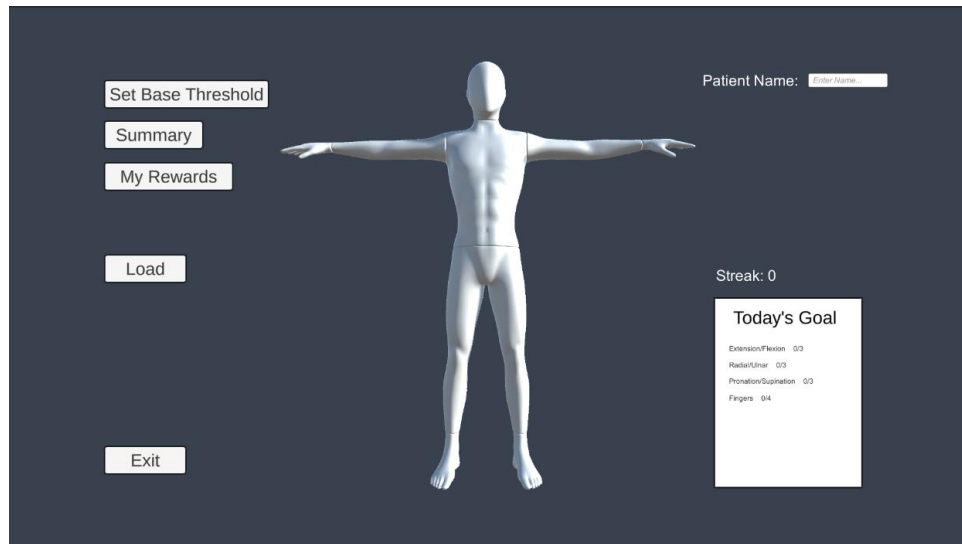


Figure 47: Main page view of the Unity application

Navigation to specific exercises is done by clicking on a relevant joint on the avatar which takes the user to a view suitable for those exercises and provides further options. For example, if the hand is pressed, the user is first taken to the view seen in Figure 48. This view allows the user to choose between different example hand movement exercises that can be selected.

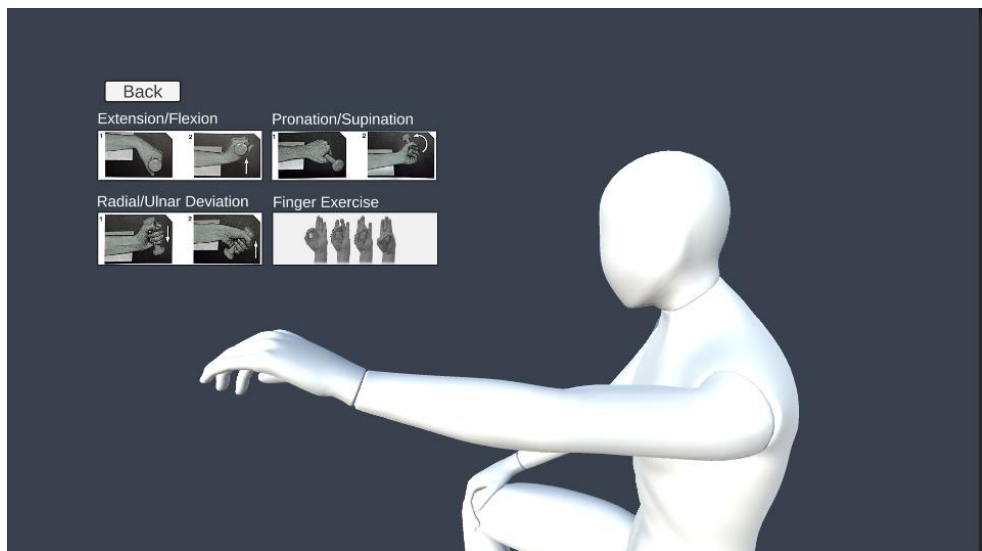


Figure 48: Example side view for wrist and hand movement exercises

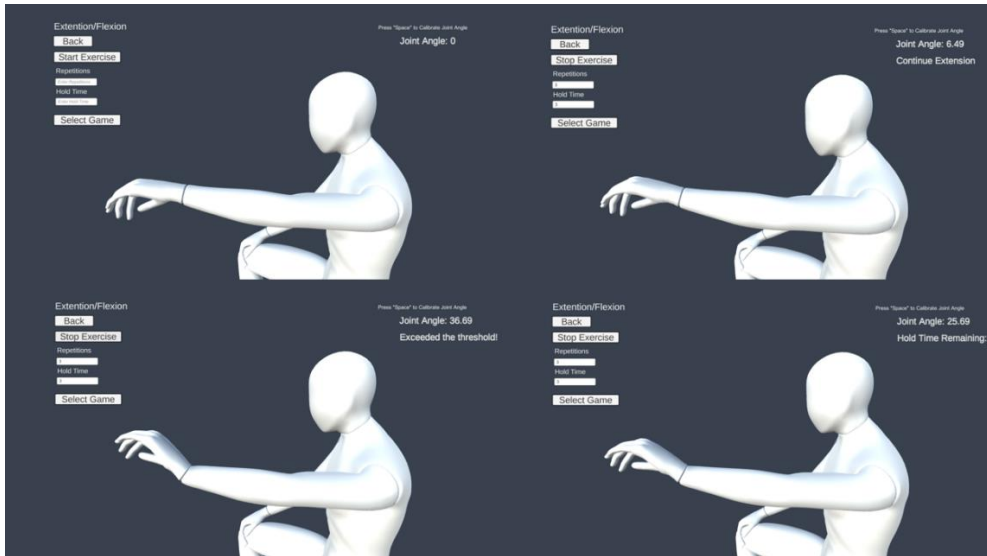


Figure 49: Example of a selected hand movement exercises (Flexion/Extension)

Figure 49 demonstrates the flexion/extension exercise that has been selected which will then show the joint angle feedback, threshold feedback and other settings to the user. Figure 50 demonstrates another exercise example for the thumb touching exercise with its relevant menus. As it can be seen, each exercise also includes a select mini game option which will then take the user to a mini game selection window relevant for the selected exercises. Each exercise can either be done in the avatar window or the mini game window. Details of the mini game part is provided in sub chapter 3.5.

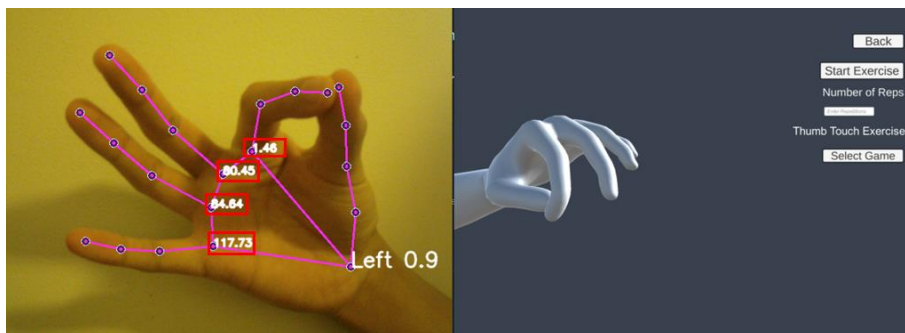


Figure 50: Example of a selected finger movement exercises (Thumb Touch)

3.4.12 Overview of the Generated Movement Data Reports

Different rehabilitation data such as exercise duration, exercise type, number of repetitions achieve, hold time, and exercise streak are stored after engaging with the exercises. The base line thresholds, joint angles, and smoothness measures are also

stored to provide an overview of the user's progress during the exercise program. Figure 51 shows an example of data that has been collected after a flexion/extension exercise.

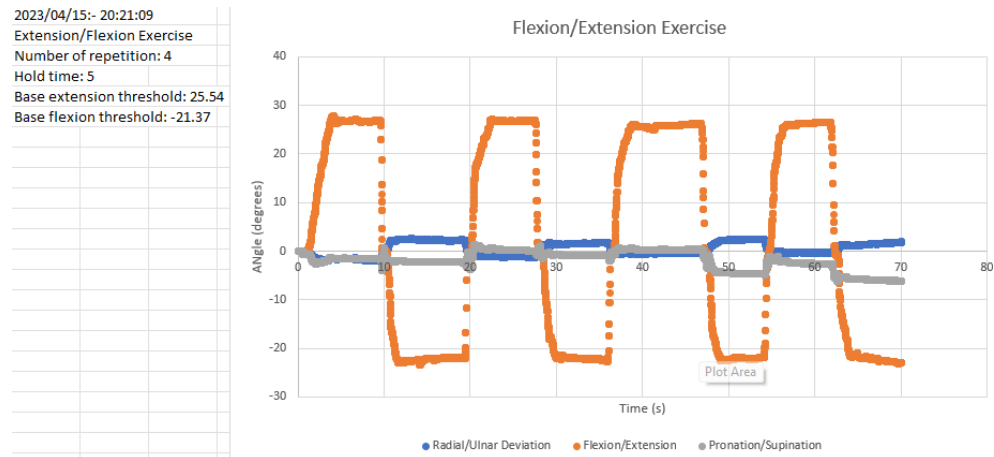


Figure 51: Example of collected data after engaging with a rehab exercise through Unity.

The joint angle measures are also collected after engaging with the mini games as seen in Figure 52 which demonstrates the ROM for this activity. In this example, clustered data demonstrates a low rate of wrist movement, and the spread-out data demonstrates a period of rapid movement.

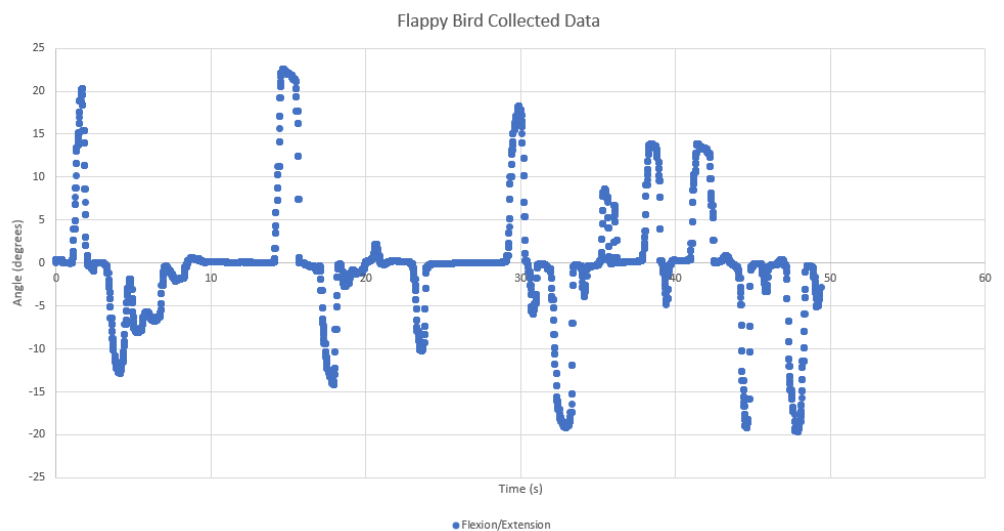


Figure 52: Example of collecting data after engaging with a mini game.

3.5 DEVELOPMENT OF GAMES FOR THE FRAMEWORK

3.5.1 Design Ideology for the Game Designs

After development of the backend for the visualisation environment and implementation of the measurement techniques, several video game experiences were developed to provide a more engaging gamified experience to the rehabilitation program. The user who is going through the rehabilitation program would be provided with the choice of engaging with the exercises either through the avatar interface or by selecting a minigame associated with a said exercise. Each of the developed video games which consisted of eight mini games and two larger games followed the following design ideology:

- 1) The games must replicate certain rehabilitation exercises.
- 2) The games must provide the means for movement data such as ROM, joint angles, and smoothness measurements to be captured.
- 3) The games must reward the user when an exercise has been done correctly via scoring systems and audiovisual feedback.
- 4) The scoring systems within the games should not encourage the user to under reach or overreach threshold for rehabilitation.
- 5) The games must be customizable based on the user's rehabilitation requirements.

The games that are described below utilize IMU sensors connected to back of the hand and slightly above the wrist as seen in Figure 53. The finger movement games utilise a laptop webcam and MediaPipe as the main input and interaction method. The interaction with these games can also be mapped to other joint movements such as elbow and knee. The use of framework for posture monitoring activities and exercises have been provided in Chapter 4.



Figure 53: IMU placement for wrist rehabilitation exercises

3.5.2 Design and Development of Mini Games in Unity

The mini games described below were designed by maintaining the ideology mentioned above. The aim was to provide a wide variety of games that can fit different tastes and age groups. The user would be able to select different games depending on which rehabilitation exercise has been selected. The mini games below utilised hand movement wrist exercises and finger movement exercises as an example of a rehabilitation program to follow.

Monster Chase Game

In this mini game, a zombie character is constantly chasing the player on a 2D plane where the movement of the player can be mapped to different types of hand movement depending on the exercise type. For example, one of the attributes could be mapping the movement to the time a threshold angle has been maintains. Once the specified number of repetitions have been completed, the player escapes the zombie. The chasing zombie will capture the player if the goals has not been met - which also leads to the game ending. The game aligns the gameplay mechanics with the selected exercises, in this case utilising IMU input, threshold settings, number of repetitions, and hold time. To provide customization capability to the game, the zombie's movement speed can be modified utilising the MoveEnemy.cs script as seen in Figure 54.


```

public class MoveEnemy : MonoBehaviour
{
    public float speed;

    private void Start()
    {
        speed = 3.0f;
    }

    void Update()
    {
        transform.position += Vector3.right * speed * Time.deltaTime;
    }
}

```

Figure 54: Script for modifying the zombie's speed

The CameraFollow.cs script was then implemented to make sure both characters can be viewed in the screen's boundary. Figure 55 demonstrates how this script follow the character's movement.

```

public class CameraFollow : MonoBehaviour
{
    public Transform target;
    public float smoothSpeed = 0.125f;
    public Vector3 offset;

    private void LateUpdate()
    {
        Vector3 desiredPosition = target.position + offset;
        Vector3 smoothedPosition = Vector3.Lerp(transform.position, desiredPosition, smoothSpeed);
        transform.position = smoothedPosition;
    }
}

```

Figure 55: Scrip that allows the camera to follow the characters.

Figure 56 shows the user view for the final implementation of the Monster Chase game. Some sample data collected from the interaction with the monster chase game has been provided in Appendix B.

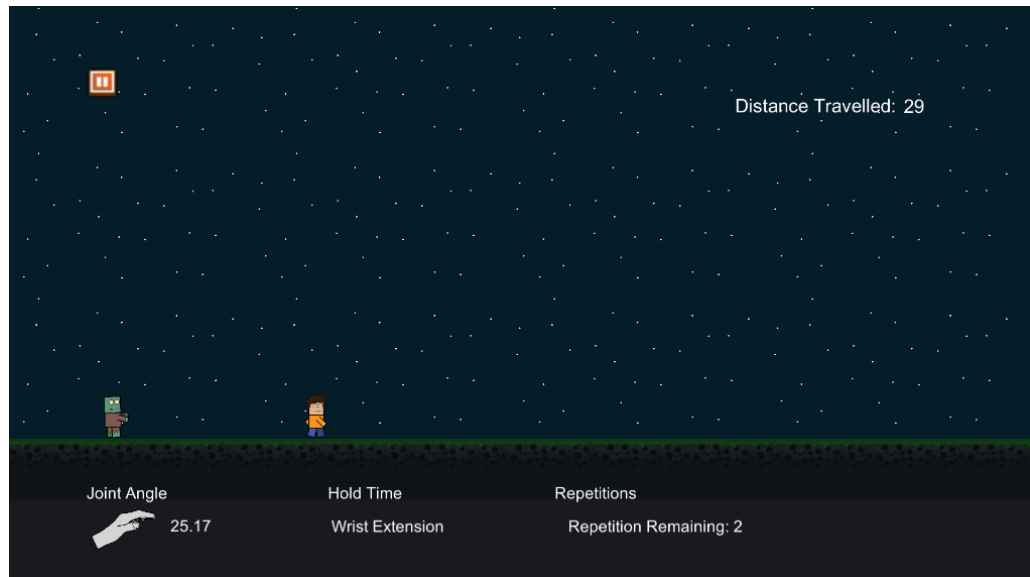


Figure 56: Screenshot of the user view in the Monster Chase game

Flap and Avoid Game

The “Flap and Avoid” game was developed to target flexion/extension exercises in hand and wrist rehabilitation. The main aim of this game as seen in Figure 57 is to avoid hitting the green pipes and collected the coins while controlling the movement of the bird character via wrist flexion/extension exercises attached to pitch angle of IMU data. The script that utilises PlayerMove() function to map IMU data to the game object can be seen in Figure 58.

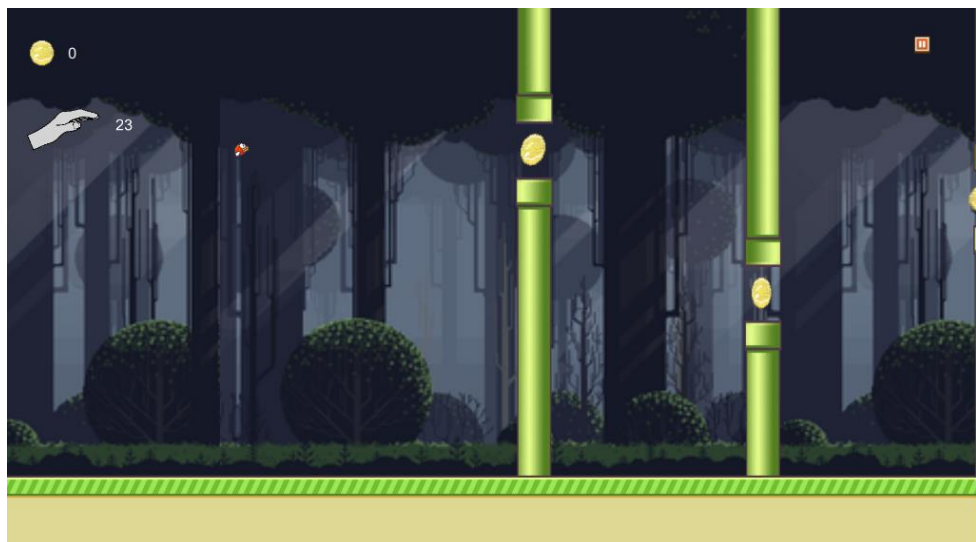


Figure 57: Screenshot of the Flap and Avoid Game

```

//move the bird along the y axis depending on the pitch angle received from the IMU
0 references
void PlayerMove()
{
    movementY = calibratedAngle / 25;

    transform.position += new Vector3(0f, movementY , 0f) * Time.deltaTime * moveForce;
}

```

Figure 58: Script for mapping IMU data to the player movement

This mini game also implements a random element by generating pipes at regular intervals of time which can be modified depending on rehabilitation requirement of the user. Each instantiation of the pipe generates a height value defined by the threshold settings of the exercise. The PipeSpawner.cs script can be seen in Figure 59 that demonstrates this process. Afterwards, the MovePipes.cs script was developed to facilitate the movement of the bird character between the pipe elements as seen in Figure 60. Finally, the OnCollisionEnter2D() function was implemented to take the player to a game over screen if the bird character collides with the pipes. This implementation can be seen in Figure 61. Sample data captured from this mini game can be seen in Appendix B.

```

@ Unity Script (1 asset reference) | 0 references
public class PipeSpawner : MonoBehaviour
{
    private float maxTime = 3f;
    private float timer = 0f;
    public GameObject Pipe;

    // Instantiate pipe prefab ever 3 seconds
    @ Unity Message | 0 references
    void Update()
    {
        if (timer > maxTime)
        {
            GameObject newPipe = Instantiate(Pipe);
            newPipe.transform.position = transform.position + new Vector3(0, Random.Range(-2, 5), 0);

            Destroy(newPipe, 20);
            timer = 0;
        }

        timer += Time.deltaTime;
    }
}

```

Figure 59: Script for implementing the random pipe height element.

```
Unity Script (1 asset reference) | 0 references
public class MovePipe : MonoBehaviour
{
    private float speed = 1.6f;

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
        transform.position += Vector3.left * speed * Time.deltaTime;
    }
}
```

Figure 60: Script for interaction between the pipe and the bird

```
Unity Message | 0 references
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag(PIPE_TAG) || collision.gameObject.CompareTag(SKY_TAG))
    {
        FindObjectOfType<AudioManager>().Play("Hit");
        FindObjectOfType<AudioManager>().Play("Die");
        sr.flipY = true;
        myBody.gravityScale = 1;

        isOver = true;
    }

    if (collision.gameObject.CompareTag(GROUND_TAG))
    {
        FindObjectOfType<AudioManager>().Play("Hit");
        StopBluetooth();
        gameManager.GameOver();
    }
}
```

Figure 61: Script for collision detection between the bird and the pipe

Hovercraft Game

The “Hovercraft” game was developed as a more advanced version of the flap and avoid game with more customizable features. The aim of this mini game is to avoid incoming obstacles while flying a hovercraft that moves up and down as seen in Figure 62.

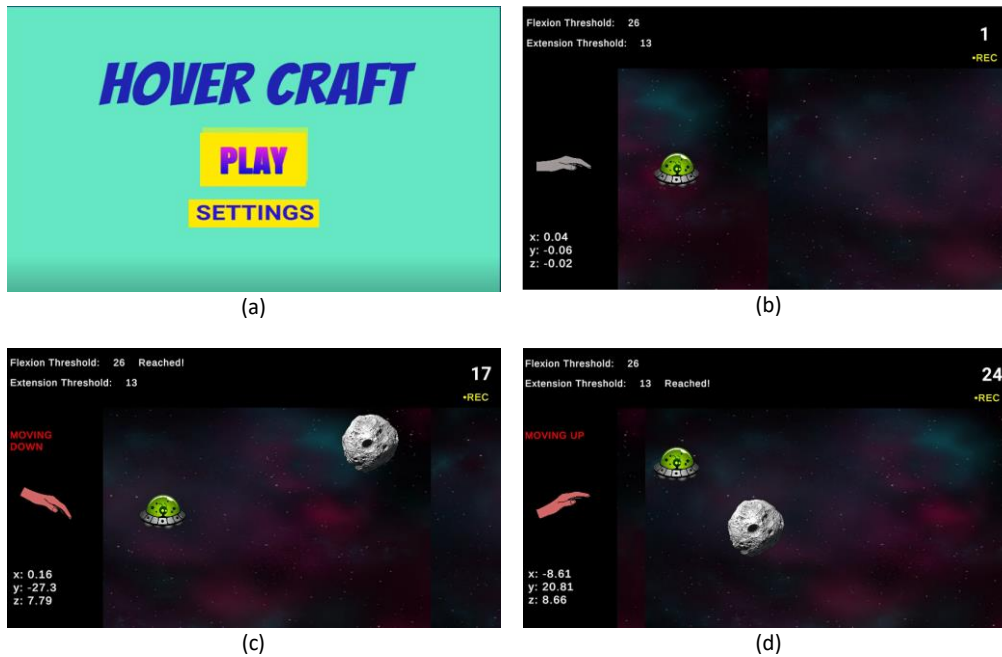


Figure 62: (a) Menu Screen; (b) Patient out of threshold region UFO movement flatlined (c) Flexion Threshold reached and UFO moves Down; (d) Extension Threshold reached and UFO moves UP.

The obstacles instantiated into the game are purposefully placed and manipulated in a way that will encourage the user to hold a flexed or extended hand position a hold time. The hovercraft game allows the customization of multiple elements such as customizing the delay in obstacle spawn time, modifying the obstacle sizes which leads to different hold times. This customizability facilitated the ability to modify the game depend on the user’s rehabilitation requirements. Sample data captured from this mini game can be seen in Appendix B.

Brick-Busting Game

The “Brick-Busting” mini games can be utilized for all three types of hand movement exercises similar to the Monster Chase game. In this game, the movement of the hand is mapped to IMU data and depending on the exercise type, different orientation data is used. The hand movement is attached to a paddle that is used to bounce a small ball to break some blocks as seen in Figure 63.

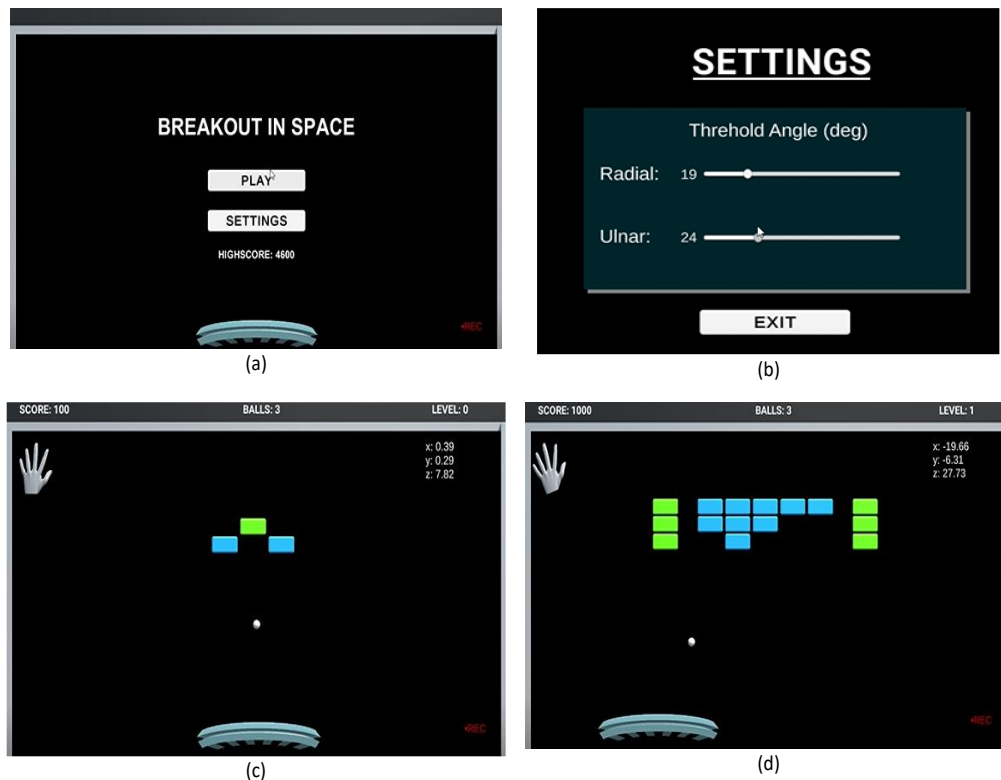


Figure 63: (a) Menu Screen; (b) Settings (c) Level 0 game environment; (d) Level 1 game Environment with Panel moving left.

Similar to previous mini games, different attributes of the game can be customized depending on the user requirements via the settings menu. Depending on the given threshold, the player paddle size will increase or decrease to allow for different initial ROM requirements. For example, Table 5 was utilized to modify ROM for ulnar/radial divisions with the ranges derived from [102]. A larger panel surface makes it easier for the user to engage in the exercise and avoids them from getting frustrated due to their limited ROM. While a smaller panel can make it more challenging as they increase their ROM.

The behaviour of the small ball is managed through the Ball.cs script which controls how the ball is moving, maintains the player score, and updates the high scores. The Rigidbody2D component is utilized for managing the physics of the ball and uses other UI elements such as highScoreText and scoreText to keep displaying the score. The implementation of game elements can be seen in Figure 64.

Table 5: Table to determining panel size based on angle thresholds.

Physical Properties	Specifications
Ulnar > 33 & Radial > 19	Small (1x original Scale)
15 > Ulnar <= 33 & 10 > Radial <= 19	Medium (3x)
Ulnar <= 15 & Radial <= 10	Large (5x)

```

@ Unity Message | 0 references
void Start()
{
    iswon = false;
    speed = -5f;
    rb = GetComponent<Rigidbody2D>();
    rb.velocity = Vector2.up * speed;

    highScoreText.text = "High Score: " + PlayerPrefs.GetInt("BreakOutHighScore", 0).ToString();
}

@ Unity Message | 0 references
void Update()
{
    brick.SetActive(true);
    scoreText.text = "Score: " + score.ToString();

    if (score > PlayerPrefs.GetInt("BreakOutHighScore", 0))
    {
        PlayerPrefs.SetInt("BreakOutHighScore", score);
        highScoreText.text = "High Score: " + score.ToString();
    }

    if(score >= 10 && score < 20)
    {
        sr.sprite = halfBroken;
    }
    else if(score >= 20 && score < 30)
    {
        sr.sprite = almostBroken;
    }
    else if(score >= 30 && iswon == false)
    {
        brick.SetActive(false);
        rb.velocity = Vector3.zero;
        manager.GameWon();
        iswon = true;
    }
}

```

Figure 64: Scrip for managing the small ball.

Every time the ball collides with the bricks, the score and speed of the ball is increased which is controlled by the script mentioned above. The AudioManager manages the audio cues when collision is detected with the brick. Figure 65 demonstrates the script for managing different collision scenarios. Sample data captured from this mini game can be seen in Appendix B.

```

Unity Message | 0 references
private void OnCollisionEnter2D(Collision2D other)
{
    if (other.gameObject.CompareTag("Brick"))
    {
        score++;
        speed -= -0.2f;
    }

    if (other.gameObject.CompareTag("Paddle"))
    {
        FindObjectOfType<AudioManager>().Play("HitPaddle");
    }

    if (other.gameObject.CompareTag("Brick"))
    {
        FindObjectOfType<AudioManager>().Play("HitBrick");
    }

    if (other.gameObject.CompareTag("Ground"))
    {
        rb.velocity = Vector3.zero;
        manager.GameOver();
    }
}

```

Figure 65: Script for managing the collision for the Brick-Busting game.

Dodge the Spike Game

The “Dodge the Spike” game was developed to facilitate pronation/supination wrist rehabilitation exercises by using the Roll angle of IMU. The aim of the game as seen in Figure 66 is to dodge falling spikes by reaching the required rehabilitation threshold. The main script for this game is similar to flap and avoid since they have similar characteristics of avoiding obstacles. This game included randomized elements, which makes it more suitable to late stages of rehabilitation program where the aim is constant movement of the joint rather than achieving certain thresholds. Sample data captured from this mini game can be seen in Appendix B.

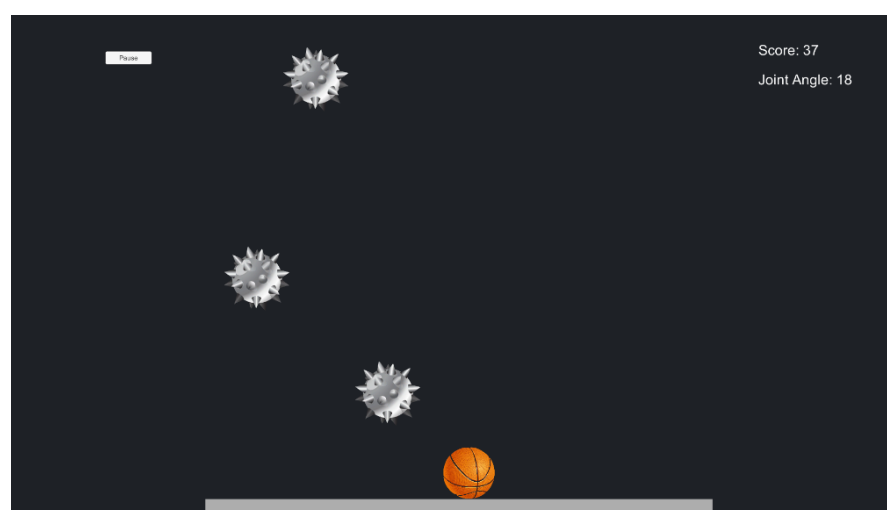


Figure 66: Screenshot of the Dodge the spike game

Tilting Mazes Game

The “Tilting Mazes” game uses rotation about the Z-axis to tilt a 2D maze left and right as seen in Figure 67. The objective of the game is to move the ball to the yellow diamond exit point while receiving feedback as seen in Figure 68.

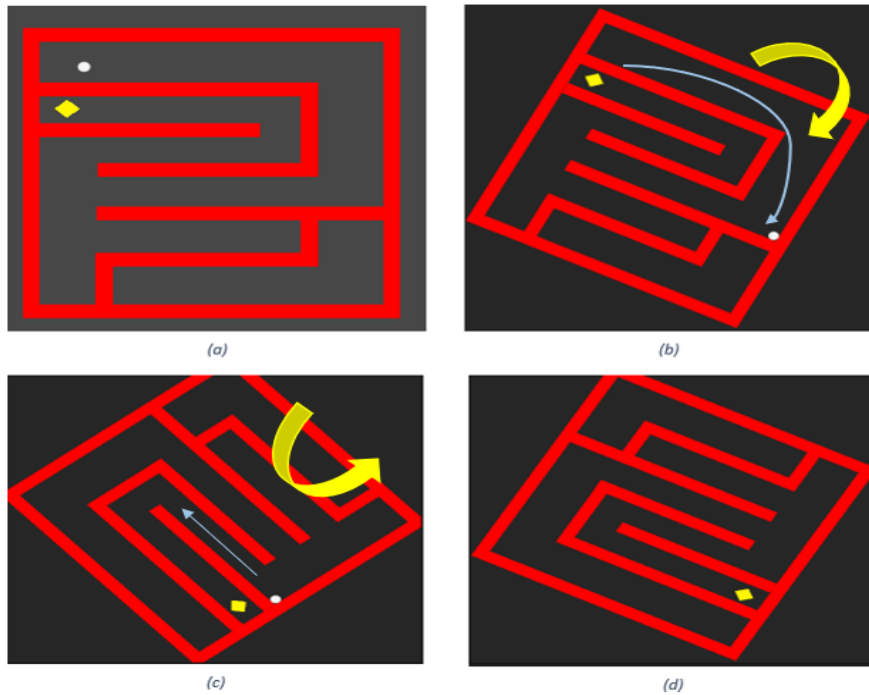


Figure 67: Tilting Maze Game: (a) Initiation for Right hand (b) Pronation tilts maze right (c) Supination rotates Maze left (d) Level complete

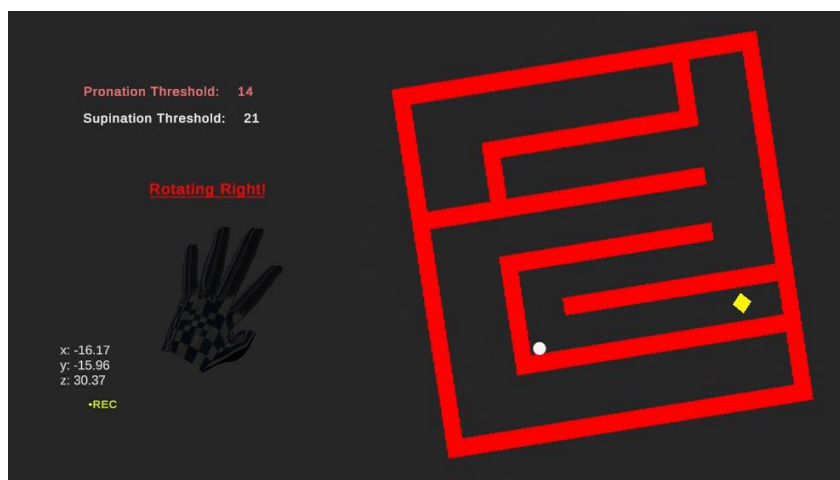


Figure 68: Feedback panel for the Tilting Maze Game

Upon entering the game environment through a main menu, the ball begins in a specify placement as chosen by use depending on their rehabilitation requirements.

The patient is then able to control the maze game object by rotating their hand to the set target angle. For a right-hand treatment, pronation movement tilts the maze right and supination left. To make this game more challenging for user three levels have been specifically designed to intuitively encourage the patient to hold their wrist in a set position as they try to reach the game objective. Sample data captured from this mini game can be seen in Appendix B.

Glasses Wiping Game

The “Glasses Wiping” game requires participants to perform bi-directional movements with their wrists repeatedly until the fog on the glasses is wiped off, presenting a tangible and measurable objective for patients to work towards as illustrated in Figure 69.

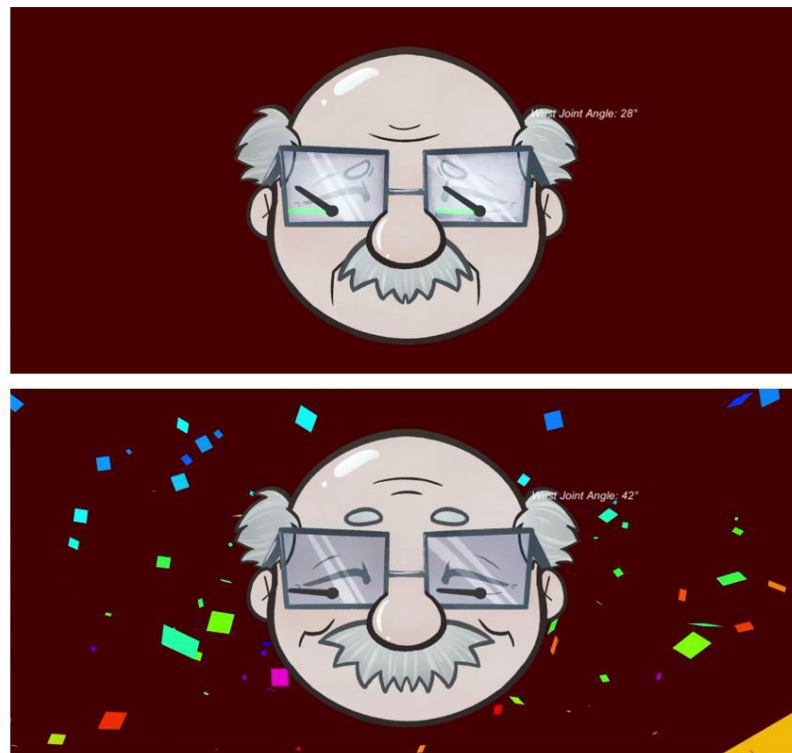


Figure 69: The Glasses Wiping game.

The user will move their wrist to meet the green wiper goal which will lead to triggering a short audio clip and 180-degree change in the wiper direction. Once the given number of repetitions have been achieved, the glasses become clean, and a particle confetti effect will trigger followed by a celebratory audio cue. The movement

of the wiper can be mapped to different wrist movements similar to the other mini games described above.

Piano Tiles Game

The “Piano Tiles” game was created as an example of finger rehabilitation exercises with a focus on thumb touching exercise. This game received angles from MediaPipe as the input method and the objective is for the user to touch their thumb using the finger that corresponds to the tile that has turned black. If the correct finger has been touched, the tile will turn green as seen in Figure 70.

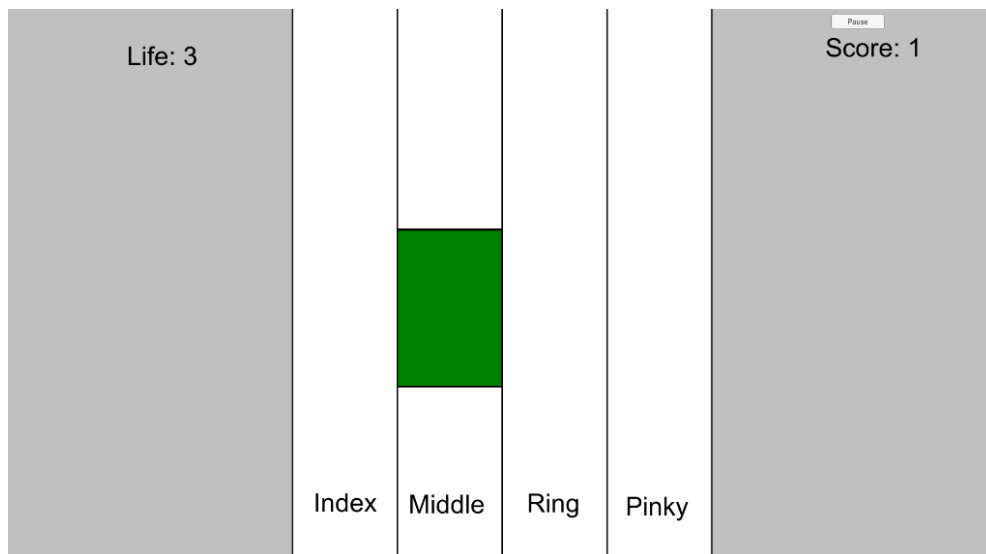


Figure 70: Piano Tiles Mini Game

The `TileSpawner.cs` script was developed to provide a random black tile generation as seen in Figure 71. The script monitors angles of the fingers and updates the corresponding tile as needed. If a read angle is less than 10 degrees, the tile colour will change to green and the score increases. This also leads to an increase in the speed of the black tile appearance followed by audio cues when correct finger is pressed managed by the `AudioManager`. If a wrong finger touches the thumb, the tile colour will change to red and the user’s life count will be reduced. The script in Figure 72 and Figure 73 maintains the user score and high score as well as displaying the timer using UI text elements.

```

if (timer > maxTime)
{
    randNum = UnityEngine.Random.Range(1, 5);

    if(randNum == 1)
    {
        newTile1 = Instantiate(tile1);
        newTile1.transform.position = transform.position + new Vector3(-3.27f, -10, 0);
        one = true;
        two = false;
        three = false;
        four = false;
    }
    else if(randNum == 2)
    {
        newTile2 = Instantiate(tile2);
        newTile2.transform.position = transform.position + new Vector3(-1.1f, -10, 0);
        one = false;
        two = true;
        three = false;
        four = false;
    }
    else if (randNum == 3)
    {
        newTile3 = Instantiate(tile3);
        newTile3.transform.position = transform.position + new Vector3(1.068f, -10, 0);
        one = false;
        two = false;
        three = true;
        four = false;
    }
    else if (randNum == 4)
    {
        newTile4 = Instantiate(tile4);
        newTile4.transform.position = transform.position + new Vector3(3.24f, -10, 0);
        one = false;
        two = false;
        three = false;
        four = true;
    }

    Destroy(newTile1, tileTime);
    Destroy(newTile2, tileTime);
    Destroy(newTile3, tileTime);
    Destroy(newTile4, tileTime);
    timer = 0;
}
}

```

Figure 71: Script for creating black tiles for the Piano Tiles game.

```

if (leftIndexAngle < 10 && one && !two && !three&&!four)
{
    newTile1.GetComponent<Renderer>().material.color = Color.green;
    currentScore++;
    maxTime -= 0.1f;
    tileTime -= 0.1f;
    FindObjectOfType<AudioManager>().Play("Index");
    one = !one;
    Debug.Log("Destroyed");
}
else if (leftMiddleAngle < 10 && one || leftRingAngle < 10 && one || leftPinkyAngle < 10 && one)
{
    newTile1.GetComponent<Renderer>().material.color = Color.red;
    one = !one;
    life--;
}

if (leftMiddleAngle < 10 && two &&!one&&!three&&!four)
{
    newTile2.GetComponent<Renderer>().material.color = Color.green;
    currentScore++;
    maxTime -= 0.1f;
    tileTime -= 0.1f;
    FindObjectOfType<AudioManager>().Play("Middle");
    two = !two;
    Debug.Log("Destroyed");
}
else if (leftRingAngle < 10 && two || leftIndexAngle < 10 && two || leftPinkyAngle < 10 && two)
{
    newTile2.GetComponent<Renderer>().material.color = Color.red;
    two = !two;
    life--;
}

if (leftRingAngle < 10 && three && !one && !two && !four)
{
    newTile3.GetComponent<Renderer>().material.color = Color.green;
    currentScore++;
    maxTime -= 0.1f;
    tileTime -= 0.1f;
    FindObjectOfType<AudioManager>().Play("Ring");
    three = !three;
    Debug.Log("Destroyed");
}
else if (leftMiddleAngle < 10 && three || leftIndexAngle < 10 && three || leftPinkyAngle < 10 && three)
{
    newTile3.GetComponent<Renderer>().material.color = Color.red;
    three = !three;
    life--;
}

```

Figure 72: Part 1 of the script for implementing the tiles game mechanism.

```

if (leftPinkyAngle < 10 && four && !one && !two && !three)
{
    newTile4.GetComponent<Renderer>().material.color = Color.green;
    currentScore++;
    maxTime -= 0.1f;
    tileTime -= 0.1f;
    FindObjectOfType<AudioManager>().Play("Pinky");
    four = !four;
    Debug.Log("Destroyed");
}
else if (leftMiddleAngle < 10 && four || leftIndexAngle < 10 && four || leftRingAngle < 10 && four)
{
    newTile4.GetComponent<Renderer>().material.color = Color.red;
    four = !four;
    life--;
}

if (currentScore > PlayerPrefs.GetInt("PTHIGHSCORE", 0))
{
    PlayerPrefs.SetInt("PTHIGHSCORE", currentScore);
    highScoreText.text = "High Score: " + currentScore.ToString();
}

scoreText.text = "Score: " + currentScore.ToString("0");
lifeText.text = "Life: " + life.ToString("0");
timer += Time.deltaTime;

```

Figure 73: Part 2 of the script for implementing the tiles game mechanism.

3.5.3 Design and Development of Larger Games in Unity

In addition to the mini games described above, two larger games were developed with the capability of covering a wide range of exercises. The detail of these implementation has been provided below.

Skiing Game

The Skiing game provides a layer skiing experience where the movement of a player character can be mapped to a different rehabilitation exercise. The game starts with a window as seen in Figure 74 fig so the use can input thresholds of a given exercise.

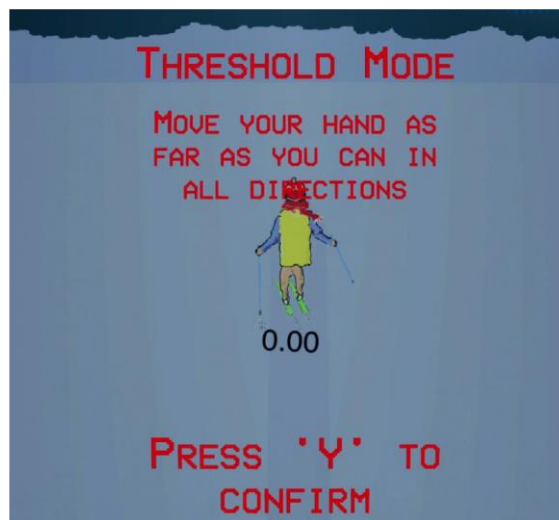


Figure 74: Threshold mode for setting baselines for the Skiing game

The objective of the game is to reach each threshold while avoiding upcoming trees in the pathway of the player avatar as seen in Figure 75. A linear interpolation to the character's position has been implemented for a smoother transition between positions on the screen providing a sense of floaty movement as seen in Figure 76.

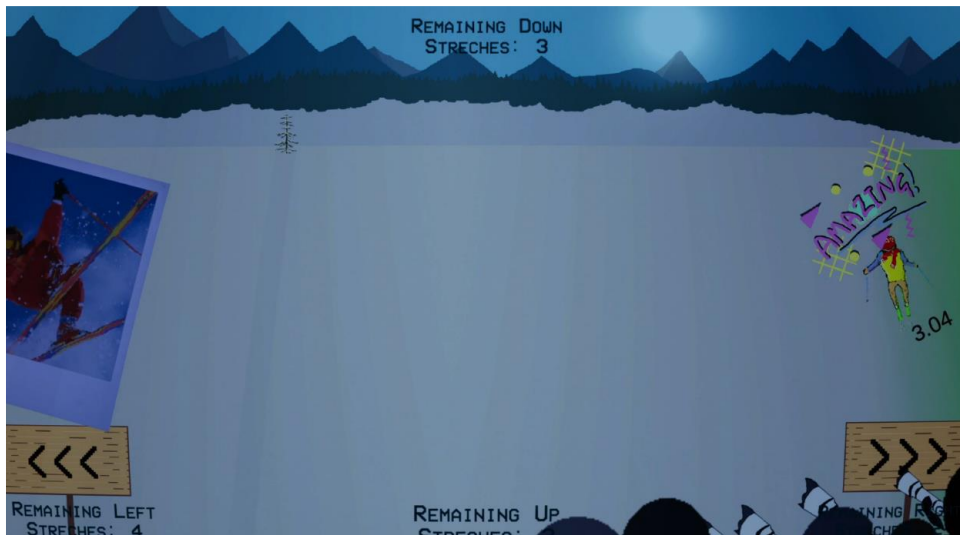


Figure 75: Main view mode of the Skiing game

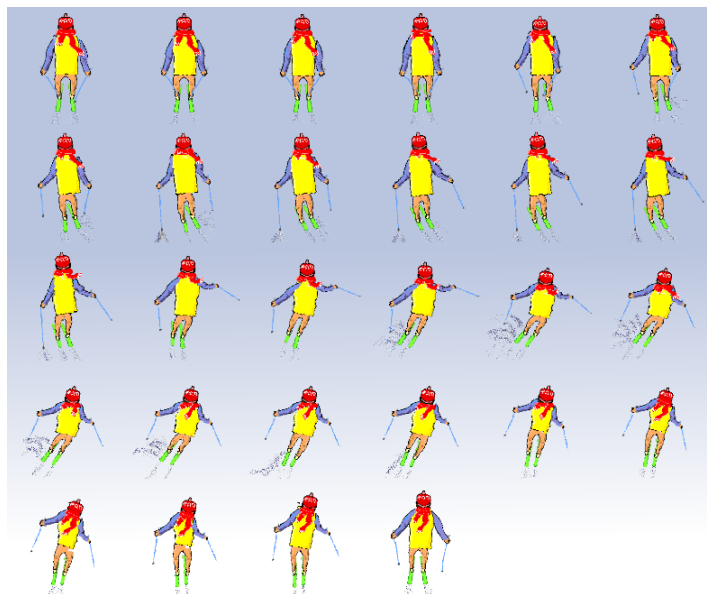


Figure 76: Character animation for the Skiing game

In addition to the top-down view which provides a free form gameplay experience, a side view was implemented as seen in Figure 77 and Figure 78. This gameplay mode the encourages a user to hold a stretch rather than just reach a given threshold. Sample data captured from this game can be seen Appendix B.



Figure 77: Gameplay mode 2 of the Skiing game encouraging to hold a stretch.

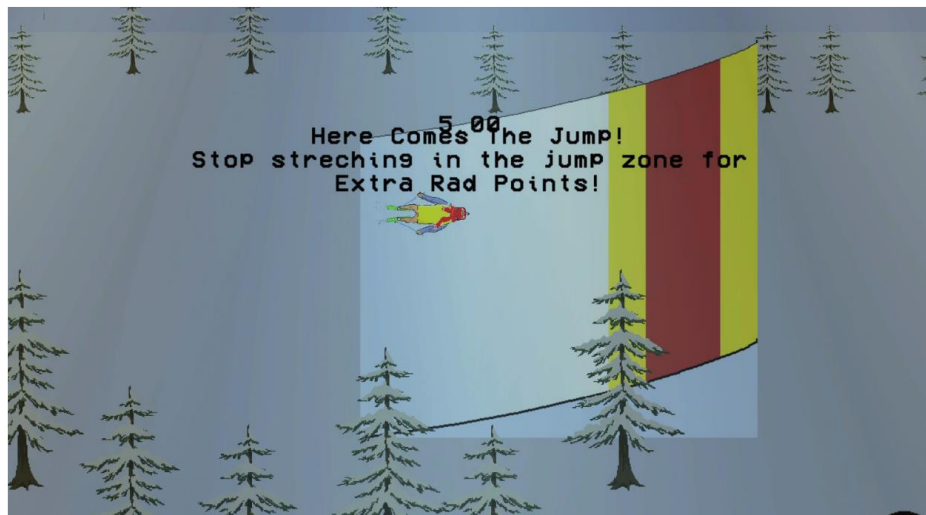


Figure 78: Feedback provided to the user while engaging in gameplay mode 2.

Aeroplane Flying Game

The Aeroplane Flying game provides an example of mapping rehabilitation exercises to hand movement in the 3D space. The objective of this game is fly an Aeroplane through floating loops in the air which can be mapped to different rehabilitation exercises. This game opens with a threshold setting similar to other games developed and provides a customization capability to the user as seen in Figure 79. The main user interface for this game can be seen in Figure 80 with demonstrates multiple characterises such as the amount of time left, score, orientation angles, visual

representation of the hand, and calibration functionality. Figure 81, Figure 82, and Figure 83 show the Aeroplane game in different feedback modes.



Figure 79: Threshold setting for the Aeroplane flying game.



Figure 80: Main feedback window for the Aeroplane flying game



Figure 81: Aeroplane flying game in Pitch mode



Figure 82: Aeroplane flying game in Yaw mode

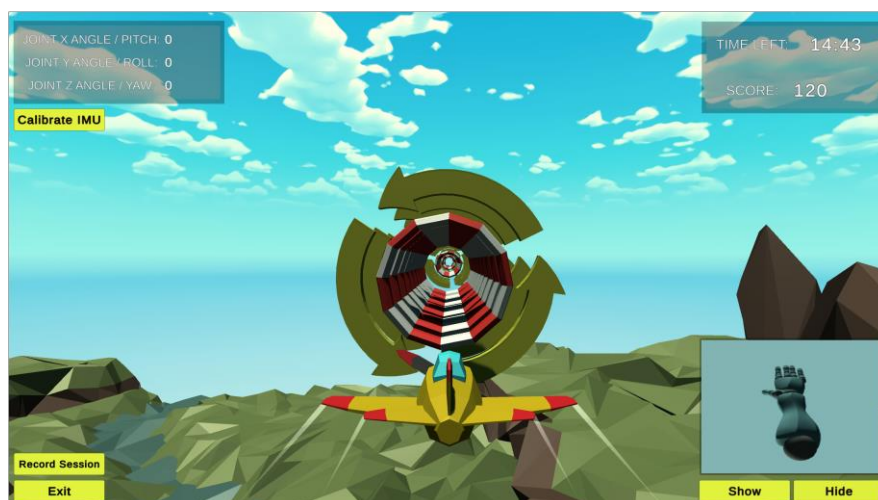


Figure 83: Aeroplane flying game in Roll mode.

To implement the control of the Aeroplane module in the game, the ControlPlane() function as seen in Figure 84 was developed which also included the capability of controlling the plane with mouse and keyboard for debugging purposes. Additional debugging controls were implemented using the MoveObjectTest.cs script as seen in Figure 85 to test the IMU data as an input for the game.

```

/**
Method: ControlPlane
Import: None
Export: None
Assertion: Uses cursor to control the pitch and yaw of the plane through transformations. The keyboard entries 'Q' and 'E' are used for left and right roll, respectively.
*/
void ControlPlane()
{
    // Record where the mouse is
    lookInput.x = Input.mousePosition.x;
    lookInput.y = Input.mousePosition.y;

    // Get the mouse distance
    mouseDistance.x = (lookInput.x - screenCenter.x) / screenCenter.x;
    mouseDistance.y = (lookInput.y - screenCenter.y) / screenCenter.y;

    // Restrict mouse distance to if in length to achieve the same rotational speed in all directions
    mouseDistance = Vector2.ClampMagnitude(mouseDistance, 1f);

    // Roll input
    rollInput = Mathf.Lerp(rollInput, Input.GetAxisRaw("Roll"), rollAcceleration * Time.deltaTime);

    // Rotate ship using the mouse distance
    //transform.Rotate(-mouseDistance.y * lookRateSpeed * Time.deltaTime, mouseDistance.x * lookRateSpeed * Time.deltaTime, 0f, Space.Self);
    transform.Rotate(-mouseDistance.y * lookRateSpeed * Time.deltaTime, mouseDistance.x * lookRateSpeed * Time.deltaTime, rollInput * rollSpeed * Time.deltaTime, Space.Self);

    // Controlling the ship
    //activeForwardSpeed = Mathf.Lerp(activeForwardSpeed, Input.GetAxisRaw("Vertical") * forwardSpeed, forwardAcceleration * Time.deltaTime);
    activeForwardSpeed = 25f;
    activeStrafeSpeed = Mathf.Lerp(activeStrafeSpeed, Input.GetAxisRaw("Horizontal") * strafeSpeed, strafeAcceleration * Time.deltaTime);
    activeHoverSpeed = Mathf.Lerp(activeHoverSpeed, Input.GetAxisRaw("Hover") * hoverSpeed, hoverAcceleration * Time.deltaTime);

    transform.position += transform.forward * activeForwardSpeed * Time.deltaTime;
    transform.position += (transform.right * activeStrafeSpeed * Time.deltaTime) + (transform.up * activeHoverSpeed * Time.deltaTime);
}

```

Figure 84: Script used for the main flying control of Aeroplane Flying game.

```

void Update()
{
    //Move Lower Component

    //Up
    if (Input.GetKey(KeyCode.I))
    {
        currentLowerComponent.x -= rotationSpeed * Time.deltaTime;
    }

    //Down
    if (Input.GetKey(KeyCode.M))
    {
        currentLowerComponent.x += rotationSpeed * Time.deltaTime;
    }

    //Left
    if (Input.GetKey(KeyCode.J))
    {
        currentLowerComponent.y -= rotationSpeed * Time.deltaTime;
    }

    //Right
    if (Input.GetKey(KeyCode.K))
    {
        currentLowerComponent.y += rotationSpeed * Time.deltaTime;
    }

    //Rotate CW
    if (Input.GetKey(KeyCode.L))
    {
        currentLowerComponent.z -= rotationSpeed * Time.deltaTime;
    }

    //Rotate CCW
    if (Input.GetKey(KeyCode.H))
    {
        currentLowerComponent.z += rotationSpeed * Time.deltaTime;
    }
}

```

Figure 85: Script for testing the IMU input for Aeroplane flying game.

Finally, after implementing the debugging functionality, the script seen in Figure 86 was implemented to map the IMU data directly to the Aeroplane movement by implementing the ControlPlane() function within the IMUcontroller.cs script.

```

/**
Method: ControlPlane
Import: None
Export: None
Assertion: Uses the two components (hand and arm) to obtain the joint angles and control the plane based on threshold Limits.
The plane will only turn in the desired direction once the threshold is met.
*/
public void ControlPlane()
{
    angleVector = JointAngle(upperComponent.rotation, lowerComponent.rotation);

    // These angles are based on the XYZ coordinate direction of the upper and lower components in the Unity environment.
    pitchControl = angleVector.x; // Flexion (negative) and Extension (Positive)
    yawControl = angleVector.z; // Radial (negative) and Ulnar (positive) deviation
    rollControl = angleVector.y; // Supination (negative) and Pronation (Positive)

    /*** PITCH CONTROL ***/
    if(pitchControl >= extensionThresh)
    {
        transform.Rotate(-lookRateSpeed * Time.deltaTime, 0, 0);
    }
    else if(pitchControl <= flexionThresh)
    {
        transform.Rotate(lookRateSpeed * Time.deltaTime, 0, 0);
    }

    /*** YAW CONTROL ***/
    if(yawControl >= ulnarDevThresh)
    {
        transform.Rotate(0, lookRateSpeed * Time.deltaTime, 0);
    }
    else if(yawControl <= radDevThresh)
    {
        transform.Rotate(0, -lookRateSpeed * Time.deltaTime, 0);
    }

    /*** ROLL CONTROL ***/
    if(rollControl >= pronateThresh)
    {
        transform.Rotate(0, 0, lookRateSpeed * Time.deltaTime * rollSpeed);
    }
    else if(rollControl <= supinateThresh)
    {
        transform.Rotate(0, 0, -lookRateSpeed * Time.deltaTime * rollSpeed);
    }

    // Controlling the ship
    //activeForwardSpeed = Mathf.Lerp(activeForwardSpeed, Input.GetAxisRaw("Vertical") * forwardSpeed, forwardAcceleration * Time.deltaTime);
    activeForwardSpeed = forwardSpeed;
    activeStrafeSpeed = Mathf.Lerp(activeStrafeSpeed, Input.GetAxisRaw("Horizontal") * strafeSpeed, strafeAcceleration * Time.deltaTime);
    activeHoverSpeed = Mathf.Lerp(activeHoverSpeed, Input.GetAxisRaw("Hover") * hoverSpeed, hoverAcceleration * Time.deltaTime);

    transform.position += transform.forward * activeForwardSpeed * Time.deltaTime;
    transform.position += (transform.right * activeStrafeSpeed * Time.deltaTime) + (transform.up * activeHoverSpeed * Time.deltaTime);
}

```

Figure 86: Implementation of the ControlPlane() function within IMUController.cs

After successfully implementing the control functions for the Aeroplane, SaveThreshold.cs and LoadThreshold.cs scripts were created to store threshold information into the PlayerPrefs class in Unity. The implementation of this script can be seen in Figure 87 and Figure 88. The next step was to implement the HoopCollect.cs script seen in Figure 89. This script is attached to the invisible portion of the inner diameters of each loop to capture the collision between plane object and the invisible portion. This process then determines the scoring system.

```

/**
Method: saveThresholds
Import: None
Export: None
Assertion: Obtains the threshold values set using the slides on the settings page through the text value, then saves these values to the player's preferences.
*/
public void saveThresholds()
{
    flexionThresh = float.Parse(flexionText.text);
    extensionThresh = float.Parse(extensionText.text);
    supinateThresh = float.Parse(supinateText.text);
    pronateThresh = float.Parse(pronateText.text);
    radDevThresh = float.Parse(radDevText.text);
    ulnarDevThresh = float.Parse(ulnarDevText.text);

    PlayerPrefs.SetFloat("flexion", flexionThresh);
    PlayerPrefs.SetFloat("extension", extensionThresh);
    PlayerPrefs.SetFloat("supinate", supinateThresh);
    PlayerPrefs.SetFloat("pronate", pronateThresh);
    PlayerPrefs.SetFloat("radial", radDevThresh);
    PlayerPrefs.SetFloat("ulnar", ulnarDevThresh);

    // Print to screen
    StartCoroutine(sendNotification("Threshold Angles Saved!", 2));
    Debug.Log("Saved Settings");
}

```

Figure 87: The script for saveThresholds() function

```

/**
Method: Start
Import: None
Export: None
Assertion: Called before the first frame update. Retrieve each saved threshold angles.
*/
void Start()
{
    flexionValue.value = PlayerPrefs.GetFloat("flexion", -30);
    extensionValue.value = PlayerPrefs.GetFloat("extension", 30);
    supinateValue.value = PlayerPrefs.GetFloat("supinate", -30);
    pronateValue.value = PlayerPrefs.GetFloat("pronate", 30);
    radDevValue.value = PlayerPrefs.GetFloat("radial", -30);
    ulnarDevValue.value = PlayerPrefs.GetFloat("ulnar", 30);
}

```

Figure 88: Script for setting starting positions for the Aeroplane Flying game.

```

/**
Method: OnTriggerEnter
Import: other (Collider)
Export: None
Assertion: Called when the plane enters the hoops trigger. An audio will be played as well as the score counter being increased.
*/
void OnTriggerEnter(Collider other)
{
    // Play audio on trigger
    this.gameObject.GetComponent().Play();
    // Count score on trigger
    ScoreCounter.scoreCount += 10;
}

/**
Method: OnTriggerExit
Import: other (Collider)
Export: None
Assertion: Called when the plane exits the hoops trigger. The hoop will be destroyed.
*/
void OnTriggerExit(Collider other)
{
    // Hide Mesh Texture
    gameObject.GetComponent<MeshRenderer>().enabled = false;

    // Destroy object at end of audio play
    Destroy(gameObject, gameObject.GetComponent<AudioSource>().clip.length);
}

```

Figure 89: Implementing the scoring system for the Aeroplane Flying Game

The GameManagerScript.cs was developed to manage all the in-game graphical user interfaces. This script starts a countdown timer to the user can position their hand before starting the exercise. The details of this implementation can be seen in Figure 90 and Figure 91. Sample data captured from this game can be seen in Appendix B.

```
/**
 * Method: Start
 * Import: None
 * Export: None
 * Assertion: Called before the first frame update.
 *            Pauses game on start, starts the countdown timer. The game over canvas is disabled while the in game canvas is enabled.
 */
public void Start()
{
    // Pause game
    Time.timeScale = 0;
    // Start countdown
    StartCoroutine(CountdownToStart());

    gameOverCanvas.SetActive(false);
    inGameCanvas.SetActive(true);
}

/**
 * Method: CountdownToStart
 * Import: None
 * Export: None
 * Assertion: Countdown from set time and pauses the game for duration. The countdown text is visualised in the GUI.
 *            Once timer is over the game will unpause and start. Users will also be prompted to keep their palm face down.
 */
IEnumerator CountdownToStart()
{
    while(countdownTimeOne > 0)
    {
        countdownDisplay.text = "" + countdownTimeOne;
        countdownMsg.text = "Keep palm face down!";
        yield return new WaitForSecondsRealtime(1f);
        countdownTimeOne--;
    }

    countdownDisplay.text = "GO!";
    countdownMsg.text = "";

    // Un-pause game
    Time.timeScale = 1;
    yield return new WaitForSecondsRealtime(1f);
    countdownDisplay.text = "";
}
}
```

Figure 90: The GameManagerScript for Aeroplane flying game

```
/**
 * Method: GameOver
 * Import: None
 * Export: None
 * Assertion: Changes from ingame canvas to gameover canvas. Also pauses the game.
 */
public void GameOver()
{
    gameOverCanvas.SetActive(true);
    inGameCanvas.SetActive(false);
    // Pause game
    Time.timeScale = 0;
}

/**
 * Method: ReturnToMenu
 * Import: None
 * Export: None
 * Assertion: Load scene menu
 */
public void ReturnToMenu()
{
    SceneManager.LoadScene(0);
}
}
```

Figure 91: Game over screen implementation for the Aeroplane flying game

3.6 SUMMARY OF THE EXPERIMENT DESIGN METHODS

This chapter demonstrated the design methodology for development of the framework in such a way that it can be applied to different types of rehabilitation requirements. IMUs were selected as the primary method of capturing simple human joints such as wrist, elbow, and knee. Additionally, HGR facilities of MediaPipe were utilised to measuring finger joints as an example of monitoring a complex joint for rehabilitation purposes. The benefit of this approach was the fact that IMUs are low cost and MediaPipe utilises data captured from simple RGB cameras found in laptops and modern smart phones which means the processing power requirement are very low. This chapter also covered the mathematics behind measure both angles and smoothness measurement during the rehabilitation exercises sessions.

As it was observed, the developed framework provides the means for a user to conduct rehabilitation exercises in a home environment and receive feedback on how accurately the exercises are being performed. The data accumulated during the exercise sessions can be provided to the clinicians so performance overtime can be monitored. The patient data is presented using abstract avatars that do not include any personal information or images. This also helps lower the bandwidth requirements by transmitting movement data only rather than images to the clinician. This chapter also described the testing procedures and methods that were used to ensure clinical accuracy of the captured data. The result of these tests as well as qualitative analysis of the use of the framework via a focus group with experts in the field has been provided in Chapter 5.

A detailed discussion of the design was implementation of human computer interface elements of the framework was provided including visualisation methods, animations, and audiovisual feedback systems. Moreover, the chapter provided details on using game design methodology to create a digital gamified experience for existing rehabilitation exercises. This was achieved by providing design guidelines for development of several game archetypes such that both clinicians and patients can gain benefit when engaging with the framework. This design approach allows patients to get feedback on how accurately they are engaging with the rehabilitation exercises and encourage them to continue the program. The clinicians benefit by being able to remotely monitor and review patients' progress through the rehabilitation program by

viewing the reports provided by the HMI. The Next chapter provides more use case examples for the developed framework.

Chapter 4: Use Case Examples

4.1 SELECTION CRITERIA FOR USE CASES

This chapter provides multiple use cases where different elements of the framework have been utilised. The content of this chapter has been covered in the author's publications related to this thesis as seen in [14], [103], [16], and [43]. Please note that Author contribution statement is available in Appendix A. Some parts of the following paper have been covered in Chapter 2 and Chapter 3 with references provided when necessary. The papers that will be covered in this chapter are:

- 1) **Khaksar, S.**, H. Pan, B. Borazjani, I. Murray, H. Himanshu, W. Liu, C. Elliott, C. Imms, A. Campbell, and C. Walmsley. 2021. "Application of Inertial Measurement Units and Machine Learning Classification in Cerebral Palsy: Randomized Controlled Trial." *JMIR Rehabilitation and Assistive Technologies* 8 (4)
- 2) Sabah Al-azzawi, S., **Khaksar, S.**, E. Khdhair Hadi, H. Himanshu, and I. Murray. 2021. "HeadUp: A Low-Cost Solution for Tracking Head Movement of Children with Cerebral Palsy Using IMU." *MDPI Sensors* 21 (23)
- 3) **Khaksar, S.**, S. Pieters, B. Borazjani, J. Hyde, H. Booker, A. Khokhar, I. Murray, and A. Campbell. 2022. "Posture Monitoring and Correction Exercises for Workers in Hostile Environments Utilizing Non-Invasive Sensors: Algorithm Development and Validation." *MDPI Sensors* 22 (24,9618)
- 4) **Khaksar, S.** L.Checker, B.Borazjani, I.Murray, 2023 "Design and Evaluation of an Alternative Control for a Quad-Rotor Drone using Hand Gesture Recognition" *MDPI Sensors* 2023

Each paper demonstrated different use cases by providing different scenarios where various elements of framework have been utilised. All papers have been submitted in open access journals so the reader of this thesis can refer to the full text of papers if they are interested in more detailed representation of the use cases. In selecting these use cases, the following criteria was followed:

1. All data acquisition methods including both hardware and software elements need to be clinically validated before being selected. Details of this validation process will be discussed in Chapter 5.
2. The interaction methods for each use case should not require expensive equipment and high processing power.
3. The use case should not rely heavily on a given sensor brand so the interaction method can be done using any equipment manufacturer as long as the required specifications are met.

4.2 USE CASE IN CLASSIFICATION OF MOVEMENT ASSOCIATED WITH CEREBRAL PALSY CASE 1

Original Paper

Application of Inertial Measurement Units and Machine Learning Classification in Cerebral Palsy: Randomized Controlled Trial

Siavash Khaksar¹, BSc, MScEng; Huizhu Pan¹, BA, MPhil; Bitu Borazjani¹, BSc; Iain Murray¹, BEng (Hons), DPhil; Himanshu Agrawal¹, DPhil; Wanquan Liu², DPhil; Catherine Elliott³, DPhil; Christine Imms⁴, DPhil; Amity Campbell³, BSc, DPhil; Corrin Walmsley³, DPhil

¹School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Bentley, Australia

²School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, China

³School of Allied Health, Curtin University, Bentley, Australia

⁴The University of Melbourne, Melbourne, Australia

Corresponding Author:

Siavash Khaksar, BSc, MScEng
School of Electrical Engineering, Computing and Mathematical Sciences
Curtin University
Kent Street
Bentley, 6102
Australia
Phone: 61 89266 ext 1592
Email: siavash.khaksar@curtin.edu.au

Abstract

Background: Cerebral palsy (CP) is a physical disability that affects movement and posture. Approximately 17 million people worldwide and 34,000 people in Australia are living with CP. In clinical and kinematic research, goniometers and inclinometers are the most commonly used clinical tools to measure joint angles and positions in children with CP.

Objective: This paper presents collaborative research between the School of Electrical Engineering, Computing and Mathematical Sciences at Curtin University and a team of clinicians in a multicenter randomized controlled trial involving children with CP. This study aims to develop a digital solution for mass data collection using inertial measurement units (IMUs) and the application of machine learning (ML) to classify the movement features associated with CP to determine the effectiveness of therapy. The results were calculated without the need to measure Euler, quaternion, and joint measurement calculation, reducing the time required to classify the data.

Methods: Custom IMUs were developed to record the usual wrist movements of participants in 2 age groups. The first age group consisted of participants approaching 3 years of age, and the second age group consisted of participants approaching 15 years of age. Both groups consisted of participants with and without CP. The IMU data were used to calculate the joint angle of the wrist movement and determine the range of motion. A total of 9 different ML algorithms were used to classify the movement features associated with CP. This classification can also confirm if the current treatment (in this case, the use of wrist extension) is effective.

Results: Upon completion of the project, the wrist joint angle was successfully calculated and validated against Vicon motion capture. In addition, the CP movement was classified as a feature using ML on raw IMU data. The Random Forrester algorithm achieved the highest accuracy of 87.75% for the age range approaching 15 years, and C4.5 decision tree achieved the highest accuracy of 89.39% for the age range approaching 3 years.

Conclusions: Anecdotal feedback from Minimising Impairment Trial researchers was positive about the potential for IMUs to contribute accurate data about active range of motion, especially in children, for whom goniometric methods are challenging. There may also be potential to use IMUs for continued monitoring of hand movements throughout the day.

Trial Registration: Australian New Zealand Clinical Trials Registry (ANZCTR) ACTRN12614001276640, <https://www.anzctr.org.au/Trial/Registration/TrialReview.aspx?id=367398>; ANZCTR ACTRN12614001275651, <https://www.anzctr.org.au/Trial/Registration/TrialReview.aspx?id=367422>

(JMIR Rehabil Assist Technol 2021;8(4):e29769) doi: 10.2196/29769

<https://rehab.jmir.org/2021/4/e29769>

JMIR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 1
(page number not for citation purposes)

XSL-FO
RenderX

KEYWORDS

inertial measurement unit; wearable sensors; biomedical sensors; machine learning; human joint measurement; occupational therapy; range of motion; wearable; sensor; children; cerebral palsy; therapy; disability;

Introduction**Background**

Cerebral palsy (CP) is a condition that affects a person's ability to move [1,2]. It occurs as a result of injury to the developing brain during pregnancy or a short time after birth [3]. CP presents with different characteristics in different people, as the damage to the brain is not identical in every person [1]. The movement difficulties experienced by people with CP are divided into three main categories: spastic motor type, in which muscles appear stiff and tight (most common); dyskinetic type, which involves involuntary movement patterns; and ataxic type, which involves uncoordinated muscle movements that can affect balance and sense of positioning in space [3,4]. The level of severity and combination of symptoms can differ from person to person [5]. For example, one person could have weakness in one hand, which can lead to difficulty in writing or tying shoelaces, whereas another person may have little control over their movement or speech because CP can also affect the person's ability to coordinate the muscles around the mouth and tongue [5].

There are many different clinical classification systems for upper limb function in children with CP with different levels of complexity. In a review by McConnell et al [6], 18 different clinical classification systems were identified and reviewed according to whether they classified function or deformity and by considering the quality of psychometric evidence for each method. These methods were rated based on the clinical utility of each system using previously published tools [6]. An example of clinical classification system is House [7] classification, which contains four categories of thumb deformities. Another example of clinical classification is that by Green and Banks [8], which contains four subgroups of poor, fair, good, and excellent based on the use of the hand by the individual with CP. These classification methods demonstrate the complexity of clinical classification of hand movement in children with CP and the diverse approaches taken to achieve it.

As of early 2021, there is no single method for completely curing or preventing CP. Public health measures such as mandatory seatbelts, pool fencing, and rubella vaccinations are among the prevention methods currently in use [9]. Physiotherapy and occupational therapy focus on encouraging a person's day-to-day movement skills and abilities, such as sitting, walking, dressing, and toileting, and use a range of specialist interventions such as movement and goal-directed training and provision of equipments, such as walking frames, wheelchairs, supportive seating, footwear, and orthotics [9]. When studying children with CP, range of motion, which is the capability of a joint to go through its complete spectrum of movement, may become a crucial component of research. Passive range of motion can be defined as the range of motion when an external force causes movement of the joint and is the maximum range of motion, whereas active range can be

achieved when opposing muscles contract and relax, resulting in child- or person-initiated joint movement [10].

Occupational therapists use upper limb orthoses for children with CP who have muscle overactivity caused by spasticity, but there is little evidence of the long-term effects of these methods [11]. The clinical rationale is that the orthoses help preserve the range of movement; however, they are complex to construct, expensive, and can cause discomfort for the children wearing them [11]. To address the need for robust evidence, a multicenter randomized controlled trial (RCT) is being used to evaluate the effectiveness of wrist hand orthoses to prevent loss of range of movement in children with CP (see *Experiment Setup and Data Collection* for details). This RCT used inertial measurement units (IMUs) to measure active movement in children with CP, to address two measurement problems: (1) the complex movement patterns of children with CP make it difficult for therapists to accurately apply typical clinical measures, such as a goniometer (an instrument that measures the available range of motion at a joint) and (2) young children's small hands and difficulty following detailed movement instructions make it difficult to achieve reliable measurements.

Existing Methods

General movement assessment is used, which is a noninvasive and cost-effective method for identifying babies at risk of CP [12]. This assessment is done by recording a 3- to 5-minute video of an awake infant lying on their back while they were calm and alert without the presence of toys and pacifiers. Parents can be present and record the video, but they should not interact with their babies. This video is then observed and assessed by trained health professionals to detect signs of the disorder [3,12]. This process becomes easier when infants grow older, as they can follow the instructions of the medical professionals to perform different tasks so that their movement can be monitored. This assessment is mainly used as a diagnostic tool for the early detection of CP, and it is not used to quantify the range of movement or motion.

In clinical research, the goniometer and inclinometer are used to measure joint angles in children with CP [13]. A goniometer is an instrument that measures the joint angle, and depending on the nature of the experiment, it can measure the available range of motion at a joint. It can be used to monitor changes in joint angles in clinical settings [14]. The traditional method of using angle-measuring tools is not accurate and reliable, according to some recent studies [13]. Accurately measuring range of motion (ROM) is an important part of clinical assessment as this information is used to guide treatment plans, determine treatment efficacy, and monitor individual's response to treatment [15]. Goniometric measures rely on the ability of the clinician to accurately palpate bony landmarks and visually estimate the alignment of the axis and arms of the goniometer to the joint that is being measured. Goniometers are versatile, reliable, and widely used, irrespective of their measurement errors of up to 15 degrees. However, for active movement, the

<https://rehab.jmir.org/2021/4/e29769>

JMIR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 2
(page number not for citation purposes)

XSL-FO
RenderX

use of goniometers is very difficult, and their use may not be possible in populations that are unable to respond to instructions reliably [15].

A general approach for capturing movement is the use of digital technologies, such as motion capture. Motion capture (also referred to as mo-cap or mocap) is the process of digitally recording the movement of people [16]. It is used in entertainment, sports, medical applications, ergonomics, and robotics. In filmmaking and game development, it refers to the recording actions of actors for animations or visual effects. It is also referred to as performance capture when it includes a full body, face, and fingers or captures subtle expressions [16]. The equipment required for motion capture is extremely costly and is not commonly available in a typical hospital; for example, according to Thewlis et al [17], a simple Vicon system [18] cost approximately Aus \$250,000 (US \$268,605.52) in 2011 [17]. Even if the equipment is available, it may be difficult to take children to these motion analysis laboratories to conduct measurements. Another limitation is the need for expert staff to run the laboratories for the motion analysis of hand movement.

Another approach is to measure gesture control using electronic sensors, such as infrared (IR) light-emitting diodes. Gesture recognition software for advanced smartphones was presented in the paper found in the study by Kong et al [19]. The leap motion sensor uses IR sensors to scan finger movements with a typical field of view of $140^{\circ} \times 120^{\circ}$ [20]. This method is mostly applied in the entertainment industry, so it does not meet the need for accuracy in capturing the movement of people with CP.

With the development of inertial sensor technologies, IMU-based motion capture systems have been introduced in the study of human motion. IMUs comprise an accelerometer, gyroscope, and magnetometer that are connected to a microcontroller and can be used to capture orientation. In recent years, there have been several IMU-based motion capture research studies, such as studies of gait modulation in patients with foot drop problems [21] and human activity recognition using thigh angle derived from a single thigh-mounted IMU data [2]. The use of IMUs for hand movement in free space is currently underdeveloped, primarily due to the lack of a clear calibration reset point compared with gait analysis. Another benefit of IMU solutions is flexibility in the collection window. From a practical point of view, the data measured during any session using motion capture technologies or any nonportable devices that require the patient to be at a certain location at a certain time, which may not be a period when certain movement characteristics are present or typical. For example, the patient could be *having a good day* or fatigued coincidentally during the clinic visit. IMU measurements outside the predefined time may avoid errors in the data collection. In addition, patient's compliance would potentially increase in the case of children, where their movement is taking place in their home environment compared with organized clinic visits. The challenge would then be to filter a larger data set to remove outliers, which is already a problem even when clinicians are involved. Therefore, the IMU data collection needs to be streamlined so that data can be captured easily without any need for clinical or technical expertise.

An overview of all the relevant existing methods, including their advantages and disadvantages can be seen in Table 1.

Table 1. Evaluation of existing methods.

Type of approach	Advantages	Disadvantages
Goniometers [14]	<ul style="list-style-type: none"> • Low cost • Can provide measurements very quickly 	<ul style="list-style-type: none"> • Lack of accuracy • Does not provide long-term tracking of movement unless repeated multiple times • Difficult when children are involved
Video capture [16]	<ul style="list-style-type: none"> • Very accurate • Can provide real-time orientation and active movement 	<ul style="list-style-type: none"> • Very costly • Continued monitoring is not possible outside the motion capture studio • Long set up time • Facilities are not available to everyone
IR ^a LED ^b gesture recognition [20]	<ul style="list-style-type: none"> • Low cost • Portable 	<ul style="list-style-type: none"> • Lack of accuracy • Not possible for continued monitoring • Mostly developed for entertainment use
IMU ^c [22]	<ul style="list-style-type: none"> • Low cost • Can provide a reasonably accurate orientation frame • Low power consumption • Portable 	<ul style="list-style-type: none"> • IMUs drift over time • The postprocessing of IMU data can be lengthy

^aIR: infrared radio.

^bLED: light-emitting diode.

^cIMU: inertial measurement unit.

<https://rehabjmhir.org/2021/4/e29769>

JMIR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 3
(page number not for citation purposes)

XSL-FO
RenderX

Contribution of the Paper

This paper presents collaborative research between the Department of Electrical Engineering and Computing at Curtin University and the investigator team of a multicenter RCT involving children with CP [11]. The novelty of this work is the mass data collection and application area of the sensor system. To achieve this goal, 2 small, low-cost, custom-built IMUs were developed to capture the hand movements of participants in 2 age groups. The first age group had participants approaching 3 years, and the second age group had participants approaching 15 years. Both groups comprised participants with and without CP. Custom sensors were needed because commercial sensors are costly and do not provide raw sensor data. This means that validation cannot be performed easily. In addition, the use of custom sensors will avoid preprocessing by a third party. The designed sensors were capable of measuring wrist joint movement as the angle difference between 2 parallel sensors, which simplifies a 3D system problem to a 2D one. Therefore, only the relative motion was used, and the impact of the environment was ignored. This approach facilitates a reliable and valid method to capture changes over time. Capturing ROM over time is important because children with CP have secondary musculoskeletal complications, which means they are at risk of losing movement range. The proposed low-cost sensor system could also provide the means for active and continuous tracking of wrist joint movement during usual or predetermined tasks and actions that are currently not possible using traditional goniometric methods.

A second contribution of this paper is the application of ML to raw IMU data to classify the movement features associated with CP without the need to measure Euler, quaternion, and joint measurement calculations. This means that the processing time will be reduced because of using raw data for classification. This classification aims to investigate the existence of characteristics of CP movement, which is different from the clinical classification used for CP as a condition. This

classification can also confirm if treatment (in this case, the use of wrist extension) is effective. After the initial data collection, 9 different ML algorithms were used to classify CP as a feature: the Random Forest algorithm achieved the highest accuracy of 87.75% for the age range approaching 15 years, and C4.5 decision tree achieved the highest accuracy with 89.39% for the age range approaching 3 years. The result of this classification aligns with existing research work in which ML is applied to classify footdrop using IMUs [23]. The results of this project showed that decision tree-based ML algorithms were the most accurate compared with other methods, which could be used as a guideline for similar human joint measurements.

Methods

Sensor Development

A custom-built IMU was developed to capture the hand movements of children with CP for this project. The IMU consisted of an MPU 9250, a custom-built Arduino Pro Mini, and a 2.4-GHz radio frequency (RF) radio. Each sensor was powered by a small 90 mAh, 3.7-V rechargeable lithium battery and could support up to 3 hours of nonstop measurement. The custom Arduino Pro Mini was previously designed by Dr Weiyang Xu as part of his thesis titled *Design and Validation of a Portable Wireless Data Acquisition System for Measuring Human Joint Angles in Medical Applications* [24]. The IMU data were captured using a simple receiver dongle that used an RF radio transceiver connected to an Arduino Uno and was read from the serial communication link. Both RF modules were connected using a serial peripheral interface (SPI), and the IMU was connected using an interintegrated circuit (I²C) connection. The designed IMU is shown in Figures 1 and 2. A summary of the specifications of the IMU is presented in Table 2. These sensors were validated against a goniometer and Vicon motion capture system, the results of which can be found in the studies by Walmsley et al [15] and Xu et al [25].

Figure 1. The MPU9150 (blue printed circuit board [PCB]), custom-built Arduino Pro Mini (green PCB), and RF Module (red PCB); a comparison of the inertial measurement unit with an Australian five-cent coin; and the 3D printed case for the sensor [24].



Figure 2. The receiver dongle in the 3D printed case [24].



Table 2. Specification of the inertial measurement unit (IMU).

Electronic Module	Parameter	Value
MPU 9250 IMU	<ul style="list-style-type: none"> Accelerometer FS range Gyroscope FS range Magnetometer FS range 	<ul style="list-style-type: none"> Range of ± 2 g, ± 4 g, ± 8 g and ± 16 g Range of ± 250, ± 500, ± 1000 and $\pm 2000^\circ/\text{sec}$ Range of $\pm 1200 \mu\text{T}$
nRF24L01 Transceiver	<ul style="list-style-type: none"> ISM^a band operation Air data rate Programmable output power 	<ul style="list-style-type: none"> 2.4GHz 250 kbps, 1 and 2 Mbps 0, -6, -12 or -18 dBm
Arduino Pro mini	<ul style="list-style-type: none"> Circuit operating voltage Clock Speed Flash memory 	<ul style="list-style-type: none"> 3.3 V or 5 V 8 MHz (3.3 V version) or 16 MHz (5 V version) 32 KB
Arduino Uno	<ul style="list-style-type: none"> Circuit operating voltage Clock Speed Flash memory 	<ul style="list-style-type: none"> 5 V 16 MHz 32 KB

^aISM: Industrial, Scientific, and Medical.

<https://rehab.jmir.org/2021/4/e29769>

JMIR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 5
(page number not for citation purposes)

XSL-FO
RenderX

The SPI is a synchronous, full-duplex serial bus standard that was introduced by Motorola to support communication between a master processor and multiple slaves [26]. This protocol used Serial Clock sent by the master to synchronize master and slave; Serial Data Out to stream from the device; Serial Date In to stream into the device; Slave Select to enable slave, which is omitted in point-to-point connotations [26]. The master-slave connection for the RF module is shown in Figure 3. SPI was used to connect the RF module to the custom build module, where Arduino was the master and the RF module was the slave. The same connection was used on the receiver to connect the

RF module and Arduino Uno with the Arduino acting as the master and the RF module acting as the slave. This decision was made because of the inclusion of Master In Slave Out and Master Out Slave In data lines that facilitate full-duplex communication, a fast communication speed that can go to 10 Mbps or more; inclusion of push-pull drivers that provide good signal integrity, not limited to 8-bit words for bits transferred; use of master's clock by the slave, which removed the need for precision oscillators, and lower power requirements compared with other serial buses because of less circuitry.

Figure 3. The left diagram shows the serial peripheral interface (SPI) connection between Arduino Uno and the RF module, and the right diagram shows the SPI connection between the custom Arduino Pro mini and the RF module. MISO: Master In Slave Out; MOSI: Master Out Slave In; RF: radio-frequency; SCLK: Serial Clock.



The designed sensors needed to wirelessly transfer data to avoid hindering the hand movements of the participants in the project. Popular wireless communication technologies include Bluetooth, RF, WiFi, and infrared. The popular frequency range for wireless communication includes subGHz below 1 GHz (for long-range) and 2.4 GHz (for short-range). The proposed joint movement calculation system uses an nRF24L01 RF transceiver [27] (transmitter-receiver integrated on the same chip) module, which operates on a 2.4-GHz frequency band using 125 channels in the frequency range of 2.4 GHz-2.525 GHz. The module uses a license-free industrial, scientific, and medical frequency and can cover a distance of up to 1000 m. To improve the data loss at this crowded frequency band around 2.4 GHz, the nRF24L01 RF transceiver module uses a low noise amplifier [27]. The data rate requirement of the proposed joint movement calculation is not very high. This RF transceiver module is an improvement as it supports data rates in the range of 250 kbps-2 Mbps. The RF transceiver module connects with the Arduino module using SPI through Serial Clock, Master In Slave Out, and Master Out Slave In pins. The nRF24L01 RF transceiver is an ultralow

power drawing of 26 μ A of current in standby mode and 900 nA of current under down mode [27].

The I²C bus is a synchronous serial protocol originally developed by Philips Semiconductor (now known as NXP semiconductors) in the early 1980s [26]. The main aim of I²C was originally to support the board-level interconnection of ID modules and peripherals [26]. This protocol used serial data, and Serial Clock and ground for a half-duplex connection, which is capable of handling multiple masters and slaves. Serial Clock synchronizes all bus transfers, and serial data carries the data being transferred [26]. The connection of the MPU 9250 module is shown in Figure 4. The structure of the timing diagram for I²C is shown in Figure 5. The I²C was used to connect the IMU module to the custom-built IMU, with the Arduino acting as the master and the IMU acting as the slave. This decision was made because of the incorporation of Acknowledgment and No Acknowledgment functionality that improves error handling, flexible data transmission rates, addressability of each devices bus, and requiring only 2 signal lines.

Figure 4. Schematic of the I2C connection between the custom Arduino Pro Mini and the inertial measurement unit. SCL: Serial Clock Line; SDA: Serial Data Line.



Figure 5. I2C timing diagram. SCL: Serial Clock Line; SDA: Serial Data Line.



The IMUs comprise an accelerometer, gyroscope, and magnetometer. Using sensor fusion techniques, an object's orientation can be captured using differential equations describing its dynamic behavior, which can be derived from the Newton-Euler by means of the Euler angle parametrization [28]. Quaternion is another method for capturing the orientation of an object, which is a four-element vector that can be used to encode any rotation in a 3D coordinate system [28]. In this study, to simplify calculations, raw acceleration and angular velocity were captured and used to measure the wrist joint angle. The requirements and specifications of this research lead to the selection of IMUs owing to their low cost, low power consumption, and ability to provide orientation with the relevant update rate.

Joint Angle Calculations

The sensors collected raw acceleration and angular velocity in the X-, Y-, and Z-axes, and the results were postprocessed in MATLAB using a 2-sensor-based joint orientation algorithm. This algorithm shows the difference in relative movements between 2 sensors when they share the same frame and zero position [24]. The Z- and Y-axes of both sensors need to be parallel to each other, so the X-axis of both sensors merge into the wrist center. This means that the wrist joint movement can be measured as the angle difference between the 2 sensors. The use of 2 parallel sensors for joint calculation simplifies the 3D system problem to a 2D one. The orientation of the MPU9250 is shown in Figure 6 [29] and the placement of the sensors is shown in Figure 7.

Figure 6. Orientation of the MPU9250 inertial measurement unit chip, where X is Roll, Y is Pitch, and Z is Yaw [29].

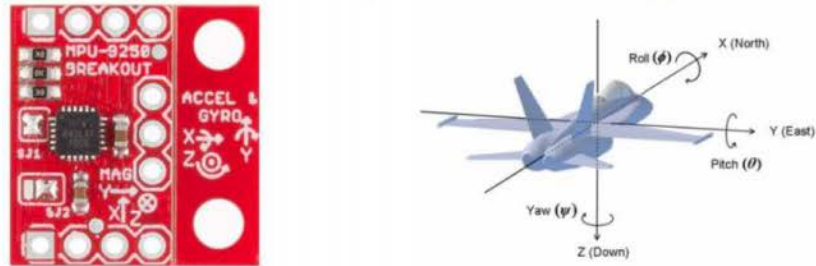


Figure 7. Sensor placement showing sensor 1 connected to the back of the hand and sensor 2 connected above the wrist.



Using 2 sensors creates a relative system, so the rotation on the Y-axis or the orientation on the X-Z plane can simply be calculated using the following formula:

$$\angle \beta_{eff} = \angle \beta_{sensor 1} - \angle \beta_{sensor 2} \tag{equation (1)}$$

According to the tangent function, the angle of β can be initially calculated using the acceleration from the X- and Z-axes, where α is the angle between the net acceleration and the acceleration on the X-Z plane. Therefore, the tangent of β can be calculated as follows:

<https://rehab.jmir.org/2021/4/e29769>

JMIR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 7
(page number not for citation purposes)

XSL-FO
RenderX

$$\tan(\beta) = \frac{A_z}{A_x} = \frac{A_{xz} \sin(\theta)}{A_{xz} \cos(\theta)} = \frac{A_{net} \cos(\gamma) \sin(\beta)}{A_{net} \cos(\gamma) \cos(\beta)} \quad \text{equation (2)}$$

The angles used in equation (2) can be seen in Figure 8.

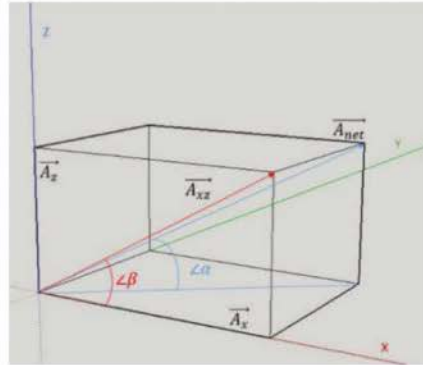
The data sample rate for both sensors was set to 100 Hz, which reduced the difference in angular velocity measurements between each sample.

Unlike traditional yaw, pitch, and roll orientation systems, a reference plane was unnecessary in the present algorithm as

both sensor axes were aligned so that the joint movement was equivalent to the orientation difference between the sensors. Therefore, only relative motion was used, and the impact from the environment was ignored [24].

The orientation of each individual sensor was calculated using the orientation reading and angle movement during each sampling period, and a complementary filter introduced a high-pass filter to the main orientation tracker and adjusted with a low passed value from the accelerometer's orientation measurement [24].

Figure 8. 3D system for acceleration.



As the desired accuracy cannot be achieved by using only the acceleration, sensor fusion was used to increase the measurement accuracy by combining the data from both the accelerometer and the gyroscope. The accelerometer output was independent of each sample during the measurement period; therefore, θ_{xz} , θ_{yz} , and θ_{zy} , which are the projected orientation angles on the X-Z, Y-Z, and Z-Y planes, respectively, were used as rough measurements. The gyroscope's angular velocity ω_{gr} was added to describe the actual change between samples and can be calculated after subtracting the average static drift and using a Savitzky-Golay filter to calibrate the moving average drift [24]. The gyroscope's angular velocity can be calculated using the following formula:

$$\omega_{gr}(t) = 0.5[\omega_{gr}(t) + 0.5\omega_{gr}(t+1) + \omega_{gr}(t-1)] - 0.5[\omega_{gr}(t+2) + \omega_{gr}(t-2)] - D_i \quad \text{equation (3)}$$

Here, D_i is the average static drift, which can be calculated using the following equation:

$$\bar{\omega}_{gr} = \frac{1}{m} \left[\sum_{i=1}^{n-1} \omega_{gr}(i) \right] + \left(\sum_{i=1}^{n-1} \omega_{gr}(i) \right) \frac{1}{m} + \left(\sum_{i=1}^{n-1} \omega_{gr}(i) \right) \frac{1}{m} \quad \text{equation (4)}$$

In the formula given above, n, m, and r are random integers and m is larger than 3. The total number of samples needs to be larger than $n + (m-1)r + 100$ m. These calculations lead to the following sensor fusion algorithm, which is based on a complementary filter:

$$\sigma_c(n+1) = k[\sigma_c(n) + \omega_{gr}(n+1) \times \Delta t(n)] + (1-\sigma_c(n+1)) \quad \text{equation (5)}$$

where a, b, and c are the names of the measurement axes and n+1 is the current order of the sample. $\sigma_c(n+1)$ is the filtered

angle along the c-axis. Therefore, ω_{gr} represents the rotation on the c-axis, and θ_{ab} is the current angle on the a-b plane, which is based on accelerometer measurements. Finally, the combination of high pas factor h and low pas factor l is 1 [24].

The results of these joint calculations were validated in the study by Sharif Bidabadi et al [30] against a 3D Vicon video capture setup. The accuracy of the setup was written in a different paper found in the study by Walmsley et al [15], where a custom-made robotic device with predetermined angles was designed, where the sensors detected peak angles with mean errors ranging from -0.95° to 0.11° when one wearable sensor was static and the other dynamic. When 2 wearable sensors were moving, movement at a higher speed ($90^\circ/s$) had a mean error range of -2.63° to 0.54° and movement at a slower speed ($30^\circ/s$) had a mean error range of -0.92° to 2.90° [15].

Data Preprocessing

The IMU sensors generated time-series data from the accelerometer, gyroscope, and magnetometer around the 3 axes. First, small sections were removed from readings taken at the beginning of the experiments when the IMU sensors were not stabilized. Then, the remaining data collected by each sensor from each experiment in 3 orientations (ie, pitch, row, and yaw) were converted into frequency-domain representations by performing fast Fourier transform. Converting data to the frequency domain can successfully capture the characteristics of gait motion, as shown by similar experiments in [23,31,32], the interval between adjacent readings was approximated as 0.1 seconds and the fundamental frequency was calculated as $1/t_{total}$.

where t_{total} is the total time of the experiment. The amplitude A , phase shift P , and peak frequency F of the first 5 harmonics were collected into a feature vector. The feature vector for each experiment was 1×270 , and the 270 features were as follows:

$$270 \text{ features} = \left(\frac{\text{frequency domain}}{f_{1st}, f_{2nd}, f_{3rd}, f_{4th}, f_{5th}} \right) \times \left(\frac{\text{amplitude}}{A_{1st}, A_{2nd}, A_{3rd}, A_{4th}, A_{5th}} \right) \times \left(\frac{\text{phase}}{P_{1st}, P_{2nd}, P_{3rd}, P_{4th}, P_{5th}} \right) \times \left(\frac{\text{time domain}}{t_{1st}, t_{2nd}, t_{3rd}, t_{4th}, t_{5th}} \right) \quad \text{equation (6)}$$

Each experiment was then labeled 0 for a typically developing child and 1 for a child with CP.

Classification by ML Algorithms

The problem of distinguishing typical hand movements from hand movements of children with CP constitutes a binary classification problem, that is, classification between two classes. Various algorithms can be constructed using different ML methods based on existing data that can be used to classify unseen data. This process is called *training*. Some classical ML algorithms commonly used in engineering problems include linear classifiers such as Naïve Bayes and logistic regression, decision trees such as the C4.5 decision tree and random forest, support vector machine, k-nearest neighbors, and neural networks such as multilayer perceptron and convolutional neural networks. More sophisticated deep neural networks can also be designed for classification problems; however, the size of training data sets is a major concern. Other problems include data bias, overfitting, a lack of computational resources, etc.

To decide between the 2 classes, ML algorithms for binary classification establish decision boundaries that separate the data points in the training data set from the 2 classes. This process relies on optimizing a cost function that varies between the algorithms. Most algorithms, such as logistic regression, support vector machine, decision trees, and neural networks, aim to construct a model with parameters that are learned from the training data set, whereas some algorithms operate directly on the data set, for example, k-nearest neighbors. Although there are numerous libraries and tools offering implementations of ML algorithms [33,34], the performance of the individual algorithm depends on the nature of the problem and the properties of the data set. Choosing the algorithm that performs best for a particular problem is subject to investigation.

Experiment Setup and Data Collection

As a part of an Australia-wide CP research study called the *Minimising Impairment Trial* (MIT) and *Infant Wrist Hand Orthosis Trial* (iWHOTs), the IMU sensors were used to capture the wrist movements of 2 groups of participants. The MIT trial included children with and without CP aged 5-15 years, and the iWHOT included children aged 6 months to 3 years. These studies were multisite RCTs that aimed to evaluate whether long-term use of rigid wrist or hand orthoses in children with CP, combined with usual multidisciplinary care, could prevent or reduce musculoskeletal impairments, including muscle

stiffness or tone and loss of movement range, compared with usual multidisciplinary care alone [11]. IMUs were used as an outcome measure to capture the active wrist ROM. During each assessment session, the participants completed several wrist movement activities such as making a stop sign motion, picking up small objects, playing with toys, pressing a big button, and so on. The aim of these activities was to assess the ROM used during active movement and task performance while data were collected via sensors. In addition, goniometric measurements of the joint movement was collected. The detailed protocol of this research has been published [11] if the reader is interested in more information about the clinical aspects of this trial.

For this project, the aim was to capture CP movement as a feature by ML on the raw IMU data by focusing on the data collected during the stop sign task in the MIT and iWHOT. Each participant was asked to perform a simple stop sign motion to capture the maximum wrist joint angle as well as the maximum range of movement. To achieve this study's aim, two separate experiments were run using participants who were approaching the age of 3 years from iWHOT and participants who were approaching the age of 15 years from MIT. From MIT, 263 samples from 89 participants with CP and 199 samples of typical movement data captured from 30 participants without CP were used. The participants without CP simulated typical movements to reach 199 samples. From iWHOT, 171 samples from 51 participants with CP and 149 samples from 20 participants without CP were used.

Cross-validation, which is 90% training and 10% testing, were used 10 times to train and test the classifier, which can be seen in the next section of this paper. The CP data were collected by the research teams working on the MIT and iWHOT trial according to ethically approved procedures (HREC REF 201406EP) and with signed, informed consent from all the participants' parents or guardians. Deidentified data were used to produce ML results, which are analyzed in the *Discussion* section of this paper.

Results

Figures 9 and 10 show the raw data captured for a stop sign motion trial of a participant without CP, starting from the stationary position to a stop sign and again to a stationary position. These data included the accelerometer and gyroscope in 3 axes. Figure 11 shows the placement of the sensors on the hand and above the wrist.

After the data were captured, they were processed and run through the different equations described in the joint calculation section of the report. Through these calculations, the drift was removed, and the joint angle was calculated, the results of which are shown in Figure 12.

Figure 9. Raw data captured with the sensor connected to the hand (data without CP). CP: cerebral palsy.

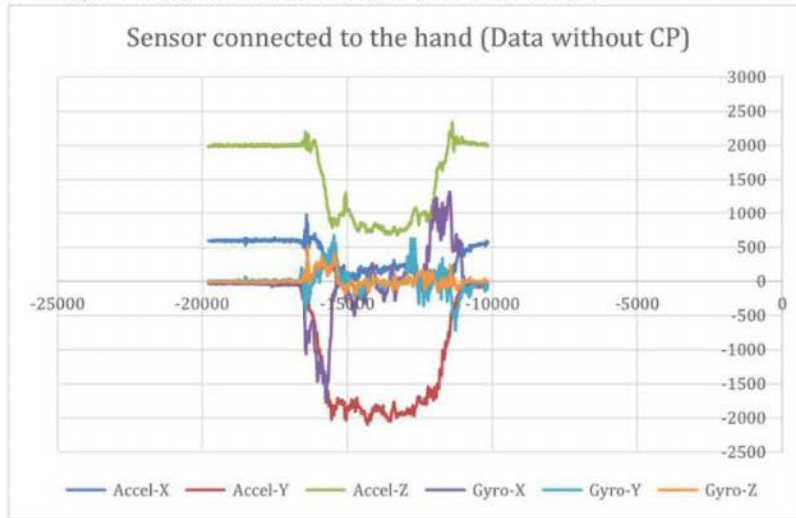


Figure 10. Raw data captured with the sensor connected above the wrist (data without CP). CP: cerebral palsy.

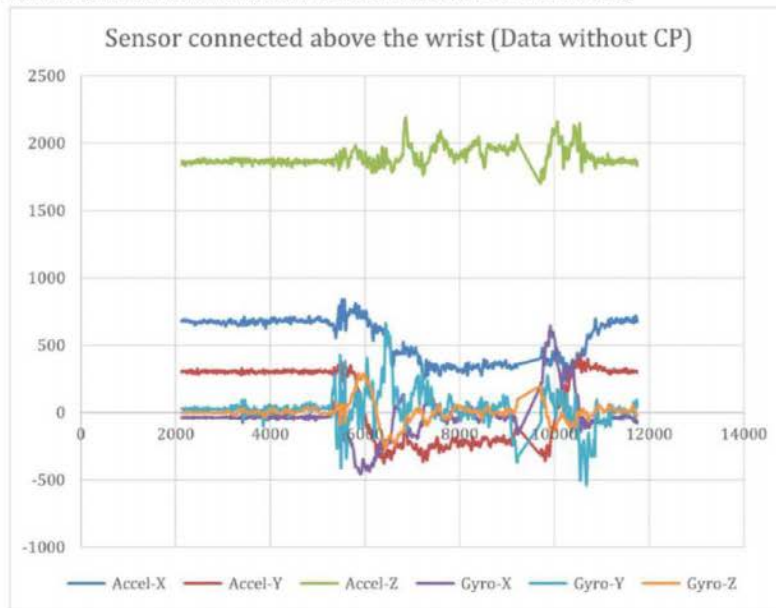
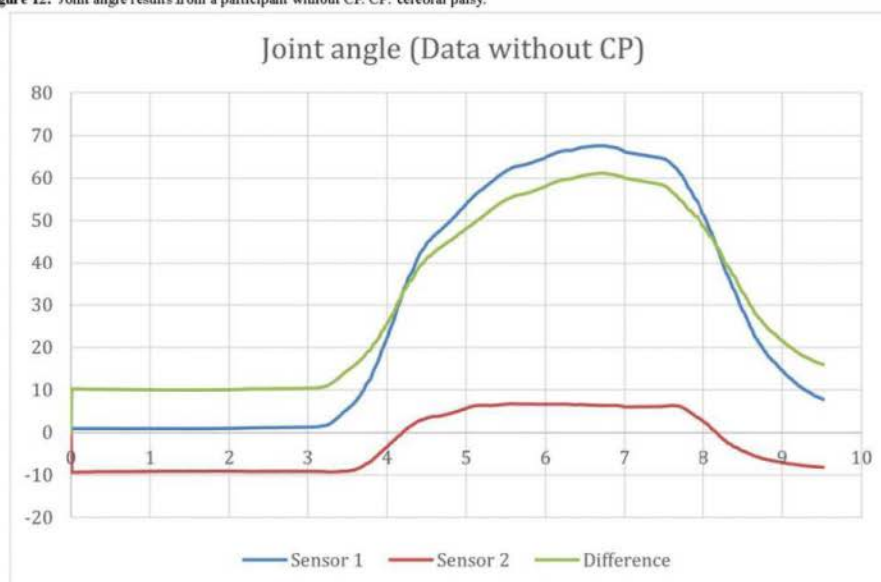


Figure 11. Stop sign motion required by the participants.**Figure 12.** Joint angle results from a participant without CP. CP: cerebral palsy.

The stop sign trials from participants with CP were captured using the same IMUs as those used in the previous group. The results of the raw data captured from the CP participants are shown in Figures 13 and 14. The results of the calculated joint angles are shown in Figure 15.

Anecdotal feedback from MIT and iWHOT researchers was positive about the potential of IMUs to contribute accurate data

about active ROM, especially in children for whom goniometric methods are challenging.

After the initial angles were calculated, several classical ML models were trained to create a classifier for the captured data. The Waikato Environment for Knowledge Analysis platform [34] version 3.8 was chosen as the platform for these experiments. Waikato Environment for Knowledge Analysis is a collection of open-source ML algorithms and contains tools

<https://rehab.jmir.org/2021/4/e29769>

JMIR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 11
(page number not for citation purposes)

XSL-FO
RenderX

for data preparation, classification, regression, clustering, association rule mining, and visualization [34]. The algorithms used consisted of ZeroR, OneR, Bayes Net, Naive Bays, logistic regression, C4.5 decision tree, random forest, support vector

machine, multilayer perceptron, and k-nearest neighbors. The authors analysis of the produced ML results can be found in the *Discussion* section of this paper.

Figure 13. Raw data captured with the sensor connected to the hand (data with CP). CP: cerebral palsy.

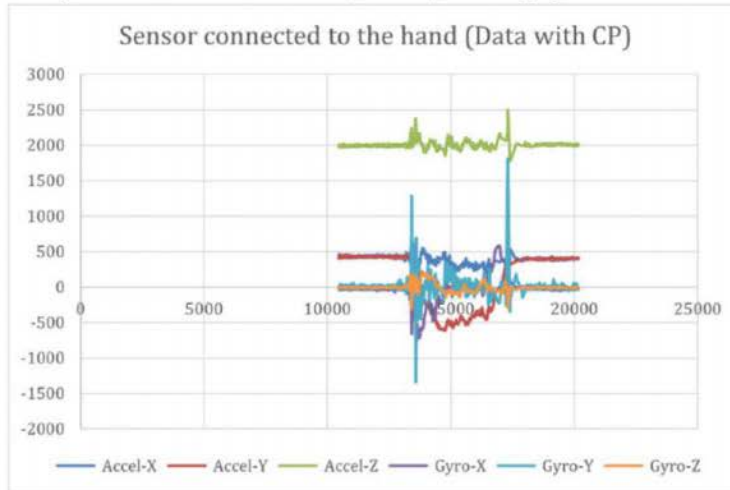


Figure 14. Raw data captured with the sensor connected above the wrist (data with CP). CP: cerebral palsy.

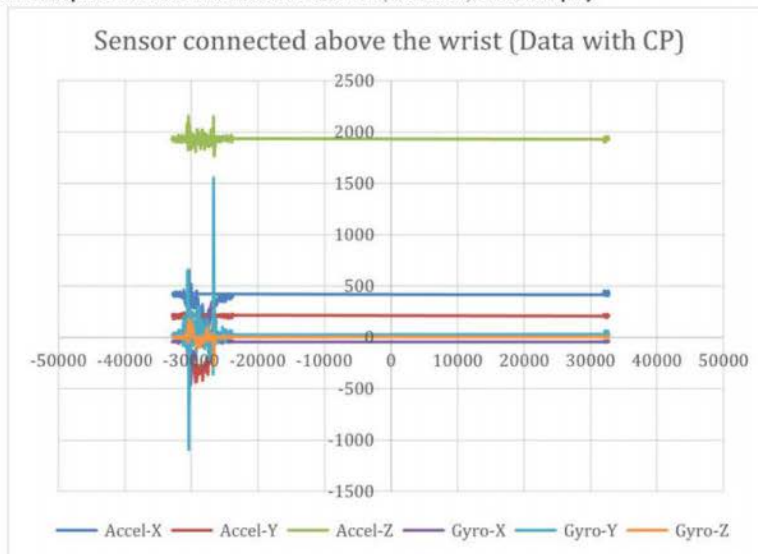
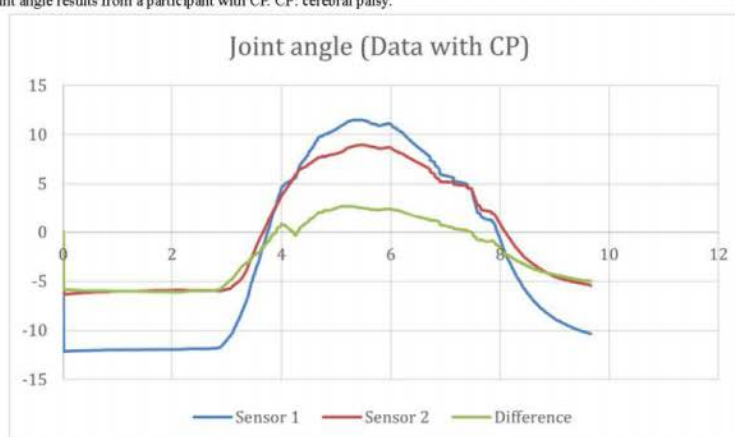


Figure 15. Joint angle results from a participant with CP. CP: cerebral palsy.

Discussion

Principal Findings

The resultant evaluation metrics are accuracy, the number of correctly classified instances over the total number of instances, the area under the curve (AUC), and the area under the receiver operating characteristic (ROC) curve. The ROC curve maps the true positive rates as the x-coordinate and false positive rates as the y-coordinate. Ten-fold cross-validation was adopted, splitting the data set into 10 parts, training the models with 9 parts, and testing with 1 part each time for a total of 10 times. The accuracy and AUC were obtained by averaging the 10 sets of results and taking the weighted average of the 2 classes. The baseline of the experiments was obtained from ZeroR, a

classifier that predicts the class that occurs most often in the training data set as the label without considering other features.

Table 3 presents the results of the 9-ML algorithms on the classification using the MIT data. The baseline obtained from ZeroR showed 57.02% accuracy and 0.493 AUC. The best accuracy was 85.75% yielded by random forest, and the best AUC was 0.890 yielded by k-nearest neighbors. Figure 16 shows the ROC curves of the 9 ML algorithms and the baseline. OneR, k-nearest neighbors, multilayer perception, and random forest all produce reasonable ROC curves and are expected to perform well for the problem. Naive Bayes performs better than the other algorithms owing to the conditional independence assumption it makes. Because the frequency space features are interrelated, it is unreasonable to make this assumption.

Table 3. Machine learning result using minimizing impairment training data, showing the best accuracy.

Algorithm	Accuracy (%)	AUC ^a
OneR	84.23	0.848
Logistic regression	72.79	0.749
Naïve Bayes	65.23	0.752
Bayes Net	80.99	0.832
C4.5 decision tree	74.95	0.740
<i>Random forest</i> ^b	<i>85.75</i>	<i>0.867</i>
Multilayer perceptron	80.35	0.865
Support vector machine	79.70	0.794
K-nearest neighbors	82.07	<i>0.890</i>
<i>Average</i>	<i>78.45</i>	<i>0.815</i>

^aAUC: area under the curve.

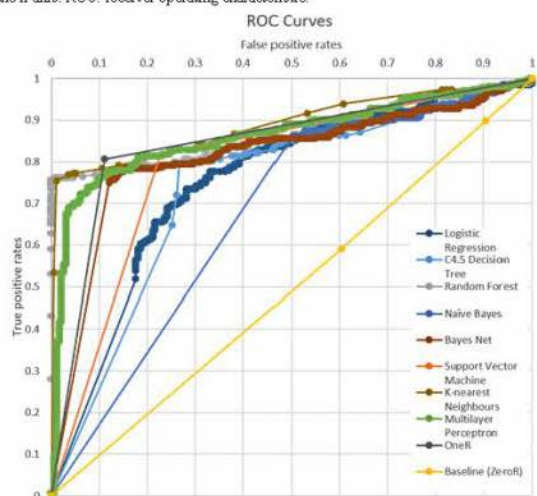
^bThe best accuracy and area under the curve values are italicized.

<https://rehab.jmir.org/2021/4/e29769>

JMIR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 13
(page number not for citation purposes)

XSL-FO
RenderX

Figure 16. The ROC curves of 10 classification algorithms using the Minimizing Impairment Trial data. The area under the curve values are the areas between the ROC curves and the x-axis. ROC: receiver operating characteristic.



Curiously, OneR uses only a single feature and achieves 84.23% classification accuracy. The algorithm uses the 91st feature, which is the phase shift corresponding to the second harmonic obtained from the hand sensor. This phenomenon may indicate that the most useful information for classification is recorded by the hand sensor and that omitting one sensor may be possible in the future.

Table 4 presents the results of the 9-ML algorithms in binary classification using the iWHOT data. The baseline obtained from ZeroR showed 53.44% accuracy and 0.494 AUC. The best accuracy was obtained by the C4.5 decision tree at 85.75%, and the best AUC was obtained by Naive Bayes at 0.890. Figure 17

shows the ROC curves of the 9-ML algorithms plus the baseline. Although all models appear to be reasonable classifiers for the problem, it is worth noting that OneR, which classifies based on one feature alone, already achieves 88.13% accuracy and 0.886 AUC. The deciding feature is the amplitude of the acceleration in the row direction on the hand sensor, which corresponds to the most important piece of information in a real-world scenario. The relative underperformance of the more sophisticated algorithms, in contrast, may be due to the observed noises in the training data that lead to biases in the learned models. Such noises include the sensors falling off the participant, the participant not following instructions, etc.

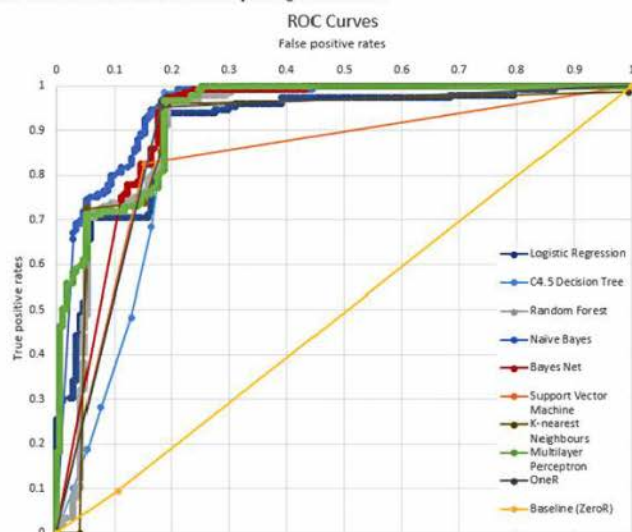
Table 4. Machine learning result using Infant Wrist Hand Orthosis Trial data.

Algorithm	Accuracy (%)	AUC ^a
OneR	88.13	0.886
Logistic regression	80.94	0.906
Naive Bayes	86.88	0.943 ^b
Bayes Net	88.43	0.921
<i>C4.5 decision tree</i>	<i>89.38</i>	0.858
Random forest	81.88	0.917
Multilayer perceptron	81.25	0.937
Support vector machine	83.75	0.783
K-nearest neighbors	83.44	0.896
<i>Average</i>	<i>84.90</i>	<i>0.894</i>

^aAUC: area under the curve.

^bThe best accuracy and area under the curve values are italicized.

Figure 17. The ROC curves of 10 classification algorithms using the Infant Wrist Hand Orthosis Trial data. The area under the curve values are the areas between the ROC curves and the x-axis. ROC: receiver operating characteristic.



Conclusions

Upon completion of the project, the wrist joint angle was successfully calculated, and CP movement was classified as a feature using ML on raw IMU data. Anecdotal positive feedback from MIT and iWHOT researchers was also received regarding the potential for IMUs to contribute accurate data about active ROM, especially where the use of goniometers can be challenging. There may also be the potential to use IMUs for continued monitoring of hand movements throughout the day. The sensor size needs to be reduced to make it more comfortable to wear. Examples of ML and IMU data captured for medical purposes can be seen in the paper titled *Classification of foot drop gait characteristic due to lumbar radiculopathy using machine learning algorithms* [23]. This paper looks at the classification of IMU data captured from hospital patients with foot drop issues using supervised learning and uses 11 different ML classifiers and shows that random forest was the most accurate method with an accuracy of 88.45% for a specific data set [23]. Some of the other ML algorithms used were SVM, Naive Bayes, and deep learning, which gave accuracies of 86.87%, 86.87%, and 86.06%, respectively [14]. Bidabadi et al [30] showed results were very similar to the current findings, although the focus was on a different joint. This suggests that decision-tree-based ML algorithms may be the best option for classifying IMU data for joint movement. The classifier used in this study would be able to distinguish atypical and reduced movement, which can potentially be useful for people with different joint movement disorders such as arthritis and Parkinson disease.

There are some limitations to the IMU setup used in this study, such as the inherent drift of IMUs, which can be corrected by

the drift mitigation techniques described in the methods. These techniques may prove problematic for longer trials. There were other issues during the data collection sessions, such as touching the 2 (hand and forearm) sensors because of the small hands of some participants or accidental touching of the sensors by the therapist while using the goniometer, which leads to an increase in noise in the data. Bugs in the data collection interface created for technicians also resulted in some corrupted data and data loss, which added to the preprocessing time of the ML section of this study. Finally, at the initial stages of the project, the scale of the accelerometer was set at +2 g because the slower moving trials rarely reached this value. Once free play situations were introduced that would usually contain rapid movement, particularly in younger children, it was observed that the scale of g needed to be extended beyond this threshold, which resulted in reduced accuracy. This reduction caused some data loss, so the scale was switched to +16 g for faster trials.

As part of future work, real-time calculation of joint angle and orientation data can be implemented so that direct quaternions can be collected and used for this calculation. The research team involved in this paper began the preliminary work on this next step and plans to publish their results once the solution has been fully created. The sensor setup will also be updated to remove the reliance on a separate receiver dongle by switching the communication module to Bluetooth Low Energy transfer to a smartphone application. These changes to the user experience and the medium of transfer would improve the utility of the process of data collection, better continued monitoring of children with CP, and quicker trial sessions in routine appointments for children with CP.

<https://rehab.jmir.org/2021/4/e29769>

JMR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 15
(page number not for citation purposes)

XSL-FO
RenderX

Acknowledgments

This research received no external funding; however, The Minimising Impairment Trial (MIT) was funded through the Australian Catholic University Research Fund (APP2013000413) and the National Health and Medical Research Council Centre for Research Excellence–Cerebral Palsy Grant number APP1057997. Additional funding was received from the Perth Children’s Hospital Foundation and Monash Health. These trials were registered with the ANZ Clinical Trials Registry (ACTRN12614001276640) and ACTRN12614001275651. The authors would like to acknowledge the guidance provided by Dr Shiva Bidabadi in sensor setup, all the hard work of occupational therapists who were in direct contact with patients with cerebral palsy, especially Ms Sherilyn Nolan, who helped explain the activities during the data collection trials, and Dr Weiyang Xu for his efforts in the validation of sensors. The cerebral palsy data were collected by the research teams working on the MIT trial according to ethically approved procedures (HREC REF 201406EP) and with signed, informed consent from all the participants parents and guardians. The identified data were used to produce the following machine learning results.

Authors' Contributions

Conceptualization of the study was by IM and SK. Data curation was conducted by CE, CI, AC, and CW. Formal analysis was conducted by SK. Investigation was conducted by SK and HA. Methodology was devised by SK and IM. Resources were provided by CE and CI. Software provision was by SK and HP. Supervision was conducted by IM and WL. Validation was carried out by SK. Visualization was conducted by SK and BB. Original draft was written by SK. BB, HA, CE, and CI were involved in reviewing and editing the paper.

Conflicts of Interest

None declared.

Multimedia Appendix 1

CONSORT-eHEALTH checklist (V 1.6.1).

[\[PDF File \(Adobe PDF File\), 1080 KB-Multimedia Appendix 1\]](#)

References

- Rosenbaum P, Paneth N, Leviton A, Goldstein M, Bax M, Damiano D, et al. A report: the definition and classification of cerebral palsy April 2006. *Dev Med Child Neurol Suppl* 2007 Feb;109:8-14. [Medline: [17370477](#)]
- Abhayasinghe N, Murray I. Human activity recognition using thigh angle derived from single thigh mounted IMU data. In: Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN). 2014 Presented at: International Conference on Indoor Positioning and Indoor Navigation (IPIN); Oct. 27-30, 2014; Busan, Korea (South) p. 27-30. [doi: [10.1109/ipin.2014.7275474](#)]
- Novak I, Hines M, Goldsmith S, Barclay R. Clinical prognostic messages from a systematic review on cerebral palsy. *Pediatrics* 2012 Nov 08;130(5):1285-1312. [doi: [10.1542/peds.2012-0924](#)] [Medline: [23045562](#)]
- Novak I. Evidence-based diagnosis, health care, and rehabilitation for children with cerebral palsy. *J Child Neurol* 2014 Aug 22;29(8):1141-1156. [doi: [10.1177/0883073814535503](#)] [Medline: [24958005](#)]
- How does cerebral palsy affect people? Cerebral Palsy Alliance Research Foundation. URL: <https://cerebralpalsy.org.au/our-research/about-cerebral-palsy/what-is-cerebral-palsy/how-cerebral-palsy-affects-people/> [accessed 2020-10-10]
- McConnell K, Johnston L, Kerr C. Upper limb function and deformity in cerebral palsy: a review of classification systems. *Dev Med Child Neurol* 2011 Sep;53(9):799-805 [FREE Full text] [doi: [10.1111/j.1469-8749.2011.03953.x](#)] [Medline: [21434888](#)]
- House JH, Gwathmey FW, Fidler MO. A dynamic approach to the thumb-in palm deformity in cerebral palsy. *J Bone Joint Surg* 1981;63(2):216-225. [doi: [10.2106/00004623-198163020-00006](#)]
- Green WT, Banks HH. Flexor carpi ulnaris transplant and its use in cerebral palsy. *J Bone Joint Surg* 1962;44(7):1343-1430. [doi: [10.2106/00004623-196244070-00006](#)]
- Treatment for cerebral palsy. Cerebral Palsy Alliance Research Foundation. URL: <https://cerebralpalsy.org.au/our-research/about-cerebral-palsy/what-is-cerebral-palsy/interventions/> [accessed 2020-10-10]
- Nelson F, Blauvelt C. 12 - Physical medicine and rehabilitation: physical therapy and occupational therapy. In: *A Manual of Orthopaedic Terminology* (Eighth Edition). Philadelphia: Saunders; 2015:365-375.
- Imms C, Wallen M, Elliott C, Hoare B, Randall M, Greaves S, et al. Minimising impairment: protocol for a multicentre randomised controlled trial of upper limb orthoses for children with cerebral palsy. *BMC Pediatr* 2016 May 27;16(1):70 [FREE Full text] [doi: [10.1186/s12887-016-0608-8](#)] [Medline: [27230616](#)]
- What is the general movements assessment? Cerebral Palsy Alliance Research Foundation. URL: <https://cerebralpalsy.org.au/our-research/about-cerebral-palsy/what-is-cerebral-palsy/signs-and-symptoms-of-cp/general-movements-assessment/> [accessed 2021-04-23]

13. Herrero P, Carrera P, Garcia E, Gómez-Trullén EM, Oliván-Blázquez B. Reliability of goniometric measurements in children with cerebral palsy: a comparative analysis of universal goniometer and electronic inclinometer. A pilot study. *BMC Musculoskelet Disord* 2011 Jul 10;12(1):155 [FREE Full text] [doi: 10.1186/1471-2474-12-155] [Medline: 21740600]
14. Goniometer. Physiopedia. URL: <https://www.physio-pedia.com/Goniometer> [accessed 2020-10-10]
15. Walmsley CP, Xu W, Ortega-Sanchez C, Campbell A, Imms C, Elliott C, et al. Validation of custom wearable sensors to measure angle kinematics: a technical report. *Health Technol* 2019 Sep 09;9(5):887-892. [doi: 10.1007/s12553-019-00360-1]
16. Motion capture? Xsens. URL: <https://www.xsens.com/motion-capture> [accessed 2020-10-28]
17. Thewlis D, Bishop C, Daniell N, Paul G. A comparison of two commercially available motion capture systems for gait analysis: high-end vs low-cost. In: Proceedings of Congress of the International Society of Biomechanics. 2011 Presented at: Congress of the International Society of Biomechanics, July 3-7, 2011; Brussels, Belgium URL: <https://eprints.qut.edu.au/49083/>
18. Vicom products. Vicon Motion Systems. URL: <https://www.vicon.com/> [accessed 2021-09-30]
19. Kong K, Kim Y, Kim J, Kim S, Baek K. Single-package motion gesture sensor for portable applications. *IEEE Trans Consumer Electron* 2013 Nov;59(4):848-853. [doi: 10.1109/tce.2013.6689698]
20. Leap motion controller. UltraLeap. URL: <https://www.ultraLeap.com/product/leap-motion-controller/> [accessed 2020-09-22]
21. Bidabadi SS, Murray I, Lee GY. The clinical application of inertial measurement unit in identification of foot drop symptoms. In: Proceedings of the IEEE 15th Student Conference on Research and Development (SCORED). 2017 Presented at: IEEE 15th Student Conference on Research and Development (SCORED), Dec. 13-14, 2017, Wilayah Persekutuan Putrajaya, Malaysia. [doi: 10.1109/scored.2017.8305397]
22. Xsens overview of IMUs. Xsens. URL: <https://www.xsens.com/inertial-sensor-modules> [accessed 2021-04-23]
23. Bidabadi S, Murray I, Lee GY, Morris S, Tan T. Classification of foot drop gait characteristic due to lumbar radiculopathy using machine learning algorithms. *Gait Posture* 2019 Jun;71:234-240 [FREE Full text] [doi: 10.1016/j.gaitpost.2019.05.010] [Medline: 31082655]
24. Xu W. Design and validation of a portable wireless data acquisition system for measuring human joint angles in medical applications. Thesis and Dissertations - Curtin University. 2018. URL: <https://espace.curtin.edu.au/handle/20.500.11937/76045> [accessed 2021-09-30]
25. Xu W, Ortega-Sanchez C, Murray I. Measuring human joint movement with IMUs: implementation in custom-made low cost wireless sensors. In: Proceedings of the IEEE 15th Student Conference on Research and Development (SCORED). 2017 Presented at: IEEE 15th Student Conference on Research and Development (SCORED), Dec. 13-14, 2017, Wilayah Persekutuan Putrajaya, Malaysia. [doi: 10.1109/scored.2017.8305399]
26. Jiménez M, Palomera R, Couvertier I. Introduction to Embedded Systems: Using Microcontroller the MSP430. New York: Springer; 2014:1-30.
27. Sparkfun Transceiver Breakout nRF24L01. Sparkfun. URL: <https://www.sparkfun.com/products/691> [accessed 2020-10-22]
28. Alaimo A, Artale V, Milazzo C, Ricciardello A. Comparison between Euler and quaternion parametrization in UAV dynamics. *AIP Conference Proceedings* 2013;1558(1):1228-1231. [doi: 10.1063/1.4825732]
29. Understanding quaternions. CH Robotics. URL: <https://sourceforge.net/projects/chrinterface/> [accessed 2020-10-20]
30. Bidabadi SS, Murray I, Lee GY. Validation of foot pitch angle estimation using inertial measurement unit against marker-based optical 3D motion capture system. *Biomed Eng Lett* 2018 Aug 17;8(3):283-290 [FREE Full text] [doi: 10.1007/s13534-018-0072-5] [Medline: 30603212]
31. Yoo JH, Nixon MS, Harris CJ. Model-driven statistical analysis of human gait motion. In: Proceedings of the International Conference on Image Processing. 2002 Presented at: International Conference on Image Processing; Sept. 22-25, 2002; Rochester, NY, USA. [doi: 10.1109/icip.2002.1038015]
32. Zhang J, Lockhart TE, Soangra R. Classifying lower extremity muscle fatigue during walking using machine learning and inertial sensors. *Ann Biomed Eng* 2014 Mar 1;42(3):600-612 [FREE Full text] [doi: 10.1007/s10439-013-0917-0] [Medline: 24081829]
33. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in python. *J Mach Learn Res* 2011;12:2825-2830. [doi: 10.5555/1953048.2078195]
34. Frank E, Hall M, Holmes G, Kirkby R, Pfahringer B, Witten IH, et al. Weka - A machine learning workbench for data mining. In: Maimon O, Rokach L, editors. *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer; 2010:1269-1277.

Abbreviations

- AUC:** area under the curve
CP: cerebral palsy
IC: interintegrated circuit
IMU: inertial measurement unit
IR: infrared
iWHOT: Infant Wrist Hand Orthosis Trial

<https://rehab.jmir.org/2021/4/e29769>

JMIR Rehabil Assist Technol 2021 | vol. 8 | iss. 4 | e29769 | p. 17
 (page number not for citation purposes)

XSL-FO
 RenderX

MIT: Minimising Impairment Trial
ML: machine learning
RCT: randomized controlled trial
RF: radio frequency
ROC: receiver operating characteristic
ROM: range of motion
SPI: serial peripheral interface
Weka: Waikato Environment for Knowledge Analysis

Edited by G Eysenbach, submitted 20.04.21; peer-reviewed by T McLennox, V van Hees; comments to author 11.05.21; revised version received 26.07.21; accepted 19.09.21; published 20.10.21

Please cite as:

*Khaksar S, Pan H, Borazjani B, Murray I, Agrawal H, Liu W, Elliott C, Imms C, Campbell A, Walmsley C
Application of Inertial Measurement Units and Machine Learning Classification in Cerebral Palsy: Randomized Controlled Trial
JMIR Rehabil Assist Technol 2021;5(4):e29769
URL: <https://rehab.jmir.org/2021/4/e29769>
doi: [10.2196/29769](https://doi.org/10.2196/29769)
PMID:*

©Siavash Khaksar, Huizhu Pan, Bitu Borazjani, Iain Murray, Himanshu Agrawal, Wanquan Liu, Catherine Elliott, Christine Imms, Amity Campbell, Corrin Walmsley. Originally published in JMIR Rehabilitation and Assistive Technology (<https://rehab.jmir.org/>), 20.10.2021. This is an open-access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work, first published in JMIR Rehabilitation and Assistive Technology, is properly cited. The complete bibliographic information, a link to the original publication on <https://rehab.jmir.org/>, as well as this copyright and license information must be included.

4.3 USE CASE IN CLASSIFICATION OF MOVEMENT ASSOCIATED WITH CEREBRAL PALSY CASE 2



Article

HeadUp: A Low-Cost Solution for Tracking Head Movement of Children with Cerebral Palsy Using IMU

Sana Sabah Al-azzawi ^{1,2,*}, Siavash Khaksar ³, Emad Khedhair Hadi ⁴, Himanshu Agrawal ³ and Iain Murray ³

¹ SKT Department, EISLAB, Luleå University of Technology, 97187 Luleå, Sweden

² College of Engineering, University of Information Technology and Communications, Baghdad 10013, Iraq

³ School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University,

Bentley, WA 6102, Australia; siavash.khaksar@curtin.edu.au (S.K.);

himanshu.himanshu@curtin.edu.au (H.A.); i.murray@curtin.edu.au (I.M.)

⁴ Rehabilitation Medical Center and Joint Diseases, Baghdad 10001, Iraq; mrc_baghdad@yahoo.com

* Correspondence: sana.sabah.sabry.al-azzawi@ltu.se or sana.sabah@uoitic.edu.iq



Citation: Al-azzawi, S.S.; Khaksar, S.; Hadi, E.K.; Agrawal, H.; Murray, I. HeadUp: A Low-Cost Solution for Tracking Head Movement of Children with Cerebral Palsy Using IMU. *Sensors* **2021**, *21*, 8148. <https://doi.org/10.3390/s21238148>

Academic Editor: Hossam A. Cabbar

Received: 14 October 2021

Accepted: 2 December 2021

Published: 6 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Cerebral palsy (CP) is a common reason for human motor ability limitations caused before birth, through infancy or early childhood. Poor head control is one of the most important problems in children with level IV CP and level V CP, which can affect many aspects of children's lives. The current visual assessment method for measuring head control ability and cervical range of motion (CROM) lacks accuracy and reliability. In this paper, a HeadUp system that is based on a low-cost, 9-axis, inertial measurement unit (IMU) is proposed to capture and evaluate the head control ability for children with CP. The proposed system wirelessly measures CROM in frontal, sagittal, and transverse planes during ordinary life activities. The system is designed to provide real-time, bidirectional communication with an Euler-based, sensor fusion algorithm (SFA) to estimate the head orientation and its control ability tracking. The experimental results for the proposed SFA show high accuracy in noise reduction with faster system response. The system is clinically tested on five typically developing children and five children with CP (age range: 2–5 years). The proposed HeadUp system can be implemented as a head control trainer in an entertaining way to motivate the child with CP to keep their head up.

Keywords: head control; cerebral palsy; rehabilitation; inertial measurement unit; head movement measurement; health; disability; sensor fusion algorithm

1. Introduction

Cerebral palsy (CP) is a group of disorders in the developmental milestone, including posture and motor function, that become evident through infancy or early childhood [1,2]. The CP prevalence ranges from 1.5 to more than 4 per 1000 live births. These statistics are expected to be much higher in developing countries due to low standards of medical care [3].

CP is a non-progressive brain disorder, and most children with CP experience spasticity, motor disorders, and a lack of selective motor control [4]. Although CP is permanent, its outcomes can be minimized [5]. These motor difficulties differ from one CP child to another CP child based on the severity level. The Gross Motor Function Classification System (GMFCS)[6] classifies children with CP into five levels, as shown in Figure 1.

Children with CP levels IV-V suffer from one of the essential cerebral palsy problems: poor head control [7].

Head control refers to the ability to control the head upright above the shoulder with respect to gravity while sitting, standing, or walking with a rotation ability in the desired direction. Head stability gives a stable reference to vertical posture. Additionally,

poor head control can affect many aspects of children's lives, such as eating, self-care, self-entertainment, vocational sitting [8,9], and self-esteem [10].

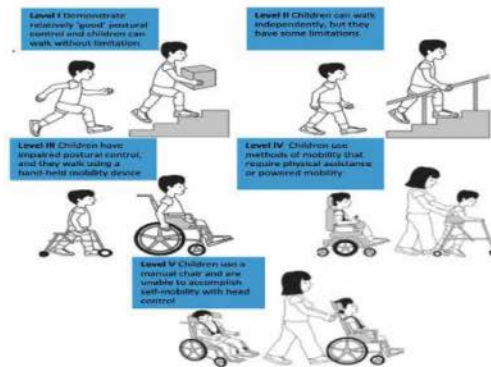


Figure 1. Cerebral palsy levels.

Multiple reasons make the assessment of head motion and head control ability (HCA) a vital study area in the rehabilitation and bioengineering speciality. The most significant reason is helping clinicians evaluate, diagnose, and provide optimal care and treatment for children with CP [11,12]. Moreover, difficulties encountered while preserving head control can be one of the first indications that a child has a development problem.

However, the current diagnosis for poor head control depends on visual inspection [13], which relies on the clinician's experience and the time he has spent with a patient. Sometimes, clinicians and physiotherapists cannot spend an adequate amount of time assessing HCA. Although parents spend much time with their children, they still cannot provide continuous monitoring for head movement. There is also the problem that people living in remote areas do not have access to medical experts.

Additionally, different studies reported poor accuracy and reliability of visual assessment for joint movement. These studies stated the need for more accurate and reliable tools combined with a visual evaluation to improve examination quality [14,15]. As a result, there is a need for a precise system that can capture head movement and discover the inability to fully control the vertical orientation of the head. There is also a need to provide the best physiotherapy program, to avoid the implications of inaccurate diagnoses and to fully utilize this program.

Inertial measurement unit (IMU) sensors can capture an object's motion without the need for external reference, cameras, emitters or environmental lighting [16]. The IMU consists of a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer. There are many clinical applications for IMUs in human movement monitoring [17–22].

This paper presents a novel, IMU-based system for HCA tracking that targets children with CP aged from 2 to 5 years old. HeadUp is a head-mounted device that is designed to be small ($3 \times 3 \times 3 \times 6.5$ cm), that is easy to wear, that is lightweight (<50 g), and that consumes low power. Additionally, the device's raw data are available and could be accessed easily for future analysis to identify the required physiotherapy program. Moreover, the proposed device is designed to be inexpensive in contrast to other existing systems and specifically for head movement monitoring. The most important challenge is extracting medical information from the raw system's readings to assess the HCA of the child and this study will address this challenge in Section 6.

The main contribution of this paper is the use of a custom built and low-cost sensor for capturing head movement of children with CP in the age range of 2 to 5 years. Most of the research in the literature target older children (5–12 years) with either higher CP levels

such as level I or II, where the child already has better motor ability and reasonably good head control [23–25]. The paper shows the application of a low-cost engineering solution to capture clinically viable data which allows clinical professionals and physiotherapists to get a better understanding of the active range of head movement for children with CP. Anecdotal feedback from the physiotherapists has been positive since there are very limited methods for capturing the active range of movement, especially for the age range of 2 to 5 years.

The remainder of this study has been constructed as follows: related work is presented in Section 2. System design is described in Section 3. This section includes the HeadUp device design and the filter algorithm to filter the measured data's noise. System implementation is explained in Section 4. Section 5 presents the results and Section 6 discusses the results. Section 7 concludes the paper and presents future work.

2. Related Work

Head movement or cervical movement is the ability to smoothly and accurately move the head to a given pattern. There are several possible methods to measure head movement. Head motion measurement systems or cervical range of motion (CROM) have been gradually adopted in the medical profession for many purposes. Different methods have been employed to measure head motion [26].

An ancient study was conducted in 1962 [27], in which the author reviewed the various methods for joint motion measurement in general and head motion measurement in particular, such as protractors with arms, optical goniometers, and pendulum goniometers. These measurements made it difficult to accurately determine head motion. Many recent studies have discussed more accurate and trustworthy methods for head movement measurement [28–32].

One of the best and most accurate methods is an optical motion capture system [32] that uses cameras and optoelectronic markers to track head movement. Although this method is convenient, such systems are easily affected by occlusion and lighting effects, which tend to reduce the system efficiency. Additionally, such systems are costly and complex to implement in laboratories [33]. Another issue is the difficulty of a testing environment for small children.

Different systems have been proposed to overcome these limitations. IMUs showed excellent implementation in the field of three-dimensional motion analysis [17,34–36]. In [36], Rudigkeit et al. compared different commercially available IMU-based systems under various conditions to control an object by head movement. The authors reviewed the control model presented in the literature and highlighted the advantages and disadvantages of each method. All previously mentioned methods can be an effective way to measure the head control ability HCA of children with cerebral palsy.

Many researchers have investigated some of these methods for training children with CP to achieve better head control [11,37,38], especially between 1970 and 1990 [7,39–42]. Harris et al. [7] developed an electronic device that can be placed in an oversized helmet to monitor head movement while converting this movement to electrical signals, providing auditory guidance to achieve an upright position. The authors tested the effectiveness of the head control device (HCD) on nine children with CP, with ages ranging from 7 to 18; all of these children improved their head stability after using the device from approximately a few seconds to more than 5 min in duration of holding a fixed posture. Harris et al. did not provide any statistical details about the enhancement after treatments, and the degree of improvement for each patient is unknown.

Subsequently, mercury tilt switches were employed to detect the head deviation from the specified angle range [39–41]. In [39], a mercury tilt switch was installed on an earphone to turn a transistor radio ON and OFF to reinforce head posture for two children with CP whose ages ranged from 9 and 17 years old. Radio music was activated when the child's head was in the upright position. The authors discovered that music can promote head control for children with CP. A wearable head position trainer (HPT) was designed

in [40] and tested on 12 CP children aged 3–10 years. HPT provided auditory feedback and number count of head position deviations beyond a given angle for each child. A large number of researchers have applied the HPT in their studies [41–43].

Although these studies stated an improvement in HCA, many drawbacks and limitations in their devices are noted [44]. Some of these disadvantages are described as follows:

- These devices were heavy (220 g without the helmet), and some of the children experienced difficulty raising their head while wearing the apparatus compared with those without wearing the device.
- Cables must connect these devices to a control unit, increasing the size of the overall system.
- These devices did not provide any information about head movement in 3 dimensions.
- The presence of a physiotherapist is mandatory during the experiments.
- Difficulty in reliable device positioning occurred because of the working principle of mercury tilt switching.
- A deviation angle threshold for each child had to be established and applied every time the device was used.

Unfortunately, there are few recent studies concerning head movement and head control ability tracking for children with CP. In 2018, head motion was measured for children with CP using a video-based approach. The authors placed markers on the ear and temporal fossa, using cameras to record head movement. This approach has a beneficial effect on IICA assessment [11]. The same approach was employed [38] to estimate the head orientation. Even though this approach solves the previously mentioned limitations and difficulties, it is costly and difficult to set by students.

The head motion tracking system presented in this paper is inexpensive, easy to wear, and designed specifically for head motion monitoring of CP to provide valuable information for therapists about HCA.

3. Methodology

3.1. HeadUp System Design

The proposed HeadUp system was designed and tested as a medical HCA evaluation system. The HeadUp device is constructed to be convenient for patients who wear it and to be informative for physicians who use it to monitor the parameters of the patient's head movements. As shown in Figure 2, head motion can be described in the following terms [45]:

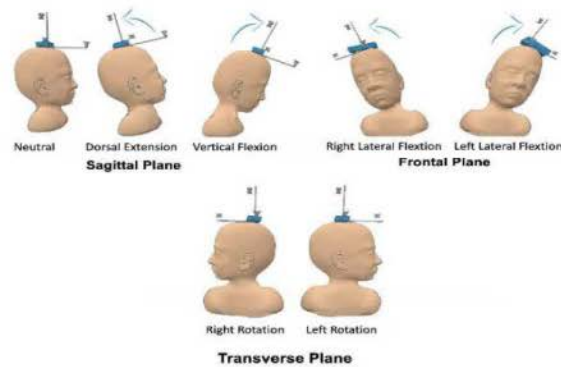


Figure 2. Head motion terms.

- Flexion (F)
- Extension (E)
- Right Lateral Flexion (RLF)
- Left Lateral Flexion (LLF)
- Right Rotation (RR)
- Left Rotation (LR)

These head motion parameters are essential for evaluating the HCA of patients with CP. As illustrated in Figure 3, the HeadUp system includes the HeadUp device and the receiver device.

The HeadUp device comprises a microcontroller (Arduino Pro Mini 3.3 V), a 9-axis IMU sensor (MPU9255), a charge indicator unit, and transceivers (nRF24L01). The MPU9255 has a 3-axis accelerometer (Acc), 3-axis gyroscope (Gyro), and 3-axis magnetometer (Mag). The receiver device includes another microcontroller and nRF24L01 transceiver to receive the data. Both the HeadUp device and receiver device were powered by a rechargeable 1000 mAh Livion 3.7 battery. The total cost of the HeadUp system (HeadUp and receiver dongle) was approximately 18\$.

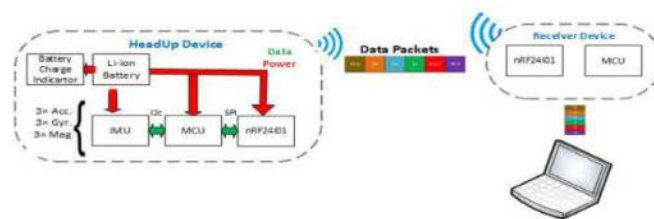
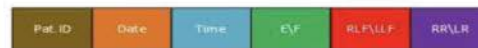


Figure 3. HeadUp system architecture.

The microcontroller, which is the main component for the HeadUp device, collects data by an inter-integrated circuit protocol (I^2C) bus, calculates the head movement parameters, encapsulates the data into packets with patient Id, dates, and sends the packets to the receiver device by radio-frequency. The data shown in Figure 4 will be collected by the receiver device and sent to a laptop through the UART protocol. The data are then saved in a (CSV) file for further analysis. The HeadUp system (transmitter and receiver devices) are shown in Figure 5.



Pat.ID correspond to Patient ID
 E/F correspond to head's Extension/Flexion
 RLF/LLF correspond to head's Right/Left Flexion
 RR/LR correspond to head's Right/Left Rotation

Figure 4. HeadUp data packet.



Figure 5. HeadUp system (transmitter and receiver). (a) HeadUp for moderate CP. (b) HeadUp for severe CP.

3.2. Data Acquisition and Sensor Fusion Algorithm

3.2.1. Data Acquisition

First, the raw data were obtained while the HeadUp device were stable; these raw data could not be applied without calibration. All three sensors (Acc+Gyro+Mag) need to be calibrated, and two values must be measured for each sensor (bias and scale). Bias is the difference between the raw value and the zeroes, while the scale value represents how much larger the range of sensor data is than the actual value of the physical movement. Without calibration, the measurements lack accuracy. Therefore, calibration is mandatory before the device is utilized.

The IMU sensor was sampled at a 100 Hz frequency and 16-bit resolution. The desired measurement from the HeadUp device is HCA, which can be evaluated by CROM angle measurement (θ , ϕ , ψ). These measurements were obtained by using the raw data from the IMU sensor.

The Acc measures the acceleration based on Newton's second law and the associated force. The head movement angles (extension/flexion (θ) and right/left lateral flexion (ϕ)) can be measured from only the calibrated Acc measurements using the following equations:

$$\theta_a = \tan^{-1} \frac{a_x}{(a_x)^2 + (a_z)^2} \times \frac{180}{\pi} \quad (1)$$

$$\phi_a = \tan^{-1} \frac{a_y}{(a_y)^2 + (a_z)^2} \times \frac{180}{\pi} \quad (2)$$

where θ_a is the estimated E/F angle and ϕ_a is the estimated RL/F/LL/F angle using only the Acc readings; a_x , a_y , a_z are the acceleration along x , y , and z axis.

The problem with these head movement angles, which are measured only from the Acc, is that they are susceptible to vibration and suffer from noise, as shown in Figure 6. A Butterworth low pass filter (LPF) was selected to filter the noise. A Fourier transform was used to calculate the cut-off frequency for the LPF.

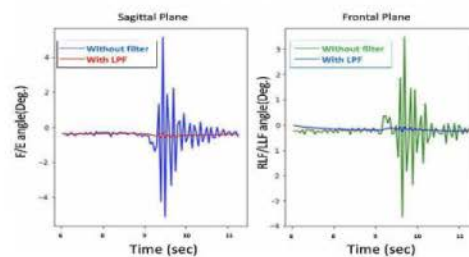


Figure 6. Before and after low pass filter with only Acc.

Although the filtered θ and ϕ were noise-free, as shown in Figure 7, the system response is sluggish, as depicted in Figure 8.

Figure 8 shows an estimation of the F/E angles, and the lag in the measurement can be clearly seen, so there is a trade-off between getting a noise-free signal and the system's response time.

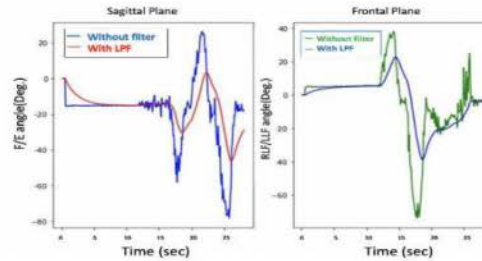


Figure 7. θ and ϕ before and after LPF with only Acc.

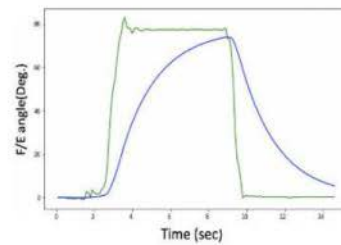


Figure 8. Delay problem with Acc. readings.

Another option for measuring head movement is to use a Gyro. The Gyro measures the object's angular velocity and can also indirectly measure head movement θ and ϕ by integrating angular velocity over time using the following equation:

$$\theta_g(t + \Delta) = \theta_g + \omega_y \Delta t \quad (3)$$

$$\phi_g(t + \Delta) = \phi_g + \omega_x \Delta t \quad (4)$$

where $\theta_g(t + \Delta)$ and θ_g are the new and previous estimated F/E angles using only the Gyro readings. $\phi_g(t + \Delta)$ and ϕ_g are the new and previous estimated RLF/LLF angles using only the Gyro readings; w_x, w_y, w_z which are the rotational velocities around $x, y,$ and z axis. The issue with the Gyro readings is that the θ_g and ϕ_g drift over time, as illustrated in Figure 9. This figure reveals that the head flexion drifts approximately 30° after 5 s. Both noise and drift contribute to unacceptable results.

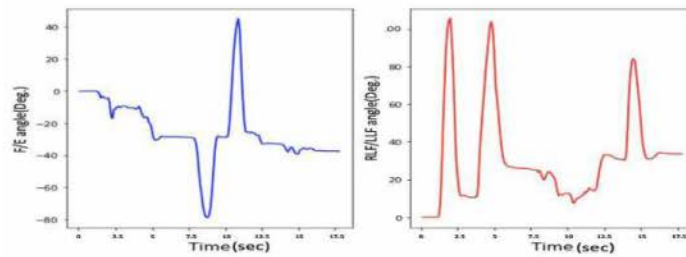


Figure 9. Drift problem in only gyroscope.

We can conclude that neither the Acc nor the Gyro provide accurate head movement measurements. Therefore, a sensor fusion algorithm (SFA) was implemented to overcome each sensor drawback and to obtain a noise-free and fast system response.

3.2.2. Sensor Fusion Algorithm

As mentioned in Section 3.2.1, the readings from the Acc are not accurate for high-frequency situations such as fast movement in short time intervals; that is why Gyro readings are used for these instances. To obtain accurate head movement measurements, an SFA was implemented to overcome the drawbacks of using each sensor alone and to get reliable readings and improve overall accuracy. To get faster response time and noise-free signal, a complementary filter was used by combining the desired low-frequency characteristic of the Acc and the desired high-frequency characteristic of the Gyro, as shown in Equations (5) and (6).

$$\theta(t + \Delta) = (1 - \alpha)[\theta(t) + \omega_y \Delta t] + \alpha \theta_a \quad (5)$$

$$\phi(t + \Delta) = (1 - \alpha)[\phi(t) + \omega_x \Delta t] + \alpha \phi_a \quad (6)$$

where $\theta(t + \Delta)$ and θ are the new and previous estimated E/F angles. $\phi(t + \Delta)$ and ϕ are the new and previous estimated RLF/LLF angles. α is a constant ($0 < \alpha < 1$). The larger the α the more Gyro is trusted, and as α converging to zero, we base our measurement more and more on the Acc readings. For the HeadUp system we chose α to be = 0.95.

Equations (5) and (6) put a high-pass filter on the Gyro measurements and a low pass filter on the Acc measurements, and then these signals are combined depending on the constant α to form the final angle estimation.

The complementary filter comprises of a high pass filter and a low pass filter and it may be applicable when the data received from the Acc and Gyro has errors at some period of time. On the one hand, small external forces can create errors in the measurements of the Acc, which means they are not reliable in the short term. On the other hand, the Gyro usually provides accurate data in a short time frame before the drift error can cause problems. To solve these issues, a low-pass filter can be utilized for the Acc, while a high-pass filter can be applied to the Gyro data [46]. As both low- and high-pass filters are included in the complementary filter, high-frequency components and low-frequency components can be handled. The fused data from the sensors will contain much less error compared to individual sensor readings.

Another filtering method that tends to be more complex is called Kalman filtering. The filter uses a recursive algorithm that can estimate unknown data using historical measurements [46]. There are several stages in the Kalman filter method. First, an initial value will be acquired from the system alongside a prediction algorithm used to correlate a prediction error [47]. Once a new measurement is available, the filter uses it to update the prediction value and the prediction error. As in this project we were aiming to implement

algorithms on an embedded Arduino microcontroller with limited computational power, we opted for a method with less computational cost [48], that is complementary filtering.

The Acc and Gyro can only measure the head's F/E and head RL/E/LL/E. A magnetometer is needed to measure the head's right/left rotation; ψ as illustrated in Equation (7). The applied SFA is presented in Figure 10; the steps are shown in Algorithm 1.

$$\psi = \tan^{-1} \frac{m_y \cos \phi - m_z \sin \phi}{m_x \cos \theta + m_y \sin \phi \sin \theta + m_z \cos \phi \sin \theta} \times \frac{180}{\pi} \quad (7)$$

where ψ is the estimated RR/LR angles and m_x, m_y, m_z are the magnetometer raw readings; strength of the earth's magnetic vector along the sensor's $x, y,$ and z axis.

Algorithm 1: Sensor Fusion Procedure

Data: IMU's Raw Data

Result: The CROM angles: θ, ϕ, ψ

initialization; $\theta, \phi, \psi = 0$;

R=True

while R **do**

 -IMU Calibration;

 -Read Data: Accelerometer data (a_x, a_y, a_z) ,

 Gyroscope data (w_x, w_y, w_z) and Magnetometer

 data (m_x, m_y, m_z)

 -calculate θ_d

 -calculate ϕ and θ

 -calculate ψ

end

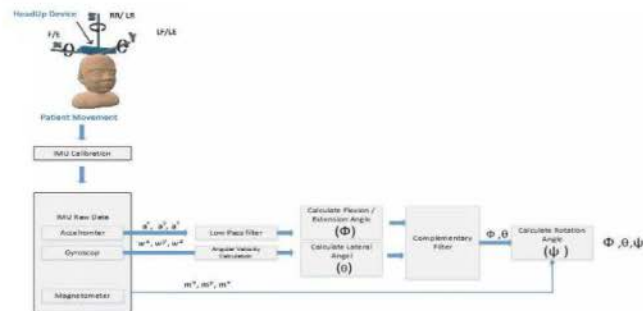


Figure 10. HeadUp SFA and filtering procedure.

As shown in Figure 10, the SFA has the following steps:

1. Calibration step for all the sensors' readings (Acc, Gyro, and Mag) to ensure that all the measurements are close to zero when the system is at rest;
2. Butterworth low pass filter for the Acc readings to get rid of high-frequency additive noise;
3. To get faster response time and noise-free measurements, a complementary filter was used;
4. Finally, the magnetometer was used along with Acc and gyro to acquire the head rotation.

The measurements from the HeadUp device are presented in Figure 11.

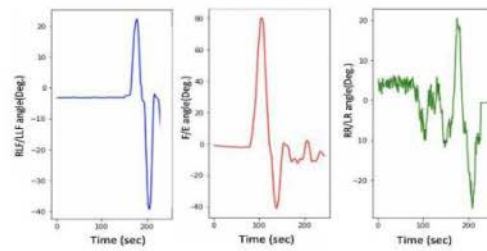


Figure 11. Head movement with a 9-axis IMU for typically developing child.

This figure shows the fast response time and the clean measurement with no cumulative drift or noise, demonstrating the proposed SFA's validity.

3.3. System Validation

The accuracy of the HeadUp device was tested by comparing the flexion angle measurement with another CROM tool in both Sagittal and frontal planes. A protractor with an arm was employed as the validation device, which was composed of two arms with a protractor in between the arms, and could be utilized to give the CROM [49]. This comparison was performed by placing both devices on a flat surface with an adjustable angle to measure the decline of the surface, as presented in Figure 12. The surface was moved at four angles (30° , 45° , 80° , and 90°). To ensure reading validity, a linear analysis was performed for the CROM measurements recorded by the protractor with the arm and HeadUp device (Figure 13), and the mean differences in the measurement from both devices were calculated. Note that the proposed SFA for the HeadUp device will filter all the noise and fix the drift problem if there is any as explained in Section 3.2. The validation results are discussed in Section 5.

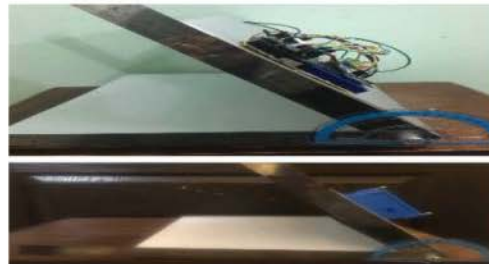


Figure 12. HeadUp device validation using protractor with arm.

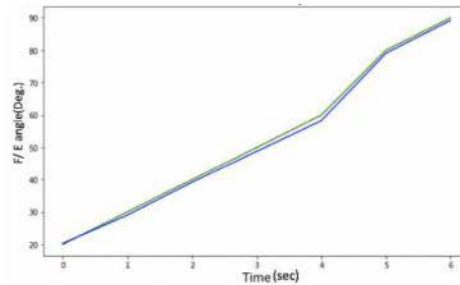


Figure 13. Protractor with arm vs. HeadUp.

4. System Implementation

4.1. Subjects' Selection

Ten male and female children have participated in this research (five children with various CP degrees (male: 4; female: 1; age: 2–5)); This age range was selected because it is a significant period to train the HCA (early age) for future improvement, and the definitive diagnosis of CP is difficult before two years of age [50]. All the selected subjects with CP suffered from poor IICA (level IV–V), responded to simple tasks, and understood straightforward instructions. Five typically developing children were also selected for this research with the same age range (2–5 y/o) [5].

Table 1 shows the characteristics of the participants with CP; the clinicians usually asked for these data during the assessment. For the typically developed children, all of them were healthy.

Table 1. Participants' with CP characteristics.

No.	Name	Age	Sex	CP Level	Test Condition	Notes
1	IM	3.5	M	V	sitting	full term baby, mild squint, Mixed Cp
2	AH	5	M	V	sitting	full term baby, sever squint
3	MA	2	M	IV	sitting	full term baby, No squint
4	FH	5	F	V	sitting	full term baby, mild squint
5	HJ	3.5	M	IV	sitting	neglected baby

4.2. Measurements

As mentioned in Section 3.1, the HeadUp device includes an IMU sensor consisting of a 3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer. As previously mentioned, the main aim of this study is the use of custom build sensors to help capture active range of head movement for children with CP (level V and IV) and help control their movements. In such CP levels, the child cannot walk or stand or sit by him/herself, the child can only sit with support. This means the lower part of the body will be stationary and will not affect head movement, as a result only one IMU has been used to capture the head movement. Additionally, different studies showed that one IMU can capture the full head orientation [24,51–53].

A HeadUp device was mounted on the subject's head using a head cap for severe children with CP, while it was located on a hair hoop for children with mild CP. Figure 14 shows that HeadUp was mounted on the child's head, allowing the child to move his/her head in different directions while seated. HeadUp was used to gather the acceleration, magnetic field, and angular velocity of the head. The data gathered from HeadUp were applied to evaluate the IICA based on the proposed method.

Each child wore the HeadUp device while sitting with lumbar support and was instructed to perform the following movement (refer to Figure 2) [24,54,55]

- Case 1: sit still without any movement (Natural).
- Case 2: dorsal extension and ventral flexion.
- Case 3: right lateral flexion and left lateral flexion.
- Case 4: right rotation and left rotation.

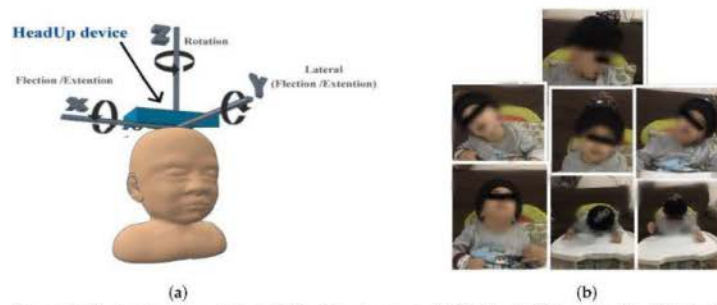


Figure 14. System implementation. (a) HeadUp placement. (b) Child with CP wearing HeadUp device.

These movements are recorded by the receiver device, which receives the data packets (Figure 4), and displays the results on the host computer and saves the data as a CSV file. The HeadUp system measured head angular movement (sagittal plane, frontal plane) and rotational movement (transverse plane) at 100 Hz.

5. Result

As mentioned in Section 1, HCA affects many aspects of a child's life. Head stability gives a stable reference to vertical posture, which is very important to achieve the vertical posture. Figure 15 illustrates the HCA in the sagittal plane, frontal plane, and transverse plane during the sitting trial for subject 1 (child with CP). Figure 11 shows the same head movement for a typically developing child of the same age and no reported CROM problem.

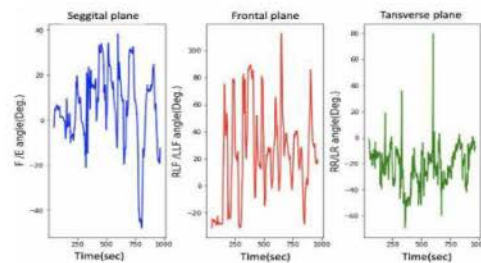


Figure 15. Head movement in 3 dimensions for a child with CP.

Figures 11 and 15 show the same sequence head movements; the subjects were asked to do F/E, L/LF/R/LF, and RR/LR head movement. Figure 11 shows that head movements for a typically developed child. That figure clearly indicates the child could perform these moves, and he/she has a good head control ability and no CROM problem. While Figure 15 indicates that the child with cerebral palsy had trouble controlling their head movement and that figure contains more random movement than Figure 15.

The research experiments start with case 1, where all participants were asked to sit without any head movement. Figures 16 and 17 shows the results for case 1. The experimental results for each participant show considerable differences among the head movement signals captured from typically developing participants and those captured from children with CP.

As illustrated in Figure 17 in the sagittal plane for children with CP, continuous variation in the F/E angle is missing for normal children, whereas normal head control in the sagittal plan can be observed in Figure 16. Additionally, this difference is noted on the RLF/LLF in the frontal plane in both figures. Figure 17 indicates the poor HCA for children with CP.

Moreover, Figure 17 shows if the child's head and trunk rotated or collapsed consistently to one side. This information helps the physiotherapist perform physical therapy for each child with CP.

Note that the head control ability for subject 1 was tested in different ways, the results of which are depicted in Figures 15 and 17. Figure 15 depicts the case where the instructions were to perform a different head movement, and that figure shows the child's poor ability to control his head. Then Figure 17 shows the same subject trying to sit still without any head movement. The subject was able to control his head rotation in the Transverse plane, but in the Sagittal plane and Frontal plane, it is hard for him to hold his head.

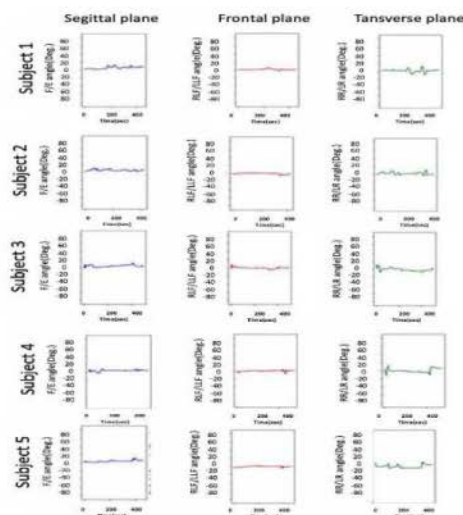


Figure 16. Head movement for typically developing children.

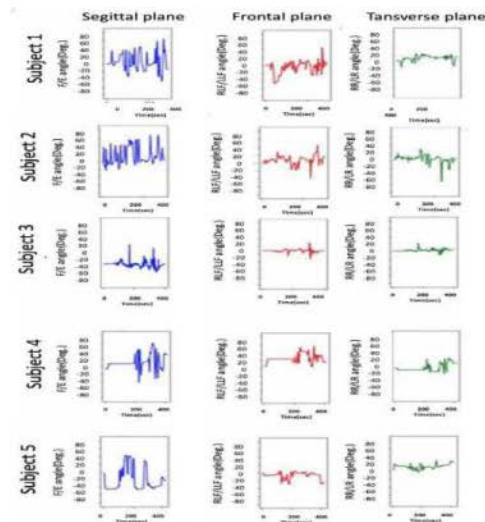


Figure 17. Head movement for children with CP.

The second part of the study experiments consisted of cases 2, 3, and 4, where all participants were asked to perform simple head movements (F, E, LLF, RLF, RR, and LR). The results for this part are shown in Table 2.

Table 2. Mean values and standard deviations of CROM for the participants.

Movements	Typically Developing Children	Children with CP
F	$60.3^{\circ} \pm 13.30^{\circ}$	$55.4^{\circ} \pm 10.11^{\circ}$
E	$30.38^{\circ} \pm 9.10^{\circ}$	$25.22^{\circ} \pm 10.40^{\circ}$
RLF	$35.7^{\circ} \pm 6.98^{\circ}$	$32.1^{\circ} \pm 7.75^{\circ}$
LLF	$35.30 \pm 8.30^{\circ}$	$33.81^{\circ} \pm 9.82^{\circ}$
LR	$80.04^{\circ} \pm 8.03^{\circ}$	$60.96^{\circ} \pm 10.23^{\circ}$
RR	$78.81^{\circ} \pm 10.86^{\circ}$	$66.03^{\circ} \pm 9.27^{\circ}$

As mentioned in Section 3.3, the device accuracy was tested against another CROM tool; protractor with arm. The mean square error was calculated to see how close the HeadUp's measurements were to the ground truth. The reported mean square error was less than 2° , which demonstrates the validity of the HeadUp device. Although the test was done for only 6 s, device validity was tested for a longer time during research experiments. Figures 11 and 15–17 show the fast response time and the clean readings with no accumulative drift or noise, demonstrating the proposed SFA's validity.

6. Discussion

This study showed an investigation in applying an IMU-based system to capture active head movement and measure head control ability for young children with severe CP; level IV or V. It was found that the HeadUp system demonstrated the capability to record the head movement in E/F, LLF/RLF, and RR/LR while sitting.

Moreover, HeadUp device allows physicians to capture children's active head movement while they are engaged in various activities—whether in the health center or through-

out their regular lives in their home. The HeadUp system could aid in improving diagnostic and in the suggestions for treatments.

The HeadUp device was mounted on the subject's head while sitting with lumbar support, as illustrated in Figure 14, and all the participants were instructed to perform different head movements.

Figures 16 and 17 show the HCA without any instructed activity. In normal cases, the head movement angles should be around zero, and this can be clearly seen in Figure 16.

Figure 16 shows that the HeadUp device is able to estimate the head movements for typically developing children in all directions. The difference in the sagittal plane in Figures 16 and 17 indicates the number of head drops for children with CP. This shows the lack of head control ability by children with CP which has been captured by the sensor. Figure 17 shows the HCA for children with CP. Different medical information can be extracted from that figure:

1. All participants with CP have a poor head control ability.
2. The head of subject 4 collapsed consistently to the right side, while the head of subject 5 collapsed consistently to the left side, which mean the muscles in these sides are weak (this can be seen from RLF/LLF angles).
3. Subject 3 has better HCA than other participants with CP.

After the first stage, different instructions were given to the participants to evaluate the ability of HeadUp system to measure the range of motion (ROM), as shown in Table 2. There were several noticeable differences between the ROM of typically developing children and children with CP. For example, the average ROM in the transverse plane (LR and RR) for a child with CP was 63° compared with an average ROM of 79° for typically developing children. This means that the head rotation ability for children with CP is lower than Typically developed children.

Head motion in the frontal plane for normal children had a larger mean ROM (35° RLF and 35° LLF) for the five healthy participants compared with a mean of 32° RLF and 33° LLF) for the children with CP. The mean CROM in the sagittal plane (Flexion) was 60° for normal children, while it was 55° for children with CP. All these medical information can be valuable for improving diagnostic and in the suggestions for treatments and physiotherapy program.

7. Conclusions

Head movement tracking for children with CP with proper efficiency is a challenge. There is a need for a low-cost, accurate, and easy-to-use device that can evaluate and analyze the HCA to detect poor head control for children with CP. Therefore, this research paper presents in-house head motion monitoring to analyze the HCA and to provide the best physiotherapy program.

On the basis of the acquired results, the proposed HeadUp device showed an average reliability of 2° compared with other CROM tools (protractor with arm). The HeadUp-based IMU device captured head movement in three planes (sagittal, frontal, and transverse). An SFA was presented to overcome the high noise in the measurements, the noise was reduced in the filtered readings, and as a result, the accuracy of the measurements was increased, which paved the way to accurately identify the HCA for children with cerebral palsy. IMU utilization resolves the problems faced by current CROM methods, such as accuracy, expense, and size, by designing a portable, small, and low-cost alternative that clinicians and physiotherapists can utilize for monitoring the HCA to identify head movement abnormalities.

The validity and reliability of the HeadUp device against one of the standard head angle measurement tools (protractor with arm) were evaluated in two planes (sagittal and frontal). HeadUp's evaluation gives therapists important information that helps in identifying the appropriate and effective rehabilitation program. In future work, several possible improvements to the system can be explored further, such as:

- Validates the HeadUp system's results against a more reliable tool in three planes with both devices on the child's head.
- Investigates different filtering algorithms, such as the Kalman filter
- Use the HeadUp device in an entertaining way as a HCA trainer and examine its performance to improve the child's head stability.
- Constructed a standalone HCA diagnosis device with the help of machine learning algorithms to distinguish head movement disorder from the typical head movement patterns pattern.

Author Contributions: S.S.A.-a. contributed to the design, data curation and implementation. S.S.A.-a., S.K., and E.K.H. contributed to formal analysis and review of results. S.S.A.-a., S.K., I.M., and H.A. contributed to supervision and conceptualization of this paper. S.S.A.-a. and E.K.H. contributed to data collection, measurement validation. The article was written and edited by S.S.A.-a., reviewed by S.S.A.-a., S.K., H.A., and I.M. All authors discussed the results and reviewed and commented on the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical approval for this study was given by the University of Information Technology and Communications and by the Health Directorate with approval number HRE-1403. The children's parents also gave their permission before the subjects participated in the research.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to patient privacy.

Acknowledgments: The authors thank the Rehabilitation Medical Centre and Joint Diseases, Iraq, for providing us with all the support towards the successful implementation of the study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are employed in this manuscript:

CP	Cerebral palsy
CROM	cervical range of motion
IMU	inertial measurement unit
SFA	sensor fusion algorithm
GMFCS	Gross Motor Function Classification System
HCD	head control device
HPT	head position trainer
F	Flexion
E	Extension
RLF	Right Lateral Flexion
LLF	Left Lateral Flexion
RR	Right Rotation
LR	Left Rotation
Acc	accelerometer
Gyro	gyroscope
LPF	low pass filter
HCA	head control ability
ROM	range of motion

References

1. Bax, M. Terminology and classification of cerebral palsy. *Dev. Med. Child Neurol.* **1964**, *6*, 295–297. [[CrossRef](#)]
2. Bax, M.; Goldstein, M.; Rosenbaum, P.; Leviton, A.; Paneth, N.; Dan, B.; Jacobsson, B.; Damiano, D. Proposed definition and classification of cerebral palsy, April 2005. *Dev. Med. Child Neurol.* **2005**, *47*, 571–576. [[CrossRef](#)] [[PubMed](#)]

3. Kirby, R.; Wingate, M.; Braun, K.; Doernberg, N.; Arneson, C.; Benedict, R.; Mulvihill, B.; Durkin, M.; Fitzgerald, R.; Maenner, M. Others Prevalence and functioning of children with cerebral palsy in four areas of the United States in 2006: A report from the Autism and Developmental Disabilities Monitoring Network. *Res. Dev. Disabil.* **2011**, *32*, 462–469. [\[CrossRef\]](#)
4. Krigger, K. Cerebral palsy: An overview. *Am. Fam. Phy.* **2006**, *73*, 91–100.
5. Berker, A.; Yalçın, M. Cerebral palsy: Orthopedic aspects and rehabilitation. *Pediatr. Clin. N. Am.* **2008**, *55*, 1209–1225. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Palisano, R.; Rosenbaum, P.; Bartlett, D.; Livingston, M. Content validity of the expanded and revised Gross Motor Function Classification System. *Dev. Med. Child Neurol.* **2008**, *50*, 744–750. [\[CrossRef\]](#)
7. Harris, F.; Spelman, F.; Hymer, J. Electronic sensory aids as treatment for cerebral-palsied children: Inappropriation: Part II. *Phys. Ther.* **1974**, *54*, 354–365. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Redstone, F.; West, J. The importance of postural control for feeding. *Pediatr. Nurs.* **2004**, *30*, 97–100.
9. Curtis, D.; Butler, P.; Saavedra, S.; Bencke, J.; Kallemose, T.; Sonne-Holm, S.; Woollacott, M. The central role of trunk control in the gross motor function of children with cerebral palsy: A retrospective cross-sectional study. *Dev. Med. Child Neurol.* **2015**, *57*, 351–357. [\[CrossRef\]](#)
10. Kramer, J.; Ashton, B.; Brander, R. Training of head control in the sitting and semi-prone positions. *Child Care Health Dev.* **1992**, *18*, 365–376. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Ian, L.; Paul, H.; John, D.; Penelope, B. Others Quantification of head motion in children with cerebral palsy when testing segmental trunk control. In Proceedings of the 2018 IX International Seminar Of Biomedical Engineering (SIB), Bogota, Colombia, 16–18 May 2018; pp. 1–4.
12. Wong, W.; Wong, M.; Lo K. Clinical applications of sensors for human posture and movement analysis: A review. *Prosthet. Orthot. Int.* **2007**, *31*, 62–75. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Heyrman, L.; Molenaers, G.; Desloovere, K.; Verheyden, G.; De Cat, J.; Monbaliu, E.; Feys, H. A clinical tool to measure trunk control in children with cerebral palsy: The Trunk Control Measurement Scale. *Res. Dev. Disabil.* **2011**, *32*, 2624–2635. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Fedorak, C.; Ashworth, N.; Marshall, J.; Paull, H. Reliability of the visual assessment of cervical and lumbar lordosis: How good are we? *Spine* **2003**, *28*, 1857–1859. [\[CrossRef\]](#)
15. Krosshaug, T.; Nakamae, A.; Boden, B.; Engebretsen, L.; Smith, G.; Slauterbeck, J.; Hewett, T.; Bahr, R. Estimating 3D joint kinematics from video sequences of running and cutting maneuvers—Assessing the accuracy of simple visual inspection. *Gait Posture* **2007**, *26*, 378–385. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Abyarjoo, F.; Barreto, A.; Cofino, J.; Ortega, F. Implementing a sensor fusion algorithm for 3D orientation detection with inertial/magnetic sensors. In *Innovations And Advances In Computing, Informatics, Systems Sciences, Networking And Engineering*; Springer: Cham, Switzerland, 2015; pp. 305–310.
17. Sharif Bidabadi, S.; Murray, I.; Lee, G. The application of inertial measurements unit for the clinical evaluation and assessment of gait events. *J. Med. Eng. Technol.* **2017**, *41*, 612–622 [\[CrossRef\]](#) [\[PubMed\]](#)
18. Ch, rasiri, R.; Abhayasinghe, N.; Murray, I. Bluetooth embedded inertial measurement unit for real-time data collection for gait analysis. In Proceedings of the International Conference On Indoor Positioning And Indoor Navigation, Montbeliard, France, 28–32 October 2013.
19. Petraglia, F.; Scarcella, L.; Pedrazzi, G.; Brancato, L.; Puers, R.; Costantino, C. Inertial sensors versus standard systems in gait analysis: A systematic review and meta-analysis. *Eur. J. Phys. Rehabil. Med.* **2019**, *55*, 265–280. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Xu, W.; Ortega-Sanchez, C.; Murray, I. Measuring human joint movement with IMUs: Implementation in custom-made low cost wireless sensors. In Proceedings of the 2017 IEEE 15th Student Conference on Research And Development (SCoReD), Wilayah Persekutuan Putrajaya, Malaysia, 13–14 December 2017; pp. 172–177.
21. Rodríguez-Martín, D.; Pérez-López, C.; Samà, A.; Català, A.; Moreno Arostegui, J.; Cabestany, J.; Mestre, B.; Alcaine, S.; Prats, A.; Crespo, C. Others A waist-worn inertial measurement unit for long-term monitoring of Parkinsons disease patients. *Sensors* **2017**, *17*, 827. [\[CrossRef\]](#)
22. O'Reilly, M.; Caulfield, B.; Ward, T.; Johnston, W.; Doherty, C. Wearable inertial sensor systems for lower limb exercise detection and evaluation: A systematic review. *Sport. Med.* **2018**, *48*, 1221–1246. [\[CrossRef\]](#)
23. Velasco, M.; Raya, R.; Muzzioli, L.; Morelli, D.; Otero, A.; Iosa, M.; Cincotti, F.; Rocon, E. Evaluation of cervical posture improvement of children with cerebral palsy after physical therapy based on head movements and serious games. *Biomed. Eng. Online* **2017**, *16*, 1–13. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Carmona-Pérez, C.; Pérez-Ruiz, A.; Garrido-Castro, J.; Vidal, F.; Alcaraz-Clariana, S.; García-Luque, L.; Souza, D. Alburquerque-Sendín, F. Design, Validity, and Reliability of a New Test, Based on an Inertial Measurement Unit System, for Measuring Cervical Posture and Motor Control in Children with Cerebral Palsy. *Diagnostics* **2020**, *10*, 661. [\[CrossRef\]](#)
25. Carmona-Pérez, C.; Garrido-Castro, J.; Torres Vidal, F.; Alcaraz-Clariana, S.; García-Luque, L.; Alburquerque-Sendixn, F.; Souza, D. Concurrent validity and reliability of an inertial measurement unit for the assessment of craniocervical range of motion in subjects with cerebral palsy. *Diagnostics* **2020**, *10*, 80. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Whitcroft, K.; Massouh, L.; Amirfeyz, R.; Bannister, G. Comparison of methods of measuring active cervical range of motion. *Spine* **2010**, *35*, E976–E980. [\[CrossRef\]](#)

27. Defibaugh, J. Measurement of Head Motion: Part I: A Review of Methods of Measuring Joint Motion. *Phys. Ther.* **1964**, *44*, 157–163. [[CrossRef](#)]
28. Voss, S.; Page, M.; Bengler, J. Methods for evaluating cervical range of motion in trauma settings. *Scand. J. Trauma Resusc. Emerg. Med.* **2012**, *20*, 50. [[CrossRef](#)]
29. Werner, I.; Ernst, M.; Treleaven, J.; Crawford, R. Intra and interrater reliability and clinical feasibility of a simple measure of cervical movement sense in patients with neck pain. *BMC Musculoskelet. Disord.* **2018**, *19*, 358. [[CrossRef](#)]
30. Yoon, T.; Kim, H.; Min, J. Validity and Reliability of an Inertial Measurement Unit—Based 3-Dimensional Angular Measurement of Cervical Range of Motion. *J. Manip. Physiol. Ther.* **2019**, *42*, 75–81. [[CrossRef](#)] [[PubMed](#)]
31. Feng, M.; Liang, L.; Sun, W.; Liu, G.; Yin, X.; Han, T.; Wei, X.; Zhu, L. Measurements of cervical range of motion using an optical motion capture system: Repeatability and validity. *Exp. Ther. Med.* **2019**, *18*, 4193–4202. [[CrossRef](#)]
32. Lecoulet, B.; Boisclair, D.; Evin, M.; Wagnac, E.; Petit, Y.; Aubin, C.; Arnoux, P. Assessing the global range of motion of the helmeted head through rotational and translational measurements. *Int. J. Crashworthiness* **2019**, *25*, 321–327. [[CrossRef](#)]
33. Cuesta-Vargas, A.; Galán-Mercant, A.; Williams, J. The use of inertial sensors system for human motion analysis. *Phys. Ther. Rev.* **2010**, *15*, 462–473. [[CrossRef](#)]
34. Abhayasinghe, N.; Murray, I.; Sharif Bidabadi, S. Validation of Thigh Angle Estimation Using Inertial Measurement Unit Data against Optical Motion Capture Systems. *Sensors* **2019**, *19*, 596. [[CrossRef](#)]
35. Han, Y.; Wong, K.; Murray, I. Gait Phase Detection for Normal and Abnormal Gaits Using IMU. *IEEE Sens. J.* **2019**, *19*, 3439–3448. [[CrossRef](#)]
36. Rudigkeit, N.; Gebhard, M.; Gräser, A. Evaluation of control modes for head motion-based control with motion sensors. In Proceedings of the 2015 IEEE International Symposium on Medical Measurements And Applications (MeMeA) Proceedings, Turin, Italy, 7–9 May 2015; pp. 135–140.
37. Velasco, M.; Raya, R.; Muzzioli, L.; Morelli, D.; Iosa, M.; Cincotti, F.; Rocon, E. Evaluation of cervical posture improvement of children with cerebral palsy after physical therapy with a HCI based on head movements and serious videogames. In *International Conference On Bioinformatics And Biomedical Engineering*; Springer: Cham, Switzerland, 2016; pp. 495–504.
38. Cunningham, R.; Sánchez, M.; Butler, P.; Southgate, M.; Loram, I. Fully automated image-based estimation of postural point-features in children with cerebral palsy using deep learning. *R. Soc. Open Sci.* **2019**, *6*, 191011. [[CrossRef](#)]
39. Ball, T.; Crady, R.; Hart, A. Automated reinforcement of head posture in two cerebral palsied retarded children. *Percept. Mot. Skills* **1975**, *40*, 619–622. [[CrossRef](#)]
40. Wooldridge, C.; Russell, G. Head position training with the cerebral palsied child: An application of biofeedback techniques. *Arch. Phys. Med. Rehabil.* **1976**, *57*, 407–414. [[PubMed](#)]
41. Walmsley, R.; Crichton, L.; Droog, D. Music as a feedback mechanism for teaching head control to severely handicapped children: A pilot study. *Dev. Med. Child Neurol.* **1981**, *23*, 739–746. [[CrossRef](#)] [[PubMed](#)]
42. Leiper, C.; Miller, A.; Lang, J.; Herman, R. Sensory feedback for head control in cerebral palsy. *Phys. Ther.* **1981**, *61*, 512–518. [[CrossRef](#)] [[PubMed](#)]
43. Malouin, F.; Gemmell, M.; Parrot, A.; Dutil, R. Effects of auditory feedback on head position training in young children with cerebral palsy: A pilot study. *Physiother. Can.* **1985**, *37*, 150–156.
44. Ashton, B.; Kramer, J.; Heaver, A. Head position trainer application for research and clinical use. *Phys. Ther.* **1986**, *66*, 1228–1232. [[CrossRef](#)]
45. Tousignant, M.; Smeesters, C.; Breton, A.; Breton, É.; Corriveau, H. Criterion validity study of the cervical range of motion (CROM) device for rotational range of motion on healthy adults. *J. Orthop. Sport. Phys. Ther.* **2006**, *36*, 242–248. [[CrossRef](#)] [[PubMed](#)]
46. Islam, T.; Islam, M.; Shajid-Ul-Mahmud, M.; Hossam-E-Haider, M. Comparison of complementary and Kalman filter based data fusion for attitude heading reference system. *AIP Conf. Proc.* **2017**, *1919*, 020002.
47. Yan, W.; Zhang, Q.; Wang, L.; Mao, Y.; Wang, A.; Zhao, C. A modified kalman filter for integrating the different rate data of gyros and accelerometers retrieved from android smartphones in the GNSS/IMU coupled navigation. *Sensors* **2020**, *20*, 5208. [[CrossRef](#)]
48. Mahmooda, D.; Javaida, N.; Imran, M.; Qasim, U.; Khand, Z. Data Fusion for Orientation Sensing in Wireless Body Area Sensor Networks using Smart Phones. *Sensors* **2015**, *15*, 14458–14486.
49. Chaves, T.; Nagamine, H.; Belli, J.; De Hannai, M.; Bevilacqua-Grossi, D.; De Oliveira, A. Reliability of fleximetry and goniometry for assessing cervical range of motion among children. *Braz. J. Phys. Ther.* **2008**, *12*, 283–289. [[CrossRef](#)]
50. Wood, E. The child with cerebral palsy: Diagnosis and beyond. *Semin. Pediatr. Neurol.* **2006**, *13*, 286–296. [[CrossRef](#)]
51. Rudigkeit, N.; Gebhard, M.; Graser, A. An analytical approach for head gesture recognition with motion sensors. In Proceedings of the 2015 9th International Conference On Sensing Technology (ICST), Auckland, New Zealand, 8–10 December 2015; pp. 1–6.
52. MacNeilage, P.; Glasauer, S. Quantification of head movement predictability and implications for suppression of vestibular input during locomotion. *Front. Comput. Neurosci.* **2017**, *11*, 47. [[CrossRef](#)]
53. Severin, I.; Dobrea, D.; Dobrea, M. Head Gesture Recognition using a 6DOF Inertial IMU. *Int. J. Comput. Commun. Control* **2020**, *15*, 3–6. [[CrossRef](#)]
54. Tecklin, J. *Pediatric Physical Therapy*; Lippincott Williams & Wilkins: Philadelphia, PA, USA, 2008.
55. Aneja, S. Evaluation of a child with cerebral palsy. *Indian J. Pediatr.* **2004**, *71*, 627–634. [[CrossRef](#)]

4.4 USE CASE IN POSTURE MONITORING AND CORRECTION EXERCISES FOR WORKERS IN HOSTILE ENVIRONMENTS



Article

Posture Monitoring and Correction Exercises for Workers in Hostile Environments Utilizing Non-Invasive Sensors: Algorithm Development and Validation

Siavash Khaksar ^{1,*}, Stefanie Pieters ¹, Bitu Borazjani ¹, Joshua Hyde ¹, Harrison Booker ¹, Adil Khokhar ¹, Iain Murray ¹ and Amity Campbell ²

¹ School of Electrical Engineering, Computing, and Mathematical Sciences, Curtin University, Bentley, WA 6102, Australia

² School of Allied Health, Curtin University, Bentley, WA 6102, Australia

* Correspondence: siavash.khaksar@curtin.edu.au

Abstract: Personal protective equipment (PPE) is an essential key factor in standardizing safety within the workplace. Harsh working environments with long working hours can cause stress on the human body that may lead to musculoskeletal disorder (MSD). MSD refers to injuries that impact the muscles, nerves, joints, and many other human body areas. Most work-related MSD results from hazardous manual tasks involving repetitive, sustained force, or repetitive movements in awkward postures. This paper presents collaborative research from the School of Electrical Engineering and School of Allied Health at Curtin University. The main objective was to develop a framework for posture correction exercises for workers in hostile environments, utilizing inertial measurement units (IMU). The developed system uses IMUs to record the head, back, and pelvis movements of a healthy participant without MSD and determine the range of motion of each joint. A simulation was developed to analyze the participant's posture to determine whether the posture present would pose an increased risk of MSD with limits to a range of movement set based on the literature. When compared to measurements made by a goniometer, the body movement recorded 94% accuracy and the wrist movement recorded 96% accuracy.

Keywords: personal protective equipment (PPE); hostile environments; posture monitoring; musculoskeletal disorder; optical motion capture; fiber-optic sensing; e-textile sensors; inertial measurement unit (IMU); Kalman filter; Euler angles; quaternion angles; Unity; human joint measurement



Citation: Khaksar, S.; Pieters, S.; Borazjani, B.; Hyde, J.; Booker, H.; Khokhar, A.; Murray, I.; Campbell, A. Posture Monitoring and Correction Exercises for Workers in Hostile Environments Utilizing Non-Invasive Sensors: Algorithm Development and Validation. *Sensors* **2022**, *22*, 9618. <https://doi.org/10.3390/s22249618>

Academic Editor: Vittorio Passaro

Received: 1 November 2022

Accepted: 5 December 2022

Published: 8 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background

PPE is an essential key factor in standardizing safety within the workplace. Harsh working environments with long working hours can cause stress on the human body may result in musculoskeletal disorder (MSD). MSD refers to injuries that impact the muscles, nerves, joints, and many other human body areas [1]. Most work-related MSD results from hazardous manual tasks involving repetitive, sustained force, or repetitive movements in awkward postures [1].

MSD impacts the workers and the employer in the form of economic loss due to absenteeism, lost productivity, increased health care, disability, and worker's compensation claims [1]. Based on the Australian Workers' Compensation Statistics from 2018 to 2019, 36% of compensation claims were due to body stress, resulting in a median of 6.2 weeks lost per severe claim [2]. The percentage rate of severe claims due to MSD between male and female workers is 87%, with laborers being the highest compared to several other working groups [2].

The age group most impacted by this issue are between 45 and 49 years of age. However, even the youngest workers under 20 years old have 3650 claims of injury and

MSD [2]. These statistics show that this is, in fact, a severe issue that needs to be dealt with and will be beneficial for all working-age groups.

1.2. Existing Methods

A standard device currently used to measure joint angles is known as a goniometer. A specific type of goniometer is used to measure motion in the spine and is known as a gravity-dependent goniometer or inclinometer [3]. This method requires precision for an accurate reading that is only obtained through practice and skillful observation [3]. The slightest misplacement can lead to an inaccurate reading and usage would not be suitable in the proposed application area and will not offer continued monitoring of the active range of movement.

Safe Work Australia's Hazardous manual task Code of Practice states that a movement that is repeated or sustained for long period that ranges 20° out of the human posture's natural state can pose a significant risk of MSD [4]. An angle of 30° for spinal range is used to make the range less conservative. In addition to this, a goniometer is used to verify the obtained data.

Optical passive motion capture technologies use retro-reflective markers attached to the body parts of the individual that reflects light onto a nearby camera lens. From this reflection, the position of the marker is calculated within three-dimensional space and recorded [5]. This approach is also known as motion capture or mo-cap which is the process of digitally recording the movement of people [6]. This approach is used in sports, entertainment, ergonomics, medical applications, and robotics and is also known as performance capture when looking at the full body, face, and fingers. Optical active motion capture uses the same technique, but rather than reflecting light, the light is emitted [5]. Optical motion capture technology provided the most accurate results based on research [5] and is well equipped for use in a laboratory environment. This method is considered as the gold standard for capturing human movement; however, due to its considerable expense, with a simple Vicon system [7] costing around \$250,000 Australian dollars in 2011 [8], its impracticality for small harsh environments, and its inherent complexity [9], optical motion sensing is impractical for most field-based settings.

Fiber-optic sensors are another example of potential field use and rely on the measurement of light traveling through an optical fiber system. This measurement can be in terms of light intensity, phase, or polarization [10]. Fiber-optic sensing provided a robust design that could withstand harsh environments by tolerating high temperatures, offered a wide dynamic range and large bandwidth, and was not susceptible to electromagnetic interference, radio frequency, or corrosive environments [11]. Even though this is a new method recently developed for posture monitoring, it has shown that it is a solid competitor compared to optical motion capture technology producing similar results [12]. However, due to its considerable expense and inherent complexity, fiber-optic sensing was not chosen.

Another potential approach could be the use of e-textile sensors, which is a common phrase referring to electronic textiles. Electronic textiles are fabrics that incorporate electronics and interconnections woven within them [13]. E-textile sensors provided a less visible and invasive design. This method provided reliable results when compared to optical motion capture technology [14]. This procedure required minimal complexity to implement. Due to this method's lack of durability in harsh environments (susceptible to interference with parasitic capacitance due to heavy sweating and relaxation of the tight stretchable fabric due to continuous use and washing) which can result in unreliable data, e-textile sensors were not chosen [14].

Inertial measurement units (IMU) are one of the popular field-based methods for tracking the movement and positioning of an object. IMU's consist of an accelerometer to measure force and acceleration, a gyroscope to measure the rate of change in angles, and lastly a magnetometer that utilizes the earth's magnetic field as a fixed reference for the current estimation of the IMU orientation to prevent drift [15].

The inertial measurement unit (IMU) provides a well-developed, non-invasive, affordable design with long battery life [16]. Less advanced theory is required to implement this method and has proven to be a reliable form of posture monitoring with several cases to refer to [17]. There is an option of customizing the IMU or choosing a pre-calibrated and developed system. Due to these advantages, IMUs were chosen as the desired method. There are several data-driven methods for using IMU data in conjunction with neural networks to classify human movement. For example, IMUs have previously been used for medical purposes such as capturing foot drop in [18] and the hand movement of children with cerebral palsy in [19]. Ref. [15] shows a system for using multiple IMUs connected to the legs of patients with foot drop issues and uses machine learning to classify the severity and need for surgery compared to healthy participants. Ref. [16] uses IMUs to capture the hand movement of children with cerebral palsy, as well as typically developing children, and uses machine learning to classify the movement associated with cerebral palsy from the IMU data. In another example, [20] provides a method for using image processing, neural networks, and public databases for capturing human movement. They implement this using 15 sensors. Even though the results look very promising, the large number of sensors and the processing power required to analyze the data are unsuitable for hostile environments. To overcome this issue, rather than relying on machine learning, the proposed system focuses on real-time quaternion data and a range of joint angle movements to monitor the user movement, as well as provide feedback to them for potential use in posture correction exercises. The detail of this implementation is explained in the methods section of the paper [20].

1.3. Contribution of the Paper

The main contribution of this paper is providing a digital, low-cost system and framework for human posture monitoring and exercises for workers in a hostile environment. The sensor set up provides a clinically accurate representation of wrist, elbow, and knee joint movement which has been validated in a Vicon motion analysis system and goniometers. This system and framework provide the ability to adjust the range of movement for different body parts and the length of time spent at each range. This framework can also be utilized as a digital rehabilitation tool where rehab exercises related to the wrist, elbow, and knee can be captured in real time and provide the user with feedback on how accurately the exercise is being completed. For this paper, the focus has been on providing this feedback for workers in a hostile environment.

2. Methods

2.1. System Requirements

The aim of this paper is to provide a system that will enable workers to have their posture monitored whilst doing certain activities and provide posture correction exercises, with feedback to the user so they can see if they are doing the exercise correctly. The target user will be any worker or individual that require having their posture monitored whilst doing activities. This system will be used to provide posture monitoring with a visual aspect for now.

Requirements that were identified as essential for the success and effectiveness of the project were that the system must be able to record live stream data, show a visual representation of movement, detect harmful and non-harmful angles which need to be defined for each user, and display a message for the user regarding the harmful position. Additional requirements that will be examined are the measurement of wrist, knee, and elbow joint angles.

A process diagram for the proposed system can be seen in Figure 1 and shows the process that was followed when implementing the system.

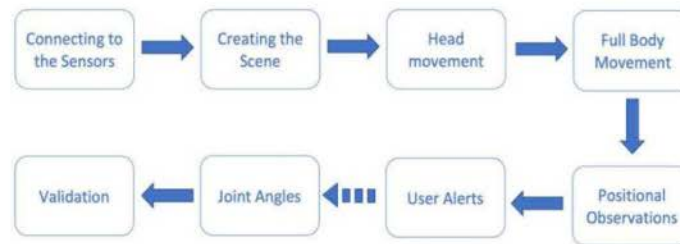


Figure 1. Process diagram.

2.2. Selecting a Suitable Sensor

One of the first steps for this project was to select suitable IMU sensors that are small, lightweight, have a long battery life, utilize BLE 5.0 (Bluetooth Low Energy), have a high sampling rate, provide continuous measurements throughout the working shift, and are economically priced. BLE 5.0 connectivity is needed as it is more robust and can transmit 8 times more data at twice the speed compared to BLE 4.3 or BLE 2.1 [21].

There are a wide variety of off-the-shelf IMU sensors on the market; however, only a select few IMU sensors seen in Table 1 were used for comparison. With the aforementioned factors considered, Xsens Dot was chosen as the IMU to use in this project as it can be seen in Figure 2 that the Xsens Dot was the most well-rounded choice, although the Vicon blue trident was a close contender. However, due to the noticeable price difference Xsens Dot was chosen. Figure 3 shows a photo of the Xsens Dot sensors used in this paper.

Table 1. Different IMU choices for posture monitoring.

IMU	Sampling Rate	Connectivity	Battery Life	Weight	Size	Price
Xsens Dot [22]	120 Hz	BLE 5.0	9 h	11.2 g	36.3 × 30 × 10.8 mm	€495.00 (~\$798.05 AUD) for 5 pack
Vicon Blue Trident [23]	100 Hz	BLE 5.0	12 h	9.5 g	42 × 27 × 11 mm	\$1600.00 USD (~\$2184.36 AUD) each
Shimmer IMU [24]	128 Hz	BLE 2.1	14 h	23.6 g	51 × 34 × 11 mm	€359.00 (~\$578.79 AUD) each
Bonsai IMU [25]	50 Hz	BLE 4.3	16 h	15 g	36.5 × 32 × 13.5 mm	€2490.00 (~\$4014.44 AUD) for 15 pack

Orientation and free acceleration are obtained from the Xsens dot by means of an in-built fusion algorithm and a Kalman filter. This fusion algorithm is referred to as the XKFCore of the Xsens Dot IMU [22].

The Xsens Dot is sized at L:36.3 × W:30.4 × H:10.8 mm with a weight of 11.2 g. This provides a small and lightweight device that does not hinder the user's movement. The internal storage is 64 MB with a sampling rate of 800 Hz. This provides enough storage for storing the captured data when necessary and a sampling rate capable of capturing fast movement. The output rate ranges from 1 Hz to 60 Hz with 120 Hz available only for recording. Communication is conducted through Bluetooth [22].

The Xsens Dot provides 9 h battery life which means the sensors can provide continuous monitoring of the posture without the need to change the battery. The electrical current consumption of one Xsens Dot is 68 mA [22]. The battery within the Xsens dot is an LIR2032H rechargeable coin battery. Battery specifications include a nominal capacity of 70 mAh, a nominal Voltage of 3.7 V, and a working temperature of −20–+60° [22].

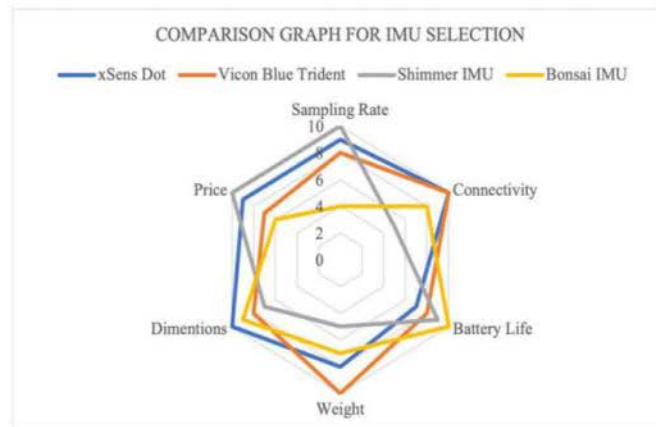


Figure 2. Comparison graph for IMU selection based on Table 1.



Figure 3. The Xsens Dot IMU [18].

The Xsens Dot can operate in temperatures ranging from 0 to 50° Celsius which is within the required standards for underground environments. The IP Rating is IP68 which indicates that the Xsens Dot can withstand damage caused by dust or water (can be submerged up to 1.5 m deep) [26].

2.3. Filtering and Sensor Fusion

It is necessary to use fusion algorithms to filter out the external noise and integrate all the sensor data. There are several different methods used for filtering, namely, for example, Kalman filtering, complementary filtering, and particle filtering. The Xsens Dot uses an in-built fusion algorithm for capturing real-time orientation and a filtering method such as a Kalman filter.

Kalman filtering is one of the most common estimation algorithms and plays an essential role in the IMU fusion algorithm [27]. Developed in 1960, the Kalman filter is used today for navigation systems and control systems [28]. The objective of the Kalman filter is to minimize the mean squared error of the measured data compared to the estimated results [29]. This is completed by using two basic steps: prediction and correction. The prediction step uses the control commands given to predict where the dynamic system will be at the next point in time. The correction step uses the data obtained by the IMU sensor to correct any potential mistakes that have been made and determines a prediction error to

use when the following prediction is made [27]. This prediction and correction step cycles continuously to provide accurate results and is known as recursive estimation.

One limitation that the Kalman filter possesses is that it is not well suited for working with nonlinearities due to the assumptions made to develop a Kalman filter; however, since human movement is linear, it is not a major issue. These assumptions being that the filter will only work with Gaussian distribution, and all models are linear [30]. An alternative Kalman filter was created, known as the extended Kalman filter, that deals with non-linearities by performing local linearization with the Taylor approximation of the non-linear model to work around this problem. This method is used to turn it into a linear model based on linearization points that need to be updated for each prediction of the recursive estimation [30].

Euler angles describe the rotation and orientation of a body in three-dimensional space from an initial frame to a final frame [31]. The angles used are commonly known as yaw, pitch, and roll. Euler angles describe the orientation between two 3D coordinate systems. This orientation can be represented in a 3×3 coordinate system parameterized by Euler angles.

Advantages to using Euler angles are that it is easier to visualize and can describe rotation and orientation in a precise manner [32]. Euler angles do have a disadvantage which is that this technique is susceptible to gimbal lock, which is the phenomenon where one degree of freedom is lost due to two axes aligning. For example, when the pitch approaches 90° , the roll and yaw is locked, thus making them indistinguishable [33]. Without an external reference, it is impossible to re-orientate the axis once gimbal lock occurs [33].

A method of working around gimbal lock is to use quaternion angles instead. Quaternion angles consist of 4 components: a real component and three imaginary components. Quaternion angles describe three-dimensional rotations and orientation with a generalization of complex numbers [32]. These angles are then later converted to a regular rotation matrix, instead of a rotation matrix as seen with Euler angles. Quaternion angles simplify the equation by using a quaternion notation to represent a rotation of θ degrees about an axis defined by the vector $\hat{u} = (u_x, u_y, u_z)$, as seen in Equation (1).

$$\hat{q} = \cos\left(\frac{\theta}{2}\right) + (u_x i + u_y j + u_z k) \sin\left(\frac{\theta}{2}\right) \quad (1)$$

2.4. Software Specification

The Unity game engine [34] and the Xsens Dot application were the tools utilized for implementing the software component of the posture monitoring framework.

Unity is a powerful system used for designing games and application scenes in 2D or 3D. With correct use of programming, Unity can be utilized to capture motion data for analysis. Programming language such as C# is used to develop scripts within the model. Unity was the main platform for developing the model as it has extensive reference and scripting documentation that can be used to start obtaining motion capture data as quickly as possible.

The hardware Xsens Dot IMU provides an application called "Xsens dot" that is used to obtain motion capture data directly for the IMU via Bluetooth. This application has the capability to record real-time streaming of the IMU and log the data into a csv file that can be exported onto a computer for analysis. From the Xsens Dot application provided, the IMUs were connected to the application through Bluetooth connectivity. The advanced application gives the users the option to measure the quaternion, Euler, free acceleration, acceleration, magnetic field, and angular velocity.

2.5. Prototype Design

A prototype has been set up with a standard PPE helmet that has an inner frame for fitting adjustments (Figure 4). It was decided that the Xsens Dot sensor will be placed on

the top of the head as this placement will provide the most accurate results. It was decided that a harness similar to the harness seen in [17] shall be created to determine the correct placement of the IMU on the user's back for the best results. It was determined that placing the IMU sensors on the back of the chest and the hips is the best placement to receive reliable results. The harness prototype can be seen in Figure 5.



Figure 4. Helmet prototype setup.



Figure 5. Harness prototype.

Harnesses similar to this are being used in some mines around the world. This harness is designed to carry additional load that would normally be placed around the belt/pants. This design prevents any possible injury or discomfort.

The Xsens Dot IMUs are placed within a plastic zip-lock bag and positioned on the harness with Velcro. This is not a permanent solution. However, it does provide a temporary solution to evaluate posture monitoring.

Utilizing the Xsens Dot application discussed, the head movement was monitored (validation of this IMU has already been completed and can be viewed in Validation). Figure 6 shows a user pivoting (bending) the head left and right to a 30° angle. Thirty degrees has been chosen as discussed in the existing Methods section. When the head pivots to the right, the Euler angle in the X-axis produced a positive 30° angle. When the head pivots to the left, the Euler angle in the X-axis produced a negative 30° angle. In the natural state the angle is nearly zero. The flexion and rotation of the head was also monitored and produced the same pattern of results with the Y-axis and Z-axis, respectively.

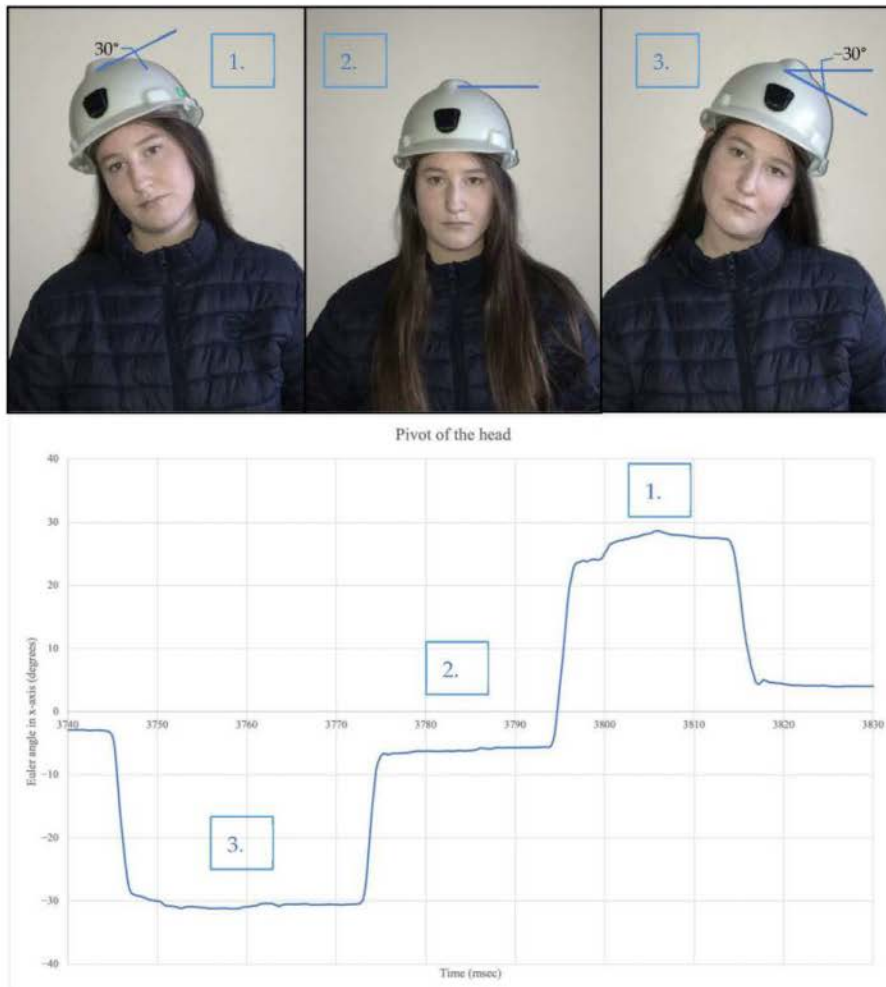


Figure 6. Orientation measurement with head pivoting.

Figure 7 shows the participant pivoting (bending) the chest left and right to a 30° angle. When the chest pivots to the right, the Euler angle in the X-axis produced a positive 30° angle. When the chest pivots to the left, the Euler angle in the X-axis produced a negative 30° angle. In the natural state, the angle is nearly zero. The flexion and rotation of the chest was also monitored and produced the same pattern of results with the Y-axis and Z-axis, respectively.

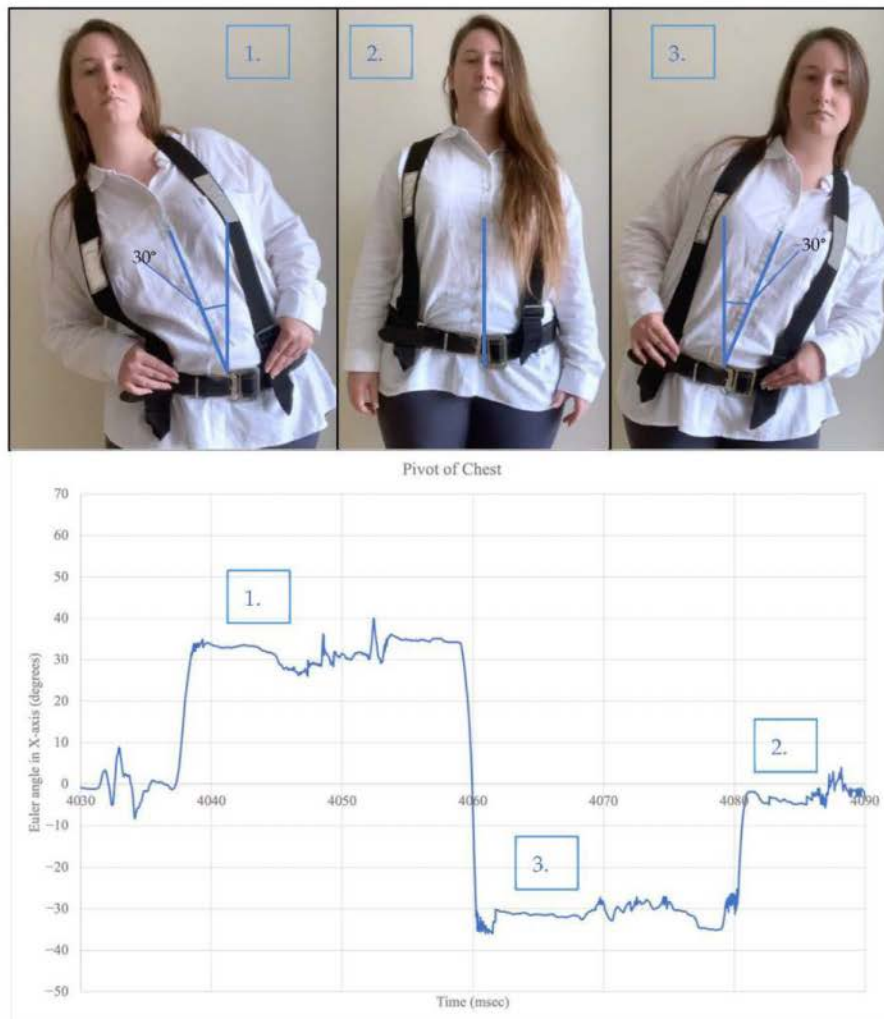


Figure 7. Orientation measurement with chest pivoting.

Figure 8 shows the participant pivoting (bending) the hips left and right to a 30° angle. When the hips pivot to the right, the Euler angle in the X-axis produced a positive 30° angle. When the hips pivot to the left, the Euler angle in the X-axis produced a negative 30° angle. In the natural state, the angle is nearly zero. The flexion and rotation of the hips were also monitored and produced the same pattern of results with the Z-axis and Y-axis, respectively.

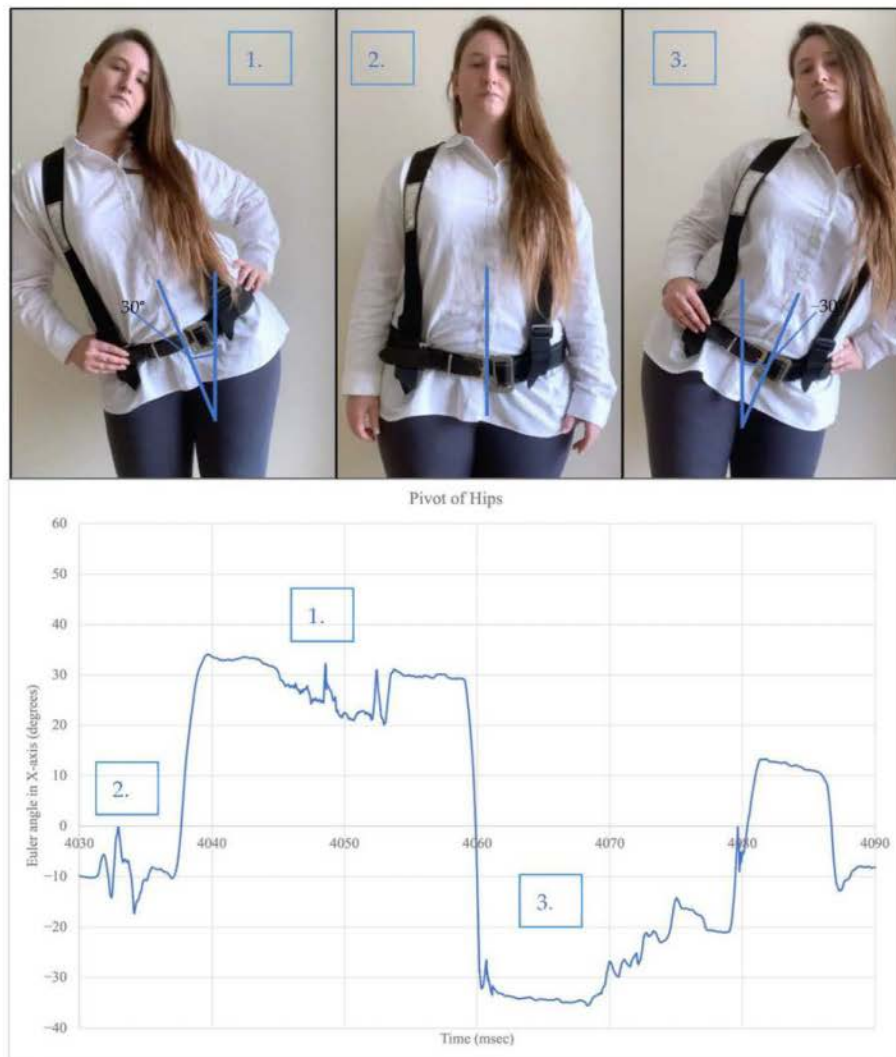


Figure 8. Orientation measurement with hips pivoting.

Standard Velcro straps were used to monitor any arm or leg movement. The orientation measurement of the arms and legs are not required as only the head, chest, and pelvis are the main body parts that will be monitored to determine poor posture. The Xsens Dot sensors will be positioned on the Velcro straps similarly by placing the Xsens Dot sensor into a plastic zip-lock bag and attaching the sensor to the arm or leg with Velcro.

2.6. Connecting to Sensors

The sensors needed to be connected to a computer via Bluetooth to transfer their data and pass them to Unity for visualization. To achieve this, a graphical user interface has been developed in Python that uses Bluetooth to scan for available IMUs, synchronizes them, and passes the information directly to Unity. The application has been developed by implementing a reliable Transmission Control Protocol (TCP) client (the Python Bluetooth module) and TCP receiver (Unity script asset called the ServerReceiver).

TCP (Transmission Control Protocol) is a transport layer protocol that is used in conjunction with IP to ensure the reliable transmission of packets. TCP is more reliable as it requires a handshake to start the session. Handshake refers to a connection establishment protocol where a connection request (CR) is first sent to the receiver from the sender and then waits for a 'connection accepted' sent to the sender from the receiver.

A GUI has been developed in Python where the users can scan the sensors, as seen in Figure 9. This interface streams the IMU orientation data to Unity where the data stream can be seen in Figure 10.



Figure 9. Bluetooth module and GUI.

Once connection from all sensors have been established, selecting the run button will start streaming the quaternion angles from each sensor to the receiver. The TCP packets in JSON format are shown in Figure 10 where each line represents a new packet. With quaternion, 'wq' represents the real component, and the remainder represents the imaginary components.

Figure 10. TCP Packets in JSON format.

2.7. Joint Angle Measurements Methods

Three different methods were examined to measure joint angle movement of the wrist, knee, and elbow. The method chosen needed to be able to replicate measurements that are made by a goniometer (a medical device used to measure joint angle during movement). The chosen method was incorporated within the simulation for users to access, if desired. The measurements made will also be accessible through a csv file within Unity.

The first method involved obtaining two quaternion angles from the child game object and the object that it is immediately attached to (the parent). The quaternion angles obtained from the parent and child are used to create a rotate vector that is in reference to a unit vector on an axis (obtained by script `quaternions.cs`). A dot product between the two rotate vectors is obtained and arc cosine is used to produce an angle which is later converted to degrees. A high-level flow chart of the process (with wrist rotation used as an example) can be seen in Figure 11.

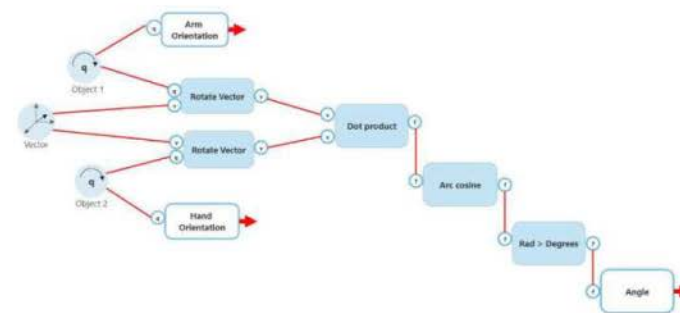


Figure 11. Method 1 joint calculation.

The second involved obtaining two quaternion angles from the child game object and the object that it is immediately attached to (the parent) and determining the difference in angle between the child body part and the parent body part. This difference is calculated by multiplying the inverse of the quaternion from the parent to the quaternion from the child. This new quaternion is converted to Euler and displayed to the user. A high-level flow chart of the process (with wrist rotation used as an example) can be seen in Figure 12.

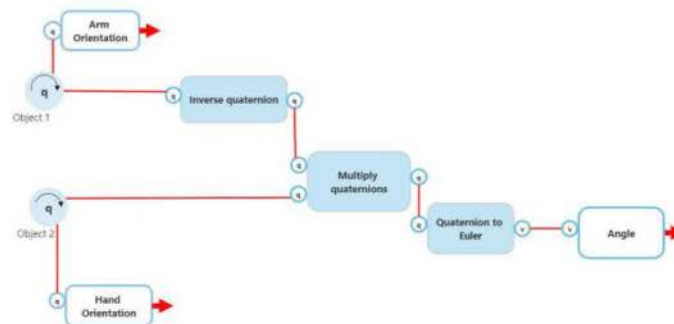


Figure 12. Method 2 joint calculation.

The third method involved using a function provided by Unity, namely “gameobject.localRotation.eulerAngles”. This function provides an angle of the child game object in reference to the object that it is immediately attached to (the parent). For this example, this function will provide an angle, which can be represented as a joint angle, between the hand and the lower arm. A high-level flow chart of the process (with wrist rotation used as an example) can be seen in Figure 13. The three methods have been simultaneously compared to measurements made by a goniometer.

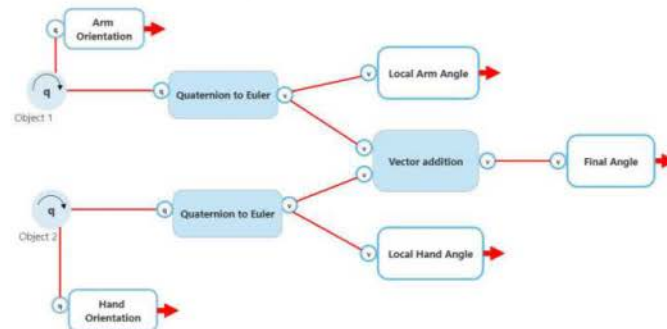


Figure 13. Method 3 joint calculation.

2.8. Visualizing the Data in Unity

The avatar used to mimic the participant’s movements was imported from the Unity asset store as it uses the ragdoll feature. This enables the user to develop a humanoid avatar with objects placed within their respective position based on the body mapping. With this helpful feature, it enables the user to have control of several joints of the humanoid with significant detail, as can be seen in Figure 14.

Further development of the scene was made, as seen in the result section to ensure users can see the date, time, movement angles of the main focused body parts, a message board when objects are out of bound, and a dropdown selection to change monitoring scenes. Further developments were made where joint angles can be monitored, thus a toggle selection for this method was incorporated as well.

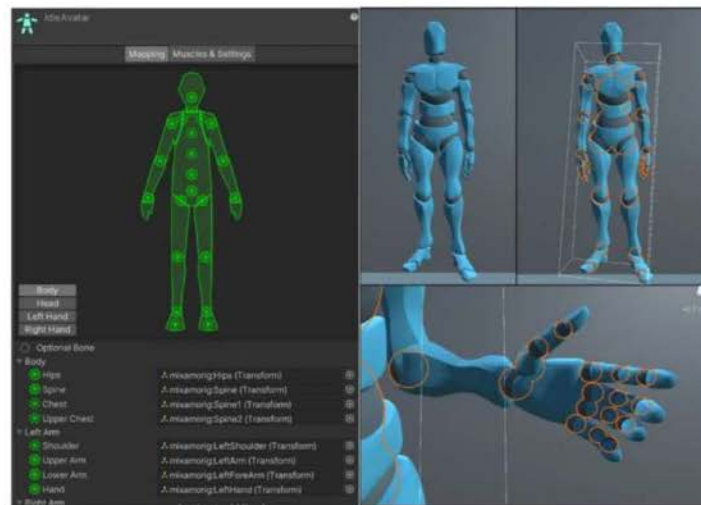


Figure 14. Body mapping of Unity avatar.

2.9. Sensor Evaluation

The accuracy of the Xsens Dot IMU sensors was validated with Curtin University's Vicon motion analysis lab. A simple wrist flexion and extension exercise was completed while the arm was resting on a table and reflectors were placed on the sensors. The data provided by the Vicon set up was then compared against the orientation data provided by the IMU. An example of the Vicon data vs. Xsens Dot data has been provided in Figure 15. Please note that the signals have been time shifted so the data can be compared more easily. The location of reflectors for the Vicon system can be seen in Figure 16.

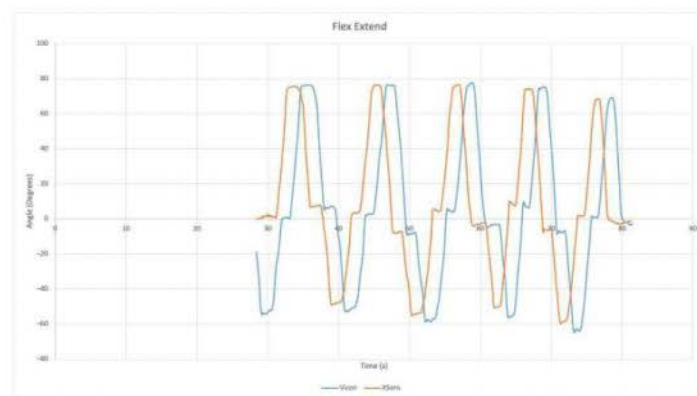


Figure 15. Validation of Xsens Dot IMU with Vicon.



Figure 16. Placement of reflectors for the Vicon validation.

After the accuracy of the IMUs were validated, a goniometer was then used to compare the reported results from the sensors to the readings of the goniometer. In this validation, full body rotation and joint angle validation was compared to a goniometer. The results of this validation have been discussed in the preliminary findings section of this paper.

3. Results

Full body movement was made possible by developing an array of structs with 13 allocations. A struct is a collection of variables that can be different types in programming. Each allocation represents an essential body part used for posture monitoring. Each struct consists of a quaternion, three Euler angles for rotation, correction, and positional control, the Xsens Dot sensor number controlling their respected body object, and lastly the name of the object that is being controlled. Adil's method was used to obtain the quaternion data of the Xsens Dot sensors. The quaternion angles of each sensor used are converted to Euler via an in-built Unity function called `quaternion.eulerAngles` [35]. Euler was chosen to be the displayed angle to the user as it is the easiest and most common angle used to understand rotation. `OnAnimatorIK` function [36] was used to provide the Euler angles to the avatar in order to create movement. This function that is provided by Unity gives the user the ability to access any object that is part of the avatar's anatomy (seen in Figure 14). The final version of the environment can be seen in Figure 17 and the movement can be seen in Figure 18. Supplementary Materials Video S1 to S5 contain several video demonstrations for the framework.

One problem that arose when implementing full body movement was that a calibration was needed to ensure that the avatar can return to in its natural state when the sensors set the avatar in an unnatural position. This calibration needed to be created in a way that still ensured that the angles provided were still deemed accurate and only needed to be completed the first time sensors were attached. This was achieved by creating a trigger token, the C button. When the user pressed the C button, the system will took a new reading of the Xsens Dot sensors and stored it in a temporary rotation float. From this, the original rotation is subtracted from the temporary rotation, providing a difference that establishes a new rotation between -180 and 180° . This method can be seen executed in Figure 19.

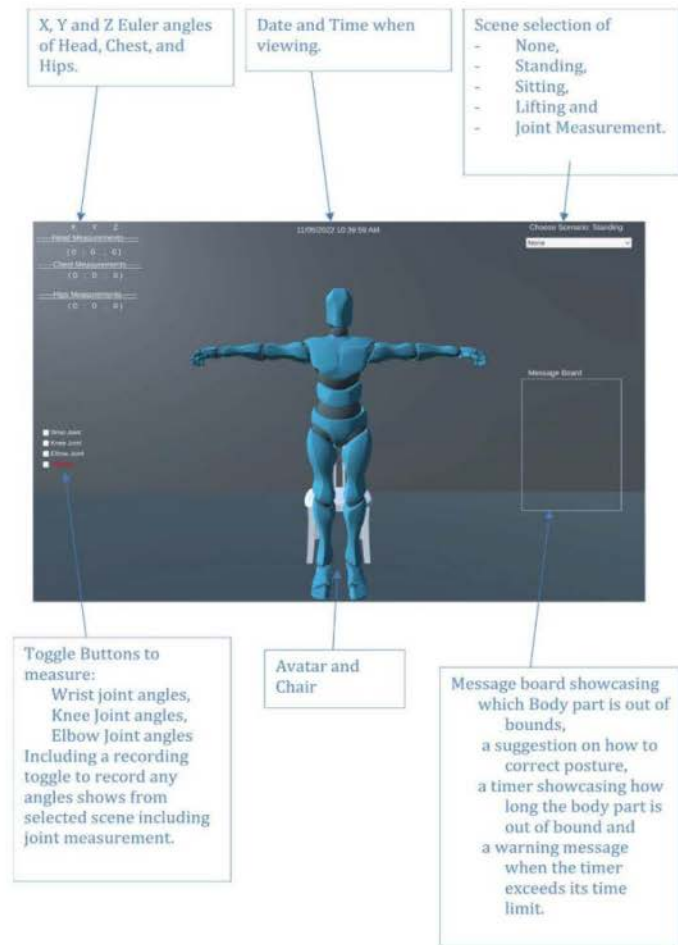


Figure 17. Scene prototype.

Once full body movement was attained, it was necessary to monitor different movement scenarios. It was determined that standing, sitting, lifting, and joint movement would be monitored in this model. A user interface dropdown selection was made that showcased the different scenarios available. This dropdown was connected to the `setPosition.csv` script. An integer variable between 0 and 4 was given from the dropdown list to the `Pos` function within `setPosition.csv` script representing scene None to Lifting, respectively. This function provided a Euler angle variable noted as "pos" to the `Rotations.csv` script. Through all 4 scenes, the head, chest, and hips are being monitored with the head, chest, hips, upper arms, and upper legs receiving rotation. Figure 20 showcases some natural movement that can be accomplished from selecting the sitting or lifting scene, respectively.

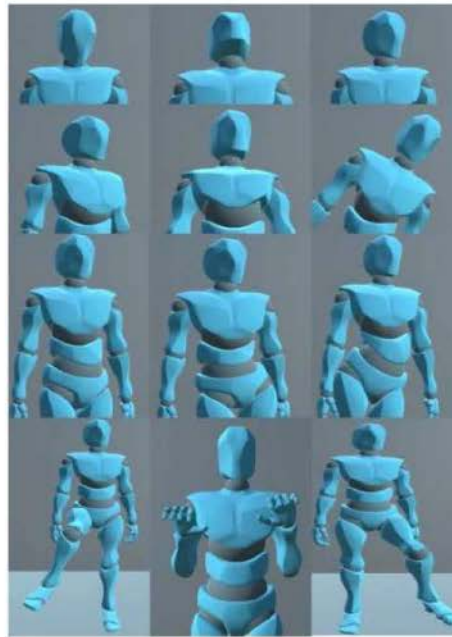


Figure 18. Full body movement of avatar.

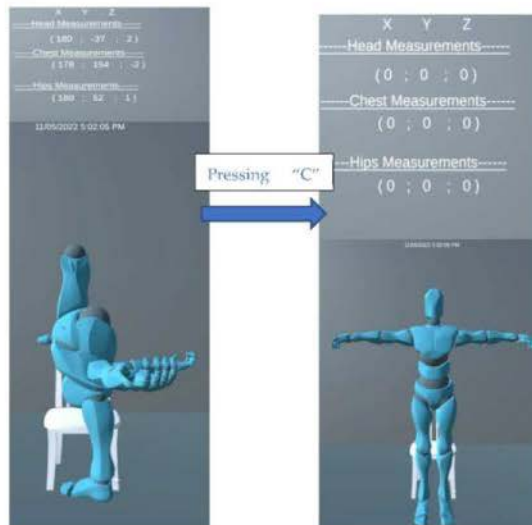


Figure 19. Implementing Calibration.

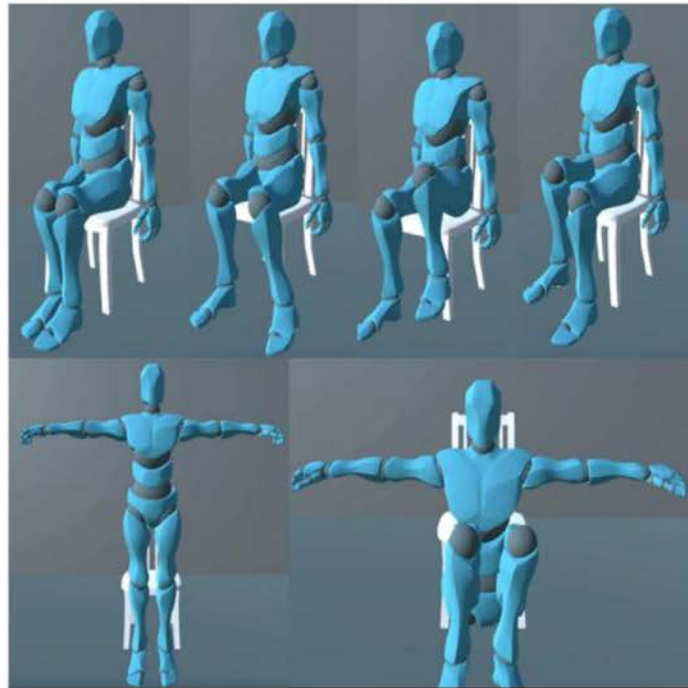


Figure 20. Natural sitting motion with sitting scene (above) and natural crouching motion with lifting scene (below).

4. Discussion

4.1. Principal Findings

As previously mentioned, three joint measurements methods were investigated. Method 1 and Method 2 stayed consistent, providing results similar to the measurements made by the goniometer, while Method 3 started producing promising results regarding the knee and elbow. Method 3 produced good results at a later state since the parent and child both started with a Euler angle of $(0^\circ, 0^\circ, 0^\circ)$. When the wrist joints were measured, the upper arm had a starting Euler angle of $(-80^\circ, 0^\circ, 0^\circ)$ and the lower arm had a starting angle of $(0^\circ, 0^\circ, 90^\circ)$. It was decided that Method 2 would be the method of choice as the results gave clear positive and negative values based on the choice of direction when completing the required motions. Method 2 also had clear singular changing X, Y and Z angles when completing each activity, which will be more favorable to the user.

The sensors were validated against a goniometer by taking angle measurements at $0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ$, and 50° angles using the goniometers and comparing the readings with the IMU-based measurements. Table 2 illustrates the validation results of the head, chest, and hips when compared to measurements made by a goniometer. It is seen that the results can be deemed as reliable. Tables 3 and 4 illustrate the validation results of a user's wrist when the hand is in an ulnar and radial deviation, respectively. Tables 5 and 6 illustrate the validation results of a user's wrist when the hand is in flexion and extension, respectively. Tables 7 and 8 illustrate the validation results of a user's wrist when the hand is in pronation and supination, respectively. Table 9 showcases the validation results of a

user's knee when the hand is in flexion and extension. Table 10 showcases the validation results of a participant's elbow when the hand is in flexion and extension. The results of the tests prove the reliability of Method 2 as the main source of joint measurement for this application.

Table 2. Measurements of full body movement (All angles are in degrees).

Goniometer	Head			Chest			Hips		
	Up and Down Motion	Left and Right Motion	Pivoting Left and Right Motion	Up and Down Motion	Left and Right Motion	Pivoting Left and Right Motion	Up and Down Motion	Left and Right Motion	Pivoting Left and Right Motion
Angle	X Angle	Y Angle	Z Angle	X Angle	Y Angle	Z Angle	X Angle	Y Angle	Z Angle
0	0	0	0	0	0	0	0	0	0
10	10	10	8	11	10	9	10	10	11
20	20	20	19	20	20	20	20	20	18
30	29	30	30	29	30	30	30	30	30
40	40	38	40	40	40	39	40	40	N/A
50	49	50	50	49	50	47	50	49	N/A

Table 3. Measurements of ulnar deviation of wrist (All angles are in degrees).

Goniometer	Method 1			Method 2			Method 3		
	X	Y	Z	X	Y	Z	X	Y	Z
0	0	0	1	-1	0	0	-7	-83	-35
10	11	NAN	11	-1	12	0	-1	-74	-36
20	19	NAN	19	-1	19	0	4	-67	-35
30	30	NAN	30	-1	31	0	11	-58	-34
40	38	NAN	38	-1	39	0	16	-49	-32
50	50	NAN	50	-1	52	0	21	-40	-29

Table 4. Measurements of radial deviation of wrist (All angles are in degrees).

Goniometer	Method 1			Method 2			Method 3		
	X	Y	Z	X	Y	Z	X	Y	Z
0	1	NAN	1	-1	1	0	-6	-82	-36
10	11	3	11	-1	-9	0	-12	-91	-34
20	20	4	20	-1	-20	0	-18	-100	-32
30	29	4	29	-1	-29	0	-23	-108	-29
40	42	5	42	-1	-38	0	-28	-120	-24
50	50	6	50	-1	-51	0	-31	-129	-19

Table 5. Measurement of flexion of wrist (All angles are in degrees).

Goniometer	Method 1			Method 2			Method 3		
	X	Y	Z	X	Y	Z	X	Y	Z
0	0	1	1	-1	0	0	-7	-83	-35
10	3	11	11	10	3	0	3	-87	-35
20	2	22	22	21	2	0	11	-93	-36
30	1	29	29	29	1	0	19	-100	-37
40	1	39	39	39	1	1	26	-106	-39
50	2	50	49	48	3	1	34	-113	-43

Table 6. Measurement of extension of wrist (All angles are in degrees).

Goniometer	Method 1			Method 2			Method 3		
Joint Angle	X	Y	Z	X	Y	Z	X	Y	Z
0	16	NAN	17	-1	16	1	2	-69	-35
10	19	5.0	22	-11	19	1	4	-61	35
20	21	18	29	-20	21	1	-10	-54	-35
30	23	31	37	-29	23	1	18	47	36
40	24	39	45	-41	23	1	-25	-40	-38
50	25	49	54	-50	24	2	-32	-32	41

Table 7. Measurements of pronation of wrist (All angles are in degrees).

Goniometer	Method 1			Method 2			Method 3		
Joint Angle	X	Y	Z	X	Y	Z	X	Y	Z
0	18	NAN	18	-1	18	-1	4	-68	-37
10	22	12	19	-1	19	11	4	-67	24
20	27	23	17	-1	17	21	3	-69	-15
30	34	28	16	-1	18	30	3	-69	-5
40	42	41	15	1	15	38	2	-70	4
50	52	51	14	-1	14	52	1	-71	13

Table 8. Measurements of supination of the wrist (All angles are in degrees).

Goniometer	Method 1			Method 2			Method 3		
Joint Angle	X	Y	Z	X	Y	Z	X	Y	Z
0	17	NAN	17	-2	17	0	3	-68	35
10	18	9	16	-1	16	-12	2	-70	-46
20	24	22	14	-1	14	21	1	-71	56
30	32	31	11	-1	11	-20	-1	-74	-66
40	41	43	9	0	9	43	1	-76	75
50	50	51	8	2	8	-51	-1	-78	-85

Table 9. Knee angle measurements from flexion of the leg (All angles are in degrees).

Goniometer	Method 1			Method 2			Method 3		
Joint Angle	X	Y	Z	X	Y	Z	X	Y	Z
0	0	NAN	1	1	0	0	0	5	0
10	11	20	4	-1	-4	-10	0	1	-10
20	21	20	6	-2	-5	-20	0	-1	-20
30	30	30	7	-2	-6	-30	1	-2	-29
40	40	40	8	-3	-8	-36	1	-4	-29
50	51	51	10	4	9	-50	0	6	-50

Table 10. Elbow angle measurements from flexion of the arm (All angles are in degrees).

Goniometer Joint Angle	Method 1			Method 2			Method 3		
	X	Y	Z	X	Y	Z	X	Y	Z
0	0	NAN	1	−1	0	0	−2	1	−80
10	10	NAN	10	−1	10	0	7	3	−80
20	21	NAN	21	−1	21	0	18	5	−79
30	31	NAN	31	−1	30	0	27	7	−78
40	40	NAN	40	−1	42	0	36	10	−77
50	51	NAN	51	−1	50	0	46	13	−74

4.2. Conclusions

Upon completion of the project, an IMU-based human movement monitoring framework has been provided that can be expanded to various possibilities beyond posture monitoring. As described in the paper, this monitoring system relies on real-time quaternion data streamed via IMUs to Unity. Once the accuracy of the IMUs was validated against Vicon motion analysis set up at Curtin University, three separate joint angle measurements were implemented and validated against the goniometer. The goniometer comparison demonstrated Method 2 as being the most accurate for the application area. The most accurate method was achieved by obtaining two quaternion angles from the child game object and the object that it is immediately attached to (the parent), and determining the difference in angle between the child body part and the parent body part. This difference is calculated by multiplying the inverse of the quaternion from the parent to the quaternion from the child, converted to Euler and displayed to the user. A calibration functionality was also implemented since IMUs will demonstrate inherent drift overtime.

There is scope for future work in the simultaneous joining of joint angle movement with full body posture monitoring such that rehabilitation exercises can be explored. Smoothness of motion can also be explored and included for further development in the accuracy of results that would be deemed useful in the medical field. One of the main advantages of the proposed system is that it does not rely on a specific type of IMU. As long as quaternion data can be read from the IMU, it can map to this framework. This has been made possible by moving all the calibration and joint angle measurement to the software. Additionally, since the software was developed in Unity, it can be easily ported to mobile platforms such as Android and Apple's IOS and open the possibility of remote training where clinical staff can provide remote guidance and advice while the sensors are worn by the workers on site.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/s22249618/s1>, Video S1: Full Interface Demo, Video S2: Menu System Demo, Video S3: Calibration Demo, Video S4: Sitting Live Demo, Video S5: Head Movement Live Demo.

Author Contributions: Conceptualization, I.M. and S.K.; data curation, S.P. and S.K.; formal analysis, S.K. and S.P.; investigation, S.K., S.P. and B.B.; methodology, S.K. and I.M.; software, S.P., S.K., J.H., H.B. and A.K.; supervision, I.M. and S.K.; validation, S.K., S.P., B.B., A.C. and J.H.; visualization, S.K., S.P. and B.B.; original draft, S.K., S.P. and B.B.; review and editing, I.M., A.C., S.K. and B.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki and approved by the Institutional Review Board (or Ethics Committee) of Curtin University Ethics Committee (Approval number HRE2021-0047 granted on 3 February 2021).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Curtin University's HIVE (Hub for Immersive Visualisation and eResearch) for providing feedback in the development of the Unity application. The authors would also like to thank Kevin Netto for providing feedback on the posture monitoring aspects of the project from a biomechanical point of view.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Work-Related Musculoskeletal Disorders & Ergonomics. Available online: <https://www.cdc.gov/workplacehealthpromotion/health-strategies/musculoskeletal-disorders/index.html> (accessed on 15 February 2022).
2. Australian Workers Compensation Statistics 2018-19p FINAL_2. Available online: https://www.safeworkaustralia.gov.au/sites/default/files/2021-01/Australian%20Workers%20%20Compensation%20Statistics%202018-19p%20FINAL_2.pdf (accessed on 20 February 2022).
3. Shultz, S.; Houghlum, P.; Perrin, D. Measuring Range of Motion This Is an Excerpt from Examination of Musculoskeletal Injuries 4th Edition with Web Resource. Available online: <https://us.humankinetics.com/blogs/excerpt/measuring-range-of-motion> (accessed on 8 March 2022).
4. Safe Work Australia. *Hazardous Manual Tasks: Code of Practice*; Safe Work Australia: Canberra, Australia, 2011.
5. What Is Motion Capture and How Does It Work? The Development of Mo-Cap. The Following Decades Saw Improvements on Harrison's Designs, with Bodysuits More Accurately Recording Movement. They Were Also Helped by What Does a Film Crew Do? Mo-Sys Launches Radical New Design for Camera Gyro-Stabilization Privacy-Terms. Available online: <https://www.mo-sys.com/what-is-motion-capture-and-how-does-it-work/> (accessed on 1 May 2022).
6. Xsense. What Is Motion Capture. Available online: <https://www.xsens.com/motion-capture> (accessed on 18 October 2022).
7. Vicom. Vicom Product Page. Available online: <https://www.vicon.com/> (accessed on 7 September 2022).
8. Thewlis, D.; Bishop, C.; Daniell, N.; Paul, G. A comparison of two commercially available motion capture systems for gait analysis: High-end vs. low-cost. In Proceedings of the Congress of the International Society of Biomechanics, Brussels, Belgium, 3–7 July 2011; pp. 1–2.
9. What You Need to Know about 3d Motion Capture. Available online: <https://www.engadget.com/2014-07-14-motion-capture-explainer.html> (accessed on 10 July 2022).
10. Fiber Optics: Understanding the Basics. Available online: https://www.photonics.com/Articles/Fiber_Optics_Understanding_the_Basics/a25151 (accessed on 10 July 2022).
11. Advantages of Fiber Optic Sensor, Disadvantages of Fiber Optic Sensor. Available online: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Fiber-Optic-Sensor.html> (accessed on 1 September 2022).
12. Williams, J.M.; Haq, I.; Lee, R.Y. Dynamic measurement of lumbar curvature using fibre-optic sensors. *Med. Eng. Phys.* **2010**, *32*, 1043–1049. [CrossRef] [PubMed]
13. Stoppa, M.; Chiolerio, A. Wearable electronics and smart textiles: A critical review. *Sensors* **2014**, *14*, 11957–11992. [CrossRef] [PubMed]
14. Sardini, E.; Serpelloni, M.; Pasqui, V. Wireless wearable t-shirt for posture monitoring during rehabilitation exercises. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 439–448. [CrossRef]
15. Filippeschi, A.; Schmitz, N.; Miezal, M.; Bleser, G.; Ruffaldi, E.; Stricker, D. Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion. *Sensors* **2017**, *17*, 1257. [CrossRef] [PubMed]
16. Fathi, A.; Curran, K. Detection of spine curvature using wireless sensors. *J. King Saud Univ. Sci.* **2017**, *29*, 553–560. [CrossRef]
17. Caputo, F.; Greco, A.; D'Amato, E.; Notaro, I.; Spada, S. Imu-based motion capture wearable system for ergonomic assessment in industrial environment. *Adv. Intell. Syst. Comput.* **2019**, *795*, 215–225. [CrossRef]
18. Bidabadi, S.S.; Tan, T.; Murray, I.; Lee, G. Tracking foot drop recovery following lumbar-spine surgery, applying multiclass gait classification using machine learning techniques. *Sensors* **2019**, *19*, 2542. [CrossRef] [PubMed]
19. Khaksar, S.; Pan, H.; Borazjani, B.; Murray, I.; Agrawal, H.; Liu, W.; Elliott, C.; Imms, C.; Campbell, A.; Walmsley, C. Application of inertial measurement units and machine learning classification in cerebral palsy: Randomized controlled trial. *JMIR Rehabil. Assist. Technol.* **2021**, *8*, e29769. [CrossRef] [PubMed]
20. Xu, C.; Chai, D.; He, J.; Zhang, X.; Duan, S. InnoHAR: A Deep Neural Network for Complex Human Activity Recognition. *IEEE Access* **2019**, *7*, 9893–9902. [CrossRef]
21. Jones, S. 5 Ways Bluetooth 5 Makes Wireless Audio Better. Available online: <https://pro.harman.com/insights/author/sjones/> (accessed on 1 September 2022).
22. Xsens DOT User Manual. Available online: <https://www.xsens.com/hubfs/Downloads/Manuals/Xsens%20DOT%20User%20Manual.pdf> (accessed on 9 September 2022).
23. Trident, B. Vicon Blue Trident. Available online: www.vicon.com/bluetrident (accessed on 9 September 2022).
24. Shimmer User Manual Revision 3p. 2017. Available online: <https://bmslab.utwente.nl/wp-content/uploads/2019/12/Shimmer-User-manual.pdf> (accessed on 15 September 2022).
25. QuantiMotion Full-Body Set-Bonsai Systems Store. Available online: <https://store.bonsai-systems.com/en/motion-capturing/9-quantimotion-full-body-set.html> (accessed on 15 September 2022).

26. Bunton, C. Water and Dust IP Ratings: What Does IP68 Actually Mean? Texas Instruments Minimize EMI with Capacitors. Available online: <https://www.pocket-lint.com/phones/news/138727-ip-ratings-what-do-they-actually-mean> (accessed on 15 September 2022).
27. Yan, W.; Zhang, Q.; Wang, L.; Mao, Y.; Wang, A.; Zhao, C. A modified kalman filter for integrating the different rate data of gyros and accelerometers retrieved from android smartphones in the gnss/imu coupled navigation. *Sensors* **2020**, *20*, 5208. [CrossRef] [PubMed]
28. Kalman Filter Tutorial. Available online: <https://www.kalmanfilter.net/default.aspx> (accessed on 10 October 2022).
29. Lacey, T. Tutorial: The Kalman Filter. Available online: <https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf> (accessed on 10 October 2022).
30. Extended Kalman Filter_Why Do We Need an Extended Version_by Harveen Singh Chadha_Towards Data Science. Available online: <https://towardsdatascience.com/extended-kalman-filter-43e52b16757d> (accessed on 10 October 2022).
31. Markley, F.L.; Crassidis, J.L. Space Technology Library Fundamentals of Spacecraft Attitude Determination and Control. Available online: <http://www.springer.com/series/6575> (accessed on 10 October 2022).
32. Ben-Ari, M. A Tutorial on Euler Angles and Quaternions. Available online: <http://www.ravvenlabs.com/uploads/1/1/8/4/118484574/quaternion-tutorial-2-0.pdf> (accessed on 15 October 2022).
33. Hughes, M. Don't Get Lost in Deep Space: Understanding Quaternions the Problem with Roll, Pitch, and Yaw. 2017. Available online: <https://www.allaboutcircuits.com/technical-articles/dont-get-lost-in-deep-space-understanding-quaternions/> (accessed on 15 October 2022).
34. Unity Real-Time Development Platform_3D, 2D VR & AR Engine. Available online: <https://unity.com/> (accessed on 20 October 2022).
35. Unity-Scripting API_quaternion.eulerAngles. Available online: <https://docs.unity3d.com/ScriptReference/Quaternion-eulerAngles.html> (accessed on 20 October 2022).
36. Unity-Scripting API_MonoBehaviour.OnAnimatorIK(int). Available online: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnAnimatorIK.html> (accessed on 20 October 2022).

4.5 USE CASE IN ALTERNATIVE CONTROL SYSTEMS FOR FLYING DRONES AS A PHYSICAL GAMIFIED EXERCISE



Article

Design and Evaluation of an Alternative Control for a Quad-Rotor Drone Using Hand-Gesture Recognition

Siavash Khaksar , Luke Checker, Bita Borazjan and Iain Murray

School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Bentley, WA 6102, Australia; ltc@checkers.org (L.C.); bita.borazjani@curtin.edu.au (B.B.); i.murray@curtin.edu.au (I.M.)

* Correspondence: siavash.khaksar@curtin.edu.au

Abstract: Gesture recognition is a mechanism by which a system recognizes an expressive and purposeful action made by a user's body. Hand-gesture recognition (HGR) is a staple piece of gesture-recognition literature and has been keenly researched over the past 40 years. Over this time, HGR solutions have varied in medium, method, and application. Modern developments in the areas of machine perception have seen the rise of single-camera, skeletal model, hand-gesture identification algorithms, such as media pipe hands (MPH). This paper evaluates the applicability of these modern HGR algorithms within the context of alternative control. Specifically, this is achieved through the development of an HGR-based alternative-control system capable of controlling of a quad-rotor drone. The technical importance of this paper stems from the results produced during the novel and clinically sound evaluation of MPH, alongside the investigatory framework used to develop the final HGR algorithm. The evaluation of MPH highlighted the Z-axis instability of its modelling system which reduced the landmark accuracy of its output from 86.7% to 41.5%. The selection of an appropriate classifier complimented the computationally lightweight nature of MPH whilst compensating for its instability, achieving a classification accuracy of 96.25% for eight single-hand static gestures. The success of the developed HGR algorithm ensured that the proposed alternative-control system could facilitate intuitive, computationally inexpensive, and repeatable drone control without requiring specialised equipment.

Keywords: alternative control; finger tracking; human computer interface (HCI); hand gesture recognition (HGR); media pipe hands (MPH)



Citation: Khaksar, S.; Checker, L.; Borazjan, B.; Murray, I. Design and Evaluation of an Alternative Control for a Quad-Rotor Drone Using Hand-Gesture Recognition. *Sensors* **2023**, *23*, 5462. <https://doi.org/10.3390/s23125462>

Academic Editor: Feng Jiang

Received: 13 April 2023

Revised: 26 May 2023

Accepted: 5 June 2023

Published: 9 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background

Alternative-control algorithms consist of two main components, a non-standard human-computer interface (HCI) and a command mapping algorithm [1–4]. An alternative-control algorithm is considered successful in its application if the alternative HCI extends upon the functionality offered by the conventional control medium. Within the literature, the degree of this success is commonly appraised against the following criteria: higher accuracy, ease of use without holding any equipment or instruments in hand, shorter user learning cycle, lower cost, offers capabilities that are not available in traditional interfaces, and computationally inexpensive [5].

This paper proposes the use of hand-gesture recognition (HGR) as an alternate HCI. Gesture recognition is the mechanism by which a predefined physical action made by a user is recognized by a system [6]. HGR has been an extensively researched topic over the past 40 years [5], resulting in a plethora of different viable approaches. Modern HGR applications use a machine-learning pipeline to achieve this recognition [5,7]. Within literature, this pipeline is defined to consist of four subcomponents: data-acquisition medium, gesture description, gesture-identification algorithm, and gesture-classification algorithm [7–9].

Sensors **2023**, *23*, 5462. <https://doi.org/10.3390/s23125462>

<https://www.mdpi.com/journal/sensors>

The application of HGR in alternative-control algorithms for drones has been a popular area of research for several years. Various studies have investigated different combinations of HGR subcomponents to optimise the process of recognising hand gestures and translating these gestures into drone actions. In [10], the authors employed sensor fusion between a mechanomyography band and a hand-mounted inertial measurement unit (IMU) to achieve robust control of an aerial drone using only the mechanical motion of the hand. In [11], the authors utilised a single RGB camera with marker gloves to recognise static gestures in combination with a hand-mounted IMU to recognise the dynamic motion of these gestures. In [12], the authors used a single RGB camera input paired with MPH and a long short-term memory neural network to achieve intuitive drone control that required no calibration or specialised equipment. In [13], the authors created a drone control simulation using a stereo camera (leap motion controller) as the primary input. In [14], the authors constructed a novel glove-based HGR system that also provided vibrotactile feedback to the system operator. The modern approaches cited above represent just a small subset of the HGR implementations that have been applied in the context of alternative control. These approaches vary in the selection of all four sub-components, the general taxonomy of known approaches for each HGR subcomponent is explained in greater detail under Section 1.2.

1.2. Existing Methods

1.2.1. Data-Acquisition Medium

The data-acquisition sources utilized by HGR algorithms can be defined into two governing categories, these being image-based approaches and non-image-based approaches [8]. The image-based category contains the following subcategories: marker, depth camera, stereo camera, and single camera. The non-image-based category contains the following subcategories: glove, band, and non-wearable. Non-wearable technologies have been omitted from this investigation as they are an emergent technology with limited implementations available [8,15]. Marker-based approaches have also been omitted from this investigation as they have been made largely obsolete by advancements in machine perception [5,8]. The remaining viable components are illustrated in Figure 1.

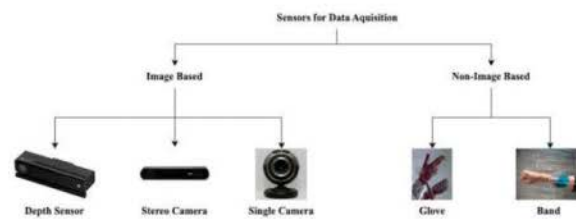


Figure 1. HGR data-acquisition categories.

1.2.2. Gesture Description

There are three aspects that form the gesture descriptor within existing HGR algorithms. These are the physiological scope of the gesture, the information interpreted from the gesture, and the model used to represent the gesture [16]. All three of these factors vary greatly between the HGR implementations detailed within the reviewed supporting literature [6,16,17].

Physiological scope refers to the pre-set taxonomy used to define the physical nature of the gestures [16]. The main distinctions that are made in the existing literature when defining this taxonomy are: the use of static or dynamic gesture set, the inclusion of wrist motion, and the number of hands that are used to form single gestures.

The information interpreted from gestures by HGR algorithms has three categories: spatial information, pathic information, and symbolic information [6]. Spatial information

refers to the position of the gestures within the environment. Pointing gestures are an example of gestures that convey spatial information. Pathic or temporal information is interpreted from the velocity and path that an observed gesture takes within an environment. Much like spatial information, this is typically observed from the world coordinates of the observed gesture. Symbolic information refers to the shape the observed gesture makes and is typically interpreted through joint angles' calculation or shape-matching techniques [6].

Within IIGR algorithms, the model used to represent an observed hand changes to reflect the desired scope of input gestures [16]. As the number, complexity, and information density of gestures increase, the complexity of the modelling method used must also increase. Model complexity is directly proportional to the number of classifiable landmarks that the model provides [16]. HGR modelling methods fall into two main categories: 3D-based models and appearance-based models. These models are demonstrated in Figure 2.



Figure 2. Hand-gesture modelling methods.

Less complex models such as the silhouette geometry model are preferably used for simple HGR applications as they offer very few classifiable landmarks. Given that it is an appearance-based model, it is computationally inexpensive to generate as it can be extracted directly from the image with little intermittent computation. These model styles are best suited for low-response-time algorithms that specialize in lightweight and fast operating applications. More complex models such as the 3D-skeleton model typically offer up to 21 landmarks [16,18,19] for classification. These models require considerably more computational power to generate accurately, but the 21 landmarks enable the calculation of exponentially more distinguishable gestures. Due to the higher computational load required to generate the model, they are typically employed in control applications where a higher accuracy and a more expansive data set are required [20].

1.2.3. Gesture Identifiers

Gesture identification is a catch all term that refers to the method by which a human hand is detected as apart from its background and transformed into a computer model used for classification [5,8]. This process is often referred to as feature extraction [21]. The model referenced is an estimation of the human hand, and the model type is as described in Section 1.2.2. The observational method used to collect the data from which the hand is detected is the data-acquisition component discussed in Section 1.2.1. As there are a multitude of different approaches for each combination of model and observation method, this review focuses on modern methods that utilize single-camera visual observation methods and 3D-skeleton representations [22].

Media pipe hands (MPI) is a complete and well researched on-device real time hand identification solution designed to operate using a single RGB camera [18]. The output produced is a list of 2.5D, 21-landmark skeleton models for each hand observed within the input frame. MPH utilizes a computationally efficient two-stage pipeline: the first stage is a palm detector, and the second stage is the hand-landmark extraction method. This pipeline was designed to minimize the computational load of 3D-skeleton identification in two key methods. The first method uses a computationally inexpensive algorithm to locate areas of interest within the image and then applies the landmark model only to these areas.

The second method uses the tracking of identified hands between frames to reduce the computation requirements necessary to perform identification of the subsequent frames.

InterHand2.6M (IHM) is a relatively new gesture-identification algorithm that utilizes a single RGB camera and a pre-trained convolutional neural network (CNN) labelled ResNet to achieve highly accurate feature extraction [19]. The output produced is a normalized 3D, 21-landmark skeleton model for up to two hands, specifically tuned to detect and correctly label the left and right hands of a single operator [19].

1.2.4. Gesture Classifiers

Gesture classification refers to the process by which a feature extracted by the gesture-identification algorithm is classified as a particular gesture from a pre-defined list [5,21]. The classification of input-gesture models is a typical machine-learning problem and can be addressed by numerous different algorithms. Popular approaches include decision trees, K-nearest neighbours (KNN), the hidden Markov model (HMM), artificial neural networks (ANNs), naïve Bayes (NB), linear regression, bounds-based classification, support vector machines (SVMs), and convolutional neural networks (CNNs). Modern HGR approaches favour the use of classifiers that can handle high-dimensional feature spaces and classify elements into many distinct non-linearly separable classes.

1.3. Contribution of the Paper

The primary contribution of this paper is the development of a cohesive, high accessibility, low-cost, alternative-control algorithm. This paper used a multi-stage method to identify, analyse, and clinically validate modern HGR components such as MPH. A cohesive alternative-control algorithm was constructed by using the results of these analytical stages to implement components that best complemented one another and satisfied the overarching design criteria of the paper. This set of complimentary HGR components operated at a high level of confidence and robustness against gesture confusion. The developed HGR algorithm had a gesture-classification accuracy of 96.25% over an array of eight input gestures, which is comparable to modern HGR algorithms [10–14]. The final alternative-control algorithm was demonstrated using a quad-rotor drone, whereby the algorithm was able to address its core developmental criteria and extend the functionality of the drone's conventional control medium. In comparison to modern alternative-control systems for drones, the final algorithm presented in this paper provides an increased level of accessibility, a higher computational efficiency, and a lower monetary cost. The increased level of accessibility was achieved due to the clinical validation of MPH, which led to the development of HGR systems that compensate for the Z-axis instability of the MPH model. This meant that users are no longer required to maintain ideal hand orientation in front of the input RGB camera, making the system easier to operate than other approaches that utilise MPH [12]. Furthermore, the final solution does not require specialised equipment, further increasing its accessibility and lowering its monetary cost [10,11,13,14]. Finally, the computational requirements of the final solution are minimal when compared to other vision-based HGR alternative-control algorithms [12,13].

The results achieved in this paper support the use of HGR algorithms such as MPH in alternative-control applications. Thus, the framework used to develop the alternative-control algorithm detailed in this paper could be re-applied to a multitude of other applications by simply reconfiguring the command mapping component. The secondary contribution of this paper is the novel validation of MPH modelling accuracy. These data can be used to inform future projects on how best to apply MPH to their respective applications even in project scopes that extend beyond alternative control.

2. Methods

2.1. Methodology Structure

2.1.1. Overview

The investigatory scope of this paper focused on using pre-existing gesture-recognition components to construct an alternative-control algorithm. This methodology section is focused on the selection, validation, and implementation of these standalone components to form a cohesive final algorithm. To achieve this, each standalone component was investigated, with subsequent investigations being adjusted to reflect the results of the previous stages. This overview describes the structure of the investigatory method, the governing criteria used to define each of the investigations, and the two key simplifications used to manage the scope of the overall investigation.

Alternative-control algorithms consist of an HCI component and a command mapping component. The HCI component selected for this project was HGR. From the literature reviewed in Section 1.1, all HGR algorithms consist of the following four components: gesture description, data-acquisition method, gesture identifier, and gesture classifier. With the inclusion of a command mapping component, the list of components required by the final solution was derived. The subsequent investigation used to select a final implementation for each of these components was broken up into six stages. The first stage was the selection of a gesture-description model. The second stage was the selection of a data-acquisition method. The third stage was the selection of a gesture-identification algorithm. The fourth stage was the validation of the selected gesture-identification algorithm. The fifth stage was the selection of a gesture-classification algorithm. The sixth and final stage was the derivation and tuning of a gesture-mapping component. The general structure of the investigation is shown in Figure 3.

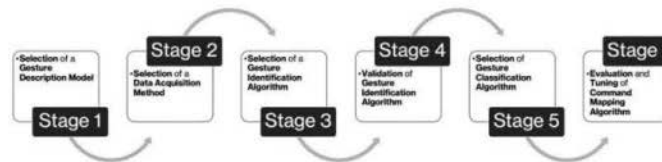


Figure 3. Investigation structure flow chart.

2.1.2. Defining Simplifications

As demonstrated in Section 1.2, due to the mature nature of HGR, there were numerous unique approaches available for each of the four algorithm subcomponents. Given the expansive scope of viable approaches, it was not feasible to investigate all possible subcomponent combinations directly. Consequently, to maintain the validity of the investigation, three simplifications were defined to manage the scope of the investigation:

1. Components selected prior to the current stage of the investigation cannot be changed. Implementations were only considered if they were applicable with the previously selected components. Selected components should not be changed to accommodate for the needs of a new proposed approach. For example, the data-acquisition method selected in stage two could not be changed to accommodate for the requirements of a gesture-identification component proposed in stage three.
2. The selection of each component was to be made without consideration for future components. The selection of each component was to be based on the applicable governing criteria, selection of previous components, and the relevant validation results.
3. The gesture-description and data-acquisition components were selected from the reviewed literature, without any new quantitative or qualitative analysis being performed.

Simplifications 1 and 2 enforced a linear investigation structure. By using this linear method of selection, the number of applicable implementations reduced exponentially with

each subsequent stage of the investigation. This reduced the scope of each investigation stage to a workable level whilst maintaining the validity of the overall investigation. However, these assumptions only worked to reduce the scope of the later stages of the investigation. Hence, the inclusion of a third simplification was required to reduce the scope of the earlier stages.

The third simplification removed the need for time-consuming experimental analysis to be performed during the first two investigatory stages. This reduced the workload required to analyse different implementations, allowing for a wider array of implementations to be investigated. This simplification did not compromise the validity of the overall investigation for two key reasons. The first reason is that it is difficult to qualitatively or quantitatively analyse the effectiveness of different gesture-description and data-acquisition methods without considering the HGR algorithms that are typically included within. The most effective way to analyse complete solutions is to review the literature used to define them, hence supporting the selection of these components following a literature review. The second key reason is that the criteria relative to these two components are not defined by analysable metrics, and as such can clearly be derived from the attributed literature in a yes or no fashion.

By utilising Simplification 3 to inform the selection of the highest-level components (Components 1 and 2), and then enforcing Simplifications 1 and 2, the scope of each individual section can be managed appropriately—facilitating an efficient derivation of the final solution. Furthermore, the application of Simplification 3 enables the selection of Components 1 and 2 within Sections 2.2 and 2.3 respectively, as no experimental investigation was needed. This allowed for more specific testing to be defined for subsequent sections.

2.1.3. Governing Criteria

To ensure that a cohesive and effective final solution was developed over the course of this investigation, a governing set of criteria were developed. The purpose of these criteria was to augment the dependent and independent variables used within each investigatory stage, in a way that led to the development of an effective alternative-control algorithm. That is, if the implementation selected for each component is selected because it conforms best to the defined criteria, then the final solution will function as an effective alternative-control system. The selected criteria were a subset of the list found in Section 1.1 of the literature review. These criteria were selected from the larger list to specifically adhere to the design approaches used in [3,4] which successfully produced enervative and effective solutions. The final set of governing criteria used in this project are listed below in order of importance:

1. Reliability in issuing the intended command: This criterion graded both the number of unique commands the algorithms can issue, and the algorithm's ability to distinguish between these unique commands.
2. Reproducibility of the intended command: This criterion graded the algorithm's ability to robustly reproduce the same action when presented with the same user input.
3. Physically non-restrictive equipment or instrumentation: This criterion graded how restrictive the algorithm's control interface was, referring to both restrictions caused by elements that were physically placed upon the user's body or elements that required the user to operate within a restricted space or in a restricted manner.
4. Ease of operation and shorter user learning cycle: This criterion graded how complex or difficult to learn the control interface was, considering factors such as complexity of inputs, complexity of input structure and physical difficulty to form inputs.
5. Computationally inexpensive: This criterion graded the computation requirements for the algorithm's operation and inversely the speed in which the algorithm could operate at if given an abundance of computational resources.
6. Monetarily inexpensive: This criterion graded the general cost needed for the algorithm to function, including the cost of the computational hardware required to run the algorithm and the sensory hardware to acquire data.

2.2. Stage One: Selection of Gesture-Description Model

2.2.1. Overview

The first stage of the investigation selected a gesture-description method that could facilitate specific and repeatable control of a complex system. This stage was affected by Simplification 3, which specified that the investigation type was a review and not a quantitative or qualitative investigation. As stated in Section 1.2.2, the gesture-description component of an HGR algorithm consists of three subcomponents: gesture type, gesture information, and gesture model. All three subcomponents were evaluated individually and selected per Governing Criteria 1, 4, and 5.

Due to the relevant simplification, the method used to determine the final implementation for these three subcomponents was a literature review. The options to be reviewed for each of the subcomponents were derived directly from the reviewed literature and are listed under each of the subcomponent's justification sections.

The final values selected for these fields are as follows: for the gesture type, single-hand static gestures were chosen; for the gesture information, symbolic information was selected; for the gesture model, a three-dimensional, 21-landmark skeleton model was selected. These values were selected for two key reasons. The first was to ensure that the final solution met the relevant governing criterion, and the second was to ensure that the simplest solution to these requirements was found.

2.2.2. Gesture-Type Selection Justification

The type of gestures observed had three main considerations to be analysed. The first of these was the motion of observed gestures, either a static or dynamic gesture set. The second was the scope of observation, specifically the inclusivity of wrist motion. Finally, the scope and number of observable gestures was considered.

Static single-hand gestures were selected to ensure future components' simplicity and ease of understanding for operators. While dynamic or two-handed gestures also conform to Criterion 4, the construction of an algorithm that operates using these gestures would have been considerably more complex than an algorithm designed to recognise static gestures [16]. Furthermore, single-hand static gestures are easier for an operator to learn and perform consistently, decreasing the user learning cycle in comparison to the more complex dynamic or two-hand gestures [16]. Thus, single-hand static gestures were selected to simplify the computational restrictions and user learning cycle of the final solution.

A wide range of gestures was initially proposed to avoid artificially biasing the identification and classification components' selection by providing an ample array of gestures. The only restriction on the initial gesture set was that it had to contain gestures defined by a recognised sign-language system. This was performed to ensure that the selected gestures were easily recognisable and easily learned [22], aiding in the final solution's conformity to Criterion 4. An example gesture set is shown in Figure 4.

2.2.3. Gesture Model Selection Justification

The gesture-model selection was governed by two key factors, the computational complexity required to generate each model and the number of classifiable landmarks offered by each model. This selection aimed to balance these two factors by selecting a modelling method that allowed for enough classifiable landmarks to differentiate between the types of gestures detailed in Section 2.2.2, whilst not requiring excessive computational power to generate. The modelling methods analysed are listed in Section 1.2.2. Given the number and complexity of the possible gestures, the most applicable model was a 3D-skeleton model [18,19]. Appearance-based models were not applicable as their low number of classifiable landmarks would limit the final algorithm's ability to differentiate between the desired input gestures [16]. More complex 3D models, such as 3D geometric models and 3D textured volumetric models, were not applicable as the additional classifiable landmarks they offer are not necessary to differentiate between the desired input

gestures. Thus, the use of these models would needlessly increase the complexity of future components without benefitting the final algorithm's performance [16].



Figure 4. Sign language example hand-gesture set [23].

2.2.4. Gesture Information Justification

The information derived from the gestures had four considerations, these being spatial, pathic, symbolic, or affective [6]. Note that the selection of these information sources was not mutually exclusive, i.e., one or all of them could be selected. Given the gesture type selected in Section 2.2.2, symbolic information was the primary source of information that was extracted from the observed gestures [6]. Additionally, the spatial information of the three-dimensional skeleton landmark model was also used to calculate the joint angles for each of the 15 observed joints. Stage four methodology will define the specific calculations required to perform this transformation.

2.3. Stage Two: Selection of Data-Acquisition Method

The second stage of the investigation was the selection of a data-acquisition method. The purpose of this stage was to select a data-acquisition method capable of efficiently and non-restrictively observing a human hand in a manner conducive to the production of the selected gesture model. Similar to Section 2.2, the investigation process for this stage was a literature review as defined by Simplification 3. The governing criteria relevant to this section were Criteria 1, 2, 3, 5, and 6. To ensure that these criteria were satisfied, this review assessed all of the HGR data-acquisition methods listed under Section 1.2.1. Each of these solutions were analysed against the criteria listed above and compared against one another to find an optimum solution.

Out of the analysed data-acquisition methods, single RGB cameras were the only analysed approach that satisfied the applicable governing criteria. In contrast, the other data-acquisition methods all posed notable drawbacks that would severely hinder the final solution's ability to satisfy these criteria. Specifically, depth cameras were omitted due to their range and availability restrictions, which would jeopardise the final solution's conformity to Criterion 3 as the range restrictions will restrict users [8,24]; stereo cameras were omitted due to their extensive computational requirements and focal pointing restrictions, which would make satisfying Criteria 3 and 5 difficult [7,8,13]; and band and glove approaches were both omitted because of their direct opposition to Criterion 3 [8,10,11,14,15].

After these omissions, the single RGB camera was the only remaining viable approach. However, single-camera approaches have some notable drawbacks that will need to be addressed by future components. These are primarily the robustness issues associated with background and operator hand, variability, and single viewpoint error sources such as self-occlusion and transform inconsistency [8]. Despite these notable drawbacks, due to the mature nature of this form of HGR [7], it is reasonable to assume that the selection of appropriate future components can appropriately manage these drawbacks [7,12].

2.4. Stage Three: Selection of Gesture-Identification Algorithm

The purpose of stage was to select a gesture-identification algorithm capable of extracting hand features from the data returned by a single RGB camera. The extracted features were to be arranged in the form of the desired three-dimensional skeleton model. The governing criteria relevant to this section were Criteria 1, 2, and 5. Additionally, the prospective algorithms were also investigated as to their ability to minimize the drawbacks of single RGB camera approaches, such as self-occlusion. A qualitative analysis was performed to facilitate this selection, focusing on the computational cost of the implementation and the observable localization accuracy of the prospective algorithms. This stage was only intended to be a minor thresholding investigation, aimed less at comparing applicable solutions and more towards ensuring the selected solution will be able to conform to the governing criterion.

Given that there are a multitude of feature-extraction methods that are applicable for HGR gesture identification, it was not feasible to test them all directly. Fortunately, this expansive scope was reduced considerably by the three design simplifications. The previously selected gesture-description and data-acquisition components reduced the scope of this investigation in the following ways: the removal of any method not initially developed to return a three-dimensional skeleton model; the removal of any gesture-identification approach not compatible with single RGB camera data; and only considering pre-existing open-source implementations. After these reductions, three solutions were marked for future investigation: media pipe hands, InterHands2.6M, and an OpenCV approach. These three algorithms represented possible solutions that applied different pre-processing and feature-extraction techniques and had drastically different computational loads.

The key dependent variables of this qualitative investigation were the localisation accuracy of the skeleton model and the computational requirements needed to perform feature extraction. These two dependent variables were analysed in two subtests. The first test aimed to observe the computational requirements to set up and operate the three algorithms. The second subtest aimed to observe the localisation capacity of the three algorithms in variable environments.

1. Identifier Implementation: The aim of this subtest was to implement a baseline variant of the three algorithms. The baseline variant of this method should be capable of observing a single human hand and printing the angle of its 15 primary joints to the terminal while also displaying the 3D-skeleton model on screen. The method used to calculate these joint angles is described in Section 2.5. The purpose of this stage is three-fold. Firstly, it serves to assess the operational readiness of the algorithms. Secondly, it assesses the computational requirements necessary to implement the algorithms. Finally, it acquires an operational version of said algorithms upon which future testing would be performed. The key independent variables of this test are the three different algorithms being tested. All algorithms are to be applied on the same 2017 Mac Book Pro that operates using a 3.5 GHz Dual-Core Intel Core i7 CPU, an Intel Iris Plus Graphics 650 1536 MB graphics card, 16 GB 2133 MHz LPDDR3 of RAM, and 250.69 GB of storage.
2. Qualitative Analysis: The aim of this subtest was to qualitatively observe the implemented algorithms' localisation accuracy. This subtest was the first step towards ensuring that the selected algorithm conforms to Criteria 1 and 2 and minimizing the drawbacks of the selected data-acquisition method. This investigation stage aimed to observe each algorithm's accuracy in cases of self-occlusion, rotation, and translation using the operational version derived in the first subtest. The method for this observation was relatively simple. First, a user's hand was held in a constant position in front of the camera. The displayed three-dimensional model was then recorded. From this position, the hand was then rotated and translated around the camera's viewport. While these rotations and translations occurred, the displayed model was constantly observed. These observations aimed to determine whether the algorithm could maintain its localisation accuracy despite the movement. The final stage of the

observation was to turn the hand so that certain aspects of the hand become occluded from the camera's POV. This was conducted to determine whether the algorithm could still produce a model despite the occlusion of hand features.

2.5. Stage Four: Validation of Selected Gesture-Identification Algorithm

The fourth stage of the investigation focused on the validation of the selected gesture-identification algorithm. Specifically, this stage centred around the evaluation of the accuracy and robustness of the model produced by the selected algorithm. This stage employed a clinically advised, quantitative approach that compared the joint angles derivable from the model with the joint angles measured with a finger goniometer. The results of this method were then used to determine the current algorithm's conformation to Governing Criterion 2. Due to the extensive nature of this validation process, it was not possible to apply it to each of the gesture-identification algorithms analysed in Stage 3; hence, it was only applied to validate the final selected identification algorithm.

The key dependent variable observed during this investigatory stage was the percentage accuracy of the generated model. The generated model, as specified in Stage 1, was a three-dimensional skeleton hand model. The accuracy of the model was found by calculating the percentage variance between the joint angles of the observed hand measured by a goniometer and the joint angles calculated from the generated model. Joint-angle comparison was used over other possible methods such as landmark-accuracy analysis or joint-positional analysis because of the clinical support available for joint-angle measurement. As a result of this, the joint angles could be measured directly and accurately using a clinically defined method which provided an excellent reference value for comparison with the model. In contrast to this, if landmark-accuracy analysis or joint-positional analysis methods had been applied, considerable sources of error could have been introduced into the reference value due to sources such as hand-size variation, joint-location variation, and joint-observation variation.

The joint angles to be tested are the joint angles of the metacarpophalangeal, proximal interphalangeal, and the distal interphalangeal joints of all fingers including the thumb. For all joints' measurements, the static arm of the finger goniometer was to be stabilized against the proximal side of the joint with the hinge of the goniometer being placed directly above the observed joint. If the participants knuckle was bulbous in nature such that it prevented the goniometer from securely sitting above the joint, then the goniometer was to be moved to the side of the finger, such that the goniometer's hinge sat directly in front of the observed joint. Once secured, the free arm of the goniometer was then lightly pressed against the distal side of the observed joint. It is imperative that little to no force is applied during this process as the goniometer is designed to move freely, and any excessive application of force could alter the pose of the observed hand. Once the free arm of the goniometer had contacted the distal side of the joint, the joint angle could then be recorded to the nearest 5°. This method was advised by Jayden Balestra [25].

Two methods that were applied to calculate the joint angles from the three-dimensional model. The first method analysed was a conventional three-dimensional vector angle calculation which first created two vectors: one traveling to the desired joint from the previous attached joint and a second vector traveling from the joint to the next joint. Once these vectors had been defined, a simple dot product calculation was then applied to calculate the angle that existed between the two joints. A second method was used as a backup, which simply ignored the depth component of the model and then performed the same calculation performed above. This method was included to quantify whether the three-dimensional nature of the model was aiding or limiting the model's performance. The code used to perform both calculations is shown in Appendix D.

In order to thoroughly test the robustness of the algorithm alongside its accuracy, two independent variables—hand pose and hand orientation to camera—were changed throughout the course of this stage:

1. Hand pose: The first of these variables was the pose of the hand. In total, three positions were investigated: a fully closed position, a partially closed position, and a fully open position. These three positions were selected because they are stable, easy to hold, repeatable positions, and again, because they were the advised positions suggested by our clinical reference, Jayden Balestra [25]. Furthermore, these three positions were used to simulate a full range of motion of the human hand, as it was important to validate the accuracy of the model across a hand's full range of motion. An example of the three poses used are shown in Appendices A and B.
2. Hand orientation with respect to camera: The second independent variable that was altered over the course of this analysis was the incident angle formed from the camera's point of view and the observed hand. By changing this angle of orientation, the algorithm's robustness against rotation and self-occlusion (the drawbacks of single camera RGB solutions) could be quantitatively observed. For each of the three hand poses defined above, four photos were taken: one from directly in front of the hand, one from a 45° offset, one from a 90° offset, and one from a 180° offset. An example of the four viewpoints used are shown in Appendices A and B. To ensure a high level of accuracy within the test itself, a wide range of controls were put in place, to make sure each stage of the analysis was repeatable and accurate.
3. Lighting: To avoid lighting variance, all tests were to be conducted in a well illuminated environment, specifically aiming that no shadowing be present on the observed hand.
4. Background: Background variation is known to have an impact on the MPH modelling process. As this is not a factor currently being analysed, a white backdrop was used for all tests. A white backdrop was used to ensure that there was a high level of contrast between the hand and the background to aid in the feature-extraction process.
5. Pose stability and body position: To ensure that the same position and viewpoint angles were observed for each participant, two controls were put in place to manage body position and hand stability. The first control is that participants are to kneel in a comfortable position, with their forearm braced against the test bench. The test bench is to contain a set of marks, indicating the appropriate positions for the background and participant forearm.
6. Pitch, roll, and yaw of the camera: To ensure that the viewpoint orientation was maintained across all participants, and only varied by the desired amounts between tests, the Halide camera application was used [26].
7. General hand size/distance from camera: Whilst changes in participant hand size were unavoidable, to avoid exacerbating these variations, a fixed camera distance was used for all participants. This was performed by simply having fixed mounting points for the camera on the test bench at the correct location and orientation for each of the desired viewpoint angles.

The final testing procedure consisted of five stages; (i) establish the aforementioned controls; (ii) the participant forms required hand pose; (iii) record the joint angles using goniometer; (iv) photograph the hand from the required viewpoints, (v) re-measure the joint angles using a goniometer. After the above procedure had been completed, the two sets of measured joint angles were compared. If the results of the second set of measurements failed to match the first, the test images were discarded, and the process was repeated. This was conducted to confirm that the participant's hand pose had remained stable throughout the test. This process was repeated for each participant and for each pose. A five-minute pause was taken between tests to ensure that participant fatigue did not affect pose stability. The valid photos were passed to the selected gesture-identification and angle-calculation algorithms, which generated a set of observed joint angles (the code used to perform this stage of the process is referenced through Appendix D). These observed joint angles were then compared against the goniometer measurements to generate a final set of accuracy percentages.

2.6. Stage Five: Selection of Gesture-Classification Algorithm

The purpose of the fifth stage of the investigation was to use a quantitative method to select the gesture-classification algorithm that best complimented the selected gesture-identification algorithm. The three relevant criteria for this stage are Governing Criteria 1, 2, and 5. From these three criteria, two quantitative metrics were calculated to inform the final selection. These metrics were the classification accuracy of the tested algorithms expressed in the form of confusion matrices and classification speed expressed in seconds (used to reflect the computational requirements of the algorithms).

To ensure that the selected classifier complimented the selected gesture-identification component, there were two possible scopes and criterion weightings defined for this stage of the investigation. Each of the defined scopes was focused on a different possible outcome which could have arisen from the results of Stage 4. Before defining the individual scopes, there was another key scope reduction that applied to both cases. The selected classifier must be a pre-trained solution, capable of classifying gestures of a globally recognised sign-language. This reduction was made to conform with the paper problem statement, and the selected gesture-description model. These two possible scopes are defined below:

1. If the Stage 4 results show that the selected gesture-identification algorithm can accurately and robustly produce a model that reflects the user's hand, then a low dimensionality classifier built around the 15 single-dimension joint angles should be investigated. The final selected algorithm is that which favours Criterion 5 over Criteria 1 and 2. The algorithms to be investigated are decision trees, KNNs, and linear regression [27].
2. If the Stage 4 results show a less than ideal model accuracy, then a higher input dimensionality classifier which uses the original 21 three-dimensional coordinate system (63 total dimensions) would be investigated. The final selected algorithm is that which favours Criteria 1 and 2 over Criterion 5. Specifically, the algorithms to be investigated are ANNs, SVM, linear regression, and a non-machine learning bounds-based approach [27,28].

Regardless of the selected scope, the experimental method that would be used to analyse the prospective classifiers remained the same. In either case, the dependent variables of the investigation remained the classification accuracy and classification time. The independent variables of this classification were the style and implementation of the classifiers themselves. To ensure a fair investigation of the defined classifiers, the following variables were kept constant:

1. Test data set: A custom data set was to be made for the selected classifiers. Ideally, after the initial group of prospective classifiers had been defined, a common set of ten gestures would be identified between the algorithms. Once this common gesture set had been defined, ten images were created for each gesture and converted into three-dimensional models using MPH. These 100 models formed the test data set for this stage of the investigation. Note, to ensure that Criteria 1 and 2 were assessed correctly, the hands present in the ten selected images varied, in scale, orientation, and pose. By introducing these variations into the common data set, the algorithm's accuracy will be tested in a more robust fashion as they are not being tested in a 'best-case scenario'.
2. Computation power provided to each algorithm: To ensure that no one algorithm is favoured during this analysis process, all testing should be performed on the same device, with no background processes running. When performing classification-time testing, the time taken should only be considered for the 'prediction stage' of the classifier. Specifically, this time value should exclude the time taken to initialize/train the classifier, load the MPH model, and any time associated with the creation of the confusion matrices.

The procedure for this investigation was relatively simple, a basic algorithm was used to sequentially test each of the prospective pre-trained algorithms against the common

data set. After each test, the prediction of the four classifiers was recorded in the respective confusion matrices, and the time taken to perform that classification was stored in a CSV file. Once all the test images had been fed into the algorithm, the final confusion matrices, and time performance data were displayed on screen for evaluation.

2.7. Stage Six: Gesture Mapping and Tuning

The purpose of this stage of the investigation was to develop a gesture-mapping component capable of translating classifiable gestures into drone actions. The mapping method selected for this component was a one-to-one command mapping approach. This approach was selected to conform with the design decisions made in [1–3,29] and served as a good initial solution capable of demonstrating the functionality of the fully developed HCI algorithm within the context of alternative control. This stage consists of two key developmental sections. The first section was the initial declaration of a gesture dictionary that translated observed gestures into commands. The second section was the tuning of these commands to allow for smooth control of the drone. The primary relevant criterion for this stage is Governing Criterion 4.

The DJI TELLO quad-rotor drone (mechanism specifications provided in Appendix E) was selected to be controlled by this mapping component for three key reasons. The first reason is that the physical characteristics of the drone made it well suited for use in prototype implementations such as this. The drone is inexpensive, light, and slow and has built-in collision recognition sensors and systems. These collision-mitigation sensors limited the consequences experienced during testing, which was beneficial given the experimental nature of the applied control algorithm. The second reason is that while DJI TELLO does not offer a python API, the mobile application offered by DJI to operate the drone has commands that can be easily replicated using Python's built-in socket library. This meant limited work was required to transfer controlling commands to the drone. The final key benefit is the breadth of commands offered within the TELLO app. The TELLO app offers numerous distinct commands ranging from simple operations, such as lift-off and land, to complex tasks such as 'do a barrel roll'.

The TELLO drone accepted two basic movement command sets, each containing six unique commands. The first command set uses positional commands that move or turn the drone by a fixed amount per transmitted command. The second command set uses velocity commands, whereby each command updates the drone's velocity in a certain way. With the addition of the take-off and land commands, two unique sets of eight commands were defined for investigation. In either case, the eight commands were mapped to the eight gestures, most accurately classifiable by the completed HGR algorithm. Once this mapping was complete, three key factors had to be experimentally tuned through a set of test flights. These factors were: the time a gesture must be held before a command is executed, the magnitude of a response once a command is executed, and the refresh rate for command execution. The final command set and values for each of the above factors were selected because they facilitated the control that best conformed to Criterion 4.

3. Results

3.1. Gesture-Identification Selection

3.1.1. Implementation Results

Three algorithms were considered and attempted to be implemented; these were media pipe hands (MPH), InterHand2.6M, and an OpenCV extension of MPH titled CVZ. However, due to the computational requirements to both train and operate, InterHands2.6M was removed from further analysis during this stage. The training data required a 365-GB download to attain all the necessary data to train the ResNet network for 30-fps operation. The 365-GB download exceeded the total storage capacity of the host machine and as such could not be completed.

MPH was successfully implemented using its open-source solution that included APIs for both java and python [30]. The python API was used for this project. As mentioned

in the Section 1.2.3, the computational requirements of MPH are minimal, as such it was expected that MPH would be able to run at a high frame rate on the selected host machine. MPH conformed to this expectation, continually maintaining its capped frame rate of 30 fps, validating its conformity to Governing Criterion 5. The output of this implementation is shown in Figure 5.

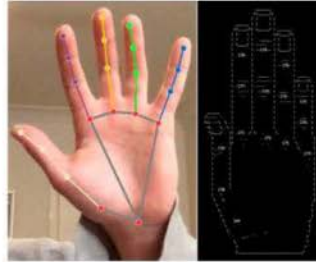


Figure 5. MPH output displaying the 21 coloured landmarks overlaid onto the input image and the text based joint angle measurement model displayed through terminal.

The joint-angle calculations were handled by a python module that applied the calculations described in Section 2.5. This module parsed the ‘hand’ objects produced by MPH to acquire the 3D coordinates of the joints. These 3D coordinates were converted to joint angles using the aforementioned calculations before displaying these angles using the ascii art hand shown in Figure 5. An ascii art hand was used to visually display the joint angles in the terminal, in order to ensure that they could be easily interpreted by the user.

CVZ [31] was successfully implemented using its python API. Despite its lack of supporting literature, CVZ was relatively easily to implement. CVZ only required one variable to be set which was the webcam input source directory. Once operational, CVZ’s performance appeared to fluctuate greatly with frame rates ranging from 16 fps to 30 fps. Unfortunately, CVZ also suffered from some considerable stability issues and had a re-occurring bug that would crash the code repetitively whenever two hands appeared in-frame together. The initial output of CVZ is shown in Figure 6.

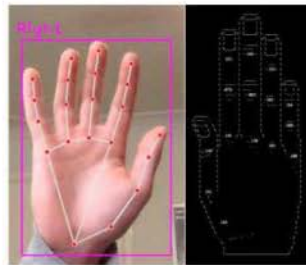


Figure 6. CVZ output displaying the 21 red landmarks overlaid onto the input image and the text based joint angle measurement model displayed through terminal.

As CVZ had the same 21-landmark model structure as MPH, only minor changes were needed to be adapt MPH’s joint-angle calculation python module to match CVZ. The only changes required were to adjust the code pair with CVZ’s landmark data structure. After these changes, the joint information was able to be successfully calculated and displayed as shown in Figure 6.

The computational performances of CVZ and MPH were also observed during this stage. Each algorithm's computational performance was observed by recording the refresh rate of the algorithm's identification component. Specifically, this was achieved by measuring the number of output models returned per second. The output rate was measured with both one and two hands on screen. MPH maintained a refresh rate of 13.22 outputs per second with two hands in frame and 14.04 with one hand in frame. CVZ maintained an output rate of 14.1 with one hand in frame and, as mentioned, would crash whenever two hands came into frame.

3.1.2. Qualitative Analysis Results

Out of the three algorithms, two had been successfully implemented, these being MPH and CVZ. The results of the qualitative analysis of both algorithms are detailed below.

Resistance to Translation

Both MPH and CVZ were exposed to a simple translation of a closed fist, facing the camera, around the camera's frame. This involved movements both toward and away from the camera (changing scale) and movements vertically and horizontally across the image frame (traditional translation). Neither algorithm showed signs of major landmark deviation or angular fluctuation during this test.

Resistance to Rotation

Both algorithms were presented with a closed fist with the palm facing the camera, the fist was then rotated about the axis along the operator's forearm. Throughout this rotation, both MPH and CVZ appeared to maintain a high degree of localisation accuracy, as the displayed landmarks never deviated from their respective joints. Despite this, the joint angles being displayed to the terminal did fluctuate greatly. This suggests that whilst the feature-extraction/landmarking method used by both algorithms is robust against rotation, the modelling method used may not be. This demonstrates that neither algorithm is entirely robust against rotation, which is an issue that may need to be addressed by the classification component.

Resistance to Self-Occlusion

In this test, the hand was held with the palm directly facing the camera, being positioned directly along its axis of reception. Then, the hand was slowly tilted forward until the fingers were directly facing the camera and the palm was completely obscured by the fingers. MPH performed relatively well during this test, as its gesture-tracking component was able to keep an understanding of the hand's position even after the palm had become completely obscured by the fingers. MPH was also able to maintain sensical joint readings throughout this entire process. CVZ, on the other hand, did not fare as well during this test. Once the hand was approximately 15° off its final position, the bounding box (pink box displayed in Figure 6) raised an exception within the code, crashing the observation algorithm. Thus, the conclusion of this third test was that MPH demonstrated robustness toward self-occlusion and, in its present form, CVZ did not.

3.1.3. Final Selection

From the results detailed in Sections 3.1.1 and 3.1.2, MPH was selected as the final gesture-identification component. MPH was selected because it was operationally stable, the least computationally intensive, and demonstrated a high degree of robustness when presented with translation and self-occlusion. These results demonstrate the potential that MPH has to aid in the satisfaction of Criteria 1, 2, and 5.

3.2. Gesture-Identification Validation Results

3.2.1. Measured Joint Angles

Tables 1–3 display the average joint-angle readings recorded for each of the three poses. These joint-angle readings are the averaged results taken from the three participants that were analysed during this study. The tables are organised to display the measured angles for each joint, metacarpophalangeal (J1), proximal interphalangeal (J2), and distal interphalangeal (J3), for each finger.

Table 1. Open-palm joint angles measured by goniometer.

Joint Number	Thumb	Index	Middle	Ring	Pinkie
J1	162	178	175	178	178
J2	177	172	172	170	172
J3	180	177	180	180	180

Table 2. Partially closed palm joint angles measured by goniometer.

Joint Number	Thumb	Index	Middle	Ring	Pinkie
J1	153	168	168	170	173
J2	157	93	92	87	97
J3	118	113	107	117	113

Table 3. Closed-palm joint angles measured by goniometer.

Joint Number	Thumb	Index	Middle	Ring	Pinkie
J1	148	98	98	107	107
J2	143	82	83	82	90
J3	117	110	108	108	112

3.2.2. Calculated Joint Angles

Tables 4–15 display the joint angles calculated from MPH. The angles displayed are the average angles that were returned from the three participants that completed testing. The tables are broken up into three groups, with each group containing the angles recorded for that pose. Within each group, there are four tables, with each table containing data calculated from one viewpoint. Each table contains the joint angles calculated using both the two-dimensional and three-dimensional methods of calculations. The tables are organised to display the calculated angles for each joint, metacarpophalangeal (J1), proximal interphalangeal (J2), and distal interphalangeal (J3), for each finger.

Table 4. Closed-palm joint angles calculated from the MPH model—front.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	151	156	142	142	144	144	142	140	132	128
J2	151	151	53	54	47	48	34	35	40	41
J3	105	100	164	161	127	115	136	128	147	143

3.2.3. Final Accuracy Percentages

As can be seen in Tables 16–19 below, the data in Sections 3.3.1 and 3.3.2 were compared and condensed to generate two separate sets of accuracy data. The tables below display the

average and minimum accuracy percentages across each viewpoint and across each finger. The tables were further reduced to generate final set of figures for the 3D accuracy, which had an average accuracy of 86.7% and a minimum accuracy of 41.5%, and the 2D accuracy, which had an average accuracy of 83.8% and a minimum accuracy of 9.8%.

Table 5. Closed-palm joint angles calculated from the MPH model—45°.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	140	138	133	157	123	133	112	112	91	85
J2	144	150	78	63	73	63	66	67	85	69
J3	147	148	133	132	120	123	120	120	118	111

Table 6. Closed-palm joint angles calculated from the MPH model—side.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	143	149	74	76	73	73	76	75	83	82
J2	184	163	107	103	95	94	92	93	95	95
J3	163	163	113	108	113	113	114	113	113	110

Table 7. Closed-palm joint angles calculated from the MPH model—back.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	159	161	125	135	121	155	116	148	108	118
J2	154	151	75	75	90	50	97	37	94	51
J3	105	105	163	170	156	173	153	170	187	163

Table 8. Partially closed-palm joint angles calculated from the MPH model—front.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	152	154	145	156	154	164	151	176	156	160
J2	161	161	119	97	99	60	135	75	110	94
J3	168	165	124	98	138	132	126	116	122	115

Table 9. Partially closed-palm joint angles calculated from the MPH model—45°.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	141	142	145	166	158	171	165	170	151	151
J2	164	167	100	87	85	76	78	74	92	88
J3	132	132	129	126	127	124	127	128	134	135

Table 10. Partially closed palm joint angles calculated from the MPH model—side.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	145	134	111	121	113	114	116	115	110	115
J2	142	146	113	113	107	106	104	104	121	122
J3	161	160	139	132	133	132	135	134	145	146

Table 11. Partially closed palm joint angles calculated from the MPH model—back.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	160	163	151	159	162	169	153	159	144	150
J2	153	153	100	89	84	38	91	63	102	85
J3	118	116	151	98	156	153	154	135	146	121

Table 12. Open-palm joint angles calculated from the MPH model—front.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	159	165	163	166	168	172	171	179	166	170
J2	175	175	168	175	169	178	171	173	171	171
J3	166	166	177	178	178	179	174	177	172	174

Table 13. Open palm joint angles calculated from the MPH model—45°.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	149	149	151	167	164	171	171	172	168	169
J2	171	176	165	173	163	171	163	164	157	157
J3	169	170	175	175	175	175	175	175	172	172

Table 14. Open palm joint angles calculated from the MPH model—side.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	163	159	140	164	147	168	163	171	161	168
J2	169	174	165	169	157	169	161	171	169	172
J3	174	168	167	177	176	178	172	174	170	172

3.3. Gesture-Classifer Selection

3.3.1. Scope Definition

As discussed in Section 2.6, there were two scopes of prospective classifiers available for investigation. Given that the results of Stage 4 demonstrated the MPH's module instability, a second scope of algorithms were selected. In total, four pre-trained classifiers, which utilised different classification techniques and had different computational loads, were selected for further investigation under this section. The four algorithms are listed below:

1. ANN classifier: Conventional machine-learning classifier built to classify Indian sign language—sourced from [32].
2. Linear-regression classifier: Conventional machine-learning classifier built to classify Russian sign language—sourced from [33].
3. SVM classifier: Conventional machine-learning classifier built to classify Indian sign language—sourced from [34].
4. Bounds-based classifier: Non-machine-learning, statically defined classifier built to classify ASLAN counting gestures—altered version, original found from [31].

Table 15. Open palm joint angles calculated from the MPH model—back.

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	159	164	161	161	172	177	162	164	152	154
J2	176	177	174	176	171	177	175	179	168	175
J3	168	169	174	177	177	178	175	176	175	176

Table 16. Model accuracy calculated from 3D data—by finger.

Hand Position	Percentage Accuracy									
	Thumb		Index		Middle		Ring		Pinkie	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Open	96.2%	92.0%	93.8%	78.7%	95.7%	84.0%	95.7%	91.0%	94.4%	85.4%
Partial	92.2%	63.6%	82.0%	66.1%	82.6%	54.2%	86.2%	68.2%	82.6%	63.6%
Closed	89.6%	60.7%	72.3%	50.9%	77.0%	53.1%	78.5%	41.5%	82.3%	44.4%

Table 17. Model accuracy calculated from 2D data—by finger.

Hand Position	Percentage Accuracy									
	Thumb		Index		Middle		Ring		Pinkie	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Open	96.2%	92.0%	95.6%	84.3%	96.8%	89.1%	96.2%	92.1%	95.4%	86.5%
Partial	91.2%	64.4%	88.9%	72.0%	77.6%	41.3%	86.8%	67.6%	85.4%	66.5%
Closed	88.5%	60.7%	57%	9.8%	59.7%	16.9%	66.4%	22.0%	76.6%	38.9%

Table 18. Model accuracy calculated from 3D data—by viewpoint.

Hand Position	Percentage Accuracy							
	Front		Forty-Five		Side		Back	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Open	96.8%	91.6%	94.5%	84.8%	93.2%	78.7%	96.0%	85.4%
Partial	88.6%	71.0%	90.0%	81.3%	76.1%	63.6%	85.7%	54.2%
Closed	67.8%	41.5%	86.3%	64.3%	84.8%	60.7%	80.7%	51.8%

As per Section 2.6, a common data set was developed such that the four classifiers listed above could be directly compared against one another. After analysing the three gesture sets recognisable by the four classifiers, it was found that they share eight common gestures. A single participant was then imaged to generate the required ten sub-images,

which were subsequently modelled using MPII to form the final data set, as shown in Appendices A and B.

Table 19. Model accuracy calculated from 2D data—by viewpoint.

Hand Position	Percentage Accuracy							
	Front		Forty-Five		Side		Back	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Open	97.6%	92.2%	96.3%	91.2%	94.1%	84.3%	96.2%	86.5%
Partial	91.3%	65.2%	90.3%	80.5%	76.9%	64.4%	85.6%	41.3%
Closed	53.2%	9.8%	81.9%	39.8%	84.9%	60.7%	58.0%	24.1%

3.3.2. Accuracy Results

Figures 7–10 display the confusion matrices used to quantitatively compare the classification accuracies of the four classifiers. The raw data from which these classifiers are generated can be found through Appendix C. From the figures shown below, the non-machine-learning bounds-based classifier performed the best on the given dataset, with an accuracy of 96.25%. The second-best performing classifier was the SVM approach with an accuracy of 81.3%, followed by ANN with an accuracy of 77.5%, and the linear regression model with an accuracy of 70%.

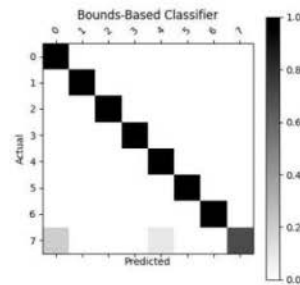


Figure 7. Confusion Matrix 1—bounds-based classifier accuracy.

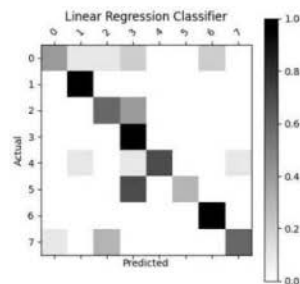


Figure 8. Confusion Matrix 2—linear-regression classifier accuracy.

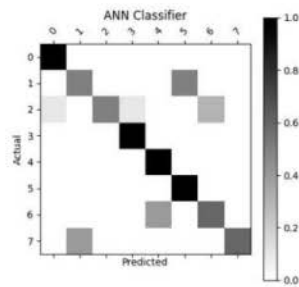


Figure 9. Confusion Matrix 3—ANN classifier accuracy.

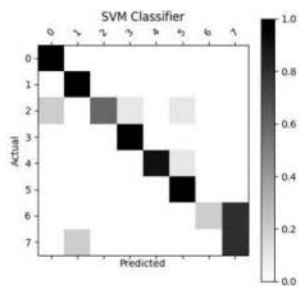


Figure 10. Confusion Matrix 4—SVM classifier accuracy.

3.3.3. Computational Performance Results

The purpose this section of the results was to quantitatively compare the computational performance of the four potential classifiers. As stated in Section 2.6, this was achieved via a comparison of the time taken by each of the classifier to classify each of the 80 images. Figure 11 displays the computational performance data generated during this testing. As displayed, the bounds-based classifier had the lowest classification time, averaging 81.4 ms. The SVM and linear-regression classifiers had comparable performances, both averaging 88 ms. The slowest of the four algorithms was the ANN model with an average classification time of 196 ms.

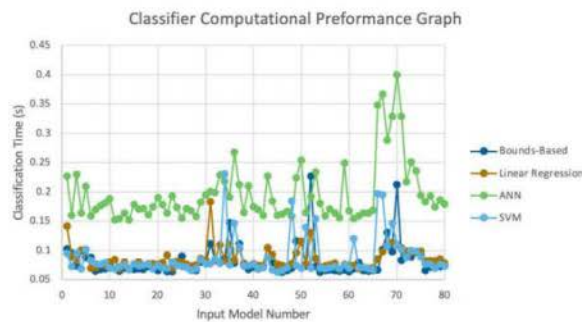


Figure 11. Comparison of tested classifiers computational performance.

3.3.4. Final Classifier Selection

As per the selected scope for this investigation, the selection of the classifier aimed to favour classification accuracy over computational speed. This preference was made to ensure the final algorithm conformed with Governing Criteria 1 and 2. As such, the bounds-based classifier was selected because it had the highest classification accuracy. Fortunately, the bounds-based classifier also had the lowest classification time, indicating that it had the lowest computational requirement, suggesting that it also best conformed to Governing Criterion 5.

3.4. Gesture-Mapping Selection and Tuning

With the selection of the gesture classifier complete, the HGR-based HCI component for the alternative-control solution was complete. The final developmental stage of the investigation could begin, with the development of the gesture-mapping component. As defined in Section 2.7, the development of this gesture-mapping component consisted of two main stages, the first being the selection of the command set, and the second stage being the tuning of the mapping component. This investigatory stage began with the implementation of the position command system.

Following the analysis performed in Stage 5, it was known that the HGR algorithm could accurately recognise the eight gestures present in the common data set. As only eight gestures were required to operate the positional control system, one-to-one mapping could be applied to construct the gesture dictionary shown in Table 20.

Table 20. Positional command set.

Gesture Identifier	Command
1	Move along z axis (forward velocity)
2	Move along $-z$ axis (backward velocity)
3	Move along y axis (upward velocity)
4	Move along $-y$ axis (downward velocity)
5	Move along x axis (bank right)
6	Move along $-x$ axis (bank left)
7	Set required movement along, x, y, and z axes to zero (stop)
8	Take-off or land (depending on whether in flight, or landed)

Through the implementation of a third-party, open source, python API [35], the above commands were able to be transmitted to the TELLO drone. The positional based control framework was then tested and tuned, using the flight paths defined in Figure 12.

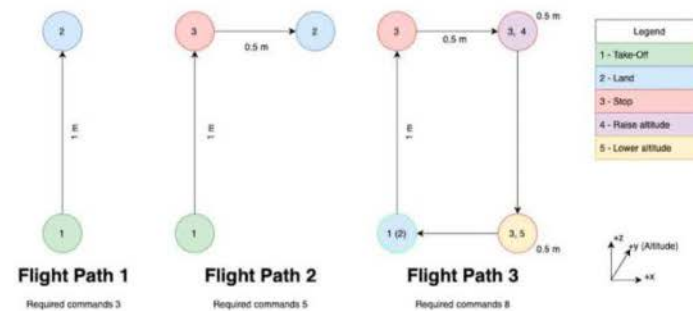


Figure 12. Test flight paths.

The most immediate concern from this initial implementation was the overloading of the TELLO's onboard command buffer. This was caused because the commands were being transmitted at too great a rate for the TELLO drone to process. To solve this, the time taken to recognise a gesture was increased to 500 ms. Whilst this did make the drone appear slightly less responsive initially, it successfully prevented further overloading of the buffer.

However, after these changes, the TELLO appeared to have periods of unresponsiveness during the testing in the final flight path. These periods of unresponsiveness were accompanied by the TELLO drone continually reporting 'error No valid IMU'. This error was attributed to the use of the positional command system, and the method the TELLO drone used to process incoming commands. If a command was received by the drone whilst the drone was processing/reading its IMU data, the drone would abandon the task and attempt to execute the command. As positional commands required IMU data to execute, whenever this interrupt situation would occur, the command would fail, and the drone would re-execute its previous command.

To combat this, the gesture dictionary was redefined to use the velocity command system, as shown in Table 21. This command system did not require readings from the IMU sensor; the TELLO drone executed the received commands immediately and consistently. This resulted in the drone appearing more responsive and subsequently easier to control. The same flight paths were used to tune the velocity control system.

Table 21. Velocity command set.

Gesture Identifier	Command
1	Increase velocity along z axis (forward velocity)
2	Increase velocity along $-z$ axis (backward velocity)
3	Increase velocity along y axis (upward velocity)
4	Increase velocity along $-y$ axis (downward velocity)
5	Increase velocity along x axis (bank right)
6	Increase velocity along $-x$ axis (bank left)
7	Set velocity along, x, y, and z axes to zero (stop)
8	Take-off or land (depending on whether in flight or landed)

The tuning for the velocity control system was the same as the tuning process used for the positional control system. The gestures' hold time remained the same, whilst the command magnitude and command refresh times were both reduced. After this tuning process, the final alternative-control algorithm was able to easily guide the drone through the defined flight paths.

3.5. Performance Overview

With the completion of Stage 6, the final alternative-control algorithm had been fully developed and proved to be functional. To ascertain the final performance of the solution, it was re-analysed against the governing criteria.

- Reliability in issuing the intended command: From the results of Stage 5, the final solution proved to be capable of accurately distinguishing between the intended commands within the command set. When combined with the mapping medium developed in Stage 6, the solution was able to reliably transmit the intended commands to the chosen application medium. Hence, this criterion is satisfied.
- Reproducibility of the intended command: From the results of Stage 4, the initial confidence for the final solution's conformity to this criterion was challenged due to MPH lack of rotational robustness. However, through the implementation of a bounds-

based classifier, the final solution was able to recognise commands robustly and repeatably despite viewpoint and scale variations. Hence, this criterion is satisfied.

- Physically non-restrictive equipment or instrumentation: Given the selection made in Stage 2, the use of a single RGB camera ensured the final solution's conformity to this criterion. Furthermore, as per the results of Stage 5, the final HGR algorithm is capable of recognising gestures from multiple viewpoints meaning the operator does not have to maintain a perfect position in front of the camera. Hence, this criterion is satisfied.
- Ease of operation and shorter user learning cycle: Given the selection made in Stage 1, the use of single-hand, static, sign language gestures ensured that the commands were simple and easy to learn. When combined with the finely tuned, one-to-one gesture-mapping component developed in Stage 6, natural and accessible drone control was facilitated. Hence, this criterion is satisfied.
- Computationally inexpensive: Through the computational analysis performed in Stage 5, and the selection of MPH justified in Stage 3, the final solution was specifically selected to be as computationally lightweight as possible. Hence, this criterion is satisfied.
- Monetarily inexpensive: Given Stage 2's selection of an inexpensive, non-specialised data-acquisition method, and the low computational requirements of the final algorithm, the final solution is monetarily efficient. Hence, this criterion is satisfied.

4. Discussion

4.1. Principal Findings

As previously mentioned, the MPH model accuracy was validated against goniometer readings. The goniometer readings used as a baseline are illustrated in Tables 1–3. The joint angles calculated from the MPH model are illustrated in Tables 4–15. The comparison of these values produced the model accuracy percentages values summarized in Tables 15–18. From these accuracy percentages, the Z-axis instability of the MPH model was characterized (explained in greater depth in Section 4.2.2). This characterization was then used to inform the selection of an initial list of prospective classifiers that were theorized to be able to handle this instability. The classification accuracy of these prospective algorithms was then tested using a common input data set, the results of this test are displayed in Figures 7–11. The final accuracy performance proved that despite MPH's drawbacks, it was still applicable in an alternative-control setting as the now-complete HCI functioned at a high confidence level. Section 3.4 then demonstrated how the control-mapping component was created and tuned to apply the fully developed HCI to drone control. The final performance of the algorithm was then judged against its original governing criteria and demonstrated in the attached multi-media video in the Supplementary Materials section.

4.2. Results Analysis

4.2.1. Gesture-Identifier Selection Analysis

These results demonstrate the potential MPH has to aid in the satisfaction of Criteria 1, 2, and 5. As stated in the methodology defined in Section 2.4, the purpose of Stage 3 was to select a gesture-identification algorithm. Stage 3 achieved its purpose through its selection of MPH based on the reasoning detailed in Section 3.1.3. As this was only a threshold-qualitative investigation was simply intended to determine the applicability of the selected solution, it had numerous limitations. The three significant limitations are the lack of a quantitative analysis, the lacklustre computational power of the host machine, and the limited investigation scope.

The first limitation of this stage was the use of a qualitative analysis method. Whilst this was a valid baseline approach to observe and compare the algorithm's general localisation accuracy, it fails to provide a solid metric from which the true localisation accuracy of MPH can be extrapolated. The primary constraint this imprints onto the investigation is that there are no quantifiable data proving that MPH is the most accurate solution out

of the prospective algorithms. The accuracy of MPH modelling method will be analysed in Stage 4, which does partially account for this. However, the true accuracy of MPH localisation of its landmarks to the joints of the human hand remains largely unknown.

The second limitation arises from the lack of computational power provided by the host machine. This limitation directly excluded InterHands2.6M, which has been proven to be a feature-extraction method comparable to MPH [19]. Whilst MPH did successfully conform to the governing criteria relevant to this section, InterHands2.6M may have had a superior accuracy or computational performance once trained. Unfortunately, this limitation was unavoidable as the host machine chosen for this paper was the only machine available.

The third limitation is derived from the limited number of independent variables that the tested gesture-identification algorithms were tested against. In an ideal setting, if Stage 3 was to be expanded, the gesture-identification algorithms should also have been tested against other sources of variance that are likely to be included in alternative-control applications. These include but are not limited to background variation, operator hand colour variation, operator hand size variation, and lighting variations. Whilst the failure to include these does not invalidate the selection of MPH, it should be considered for future work.

4.2.2. Gesture-Identifier Validation Analysis

As stated in the methodology section defined in Section 2.5, the primary purpose of this stage was to validate the performance of the gesture-identification algorithm. The results shown in Stage 4 achieved this purpose through the successful application of a clinical method to ascertain the accuracy of the MPH model. The results of this investigation effectively displayed the advantages and disadvantages of MPH, the disadvantages of which must be mitigated by future components to ensure the final solution's conformity to the governing criteria.

From the data recorded during Stage 4, four trends about MPH model accuracy became immediately apparent: confirmation of the high potential accuracy of MPH modelling systems, the model's susceptibility to rotation and pose due to its normalised depth coordinate system, the innate benefit of the three-dimensional modelling system over a conventional two-dimensional system, and the model's resistance to self-occlusion. Each of these observations had considerable implications for the future development of both this alternative-control algorithm and for future applications seeking to implement MPH (future implementations will be discussed in greater detail in Section 5).

The first key finding from the analysis performed during Stage 4 was the positive observation of MPH potential accuracy. This key finding supported the observations made in Stage 3, which are demonstrated in Tables 16 and 17. From these tables, MPH was shown to be capable of accurately observing the digits of the human hand to a high degree, having a maximum accuracy percentage of 96.2% while maintaining an average observational accuracy of 86.7%. This alludes to MPH's capability to satisfy Criterion 1, if coupled with an appropriate classification algorithm that can handle the major disadvantages of MPH which will be discussed next.

The second key finding was the observation that MPH robustness was severely limited by its "2.5D coordinate system". MPH uses a normalised depth component, meaning that while the x and y components of a landmark's coordinate are derived from the image's width and height, the z component is derived from a depth calculation between the landmark and the wrist of the observed hand. This results in a model that is not a true three-dimensional representation of the observed hand because the z coordinates have a different scale to the x and y coordinates. MPH attempts to adjust for this by normalising the z component to be within a similar numerical range to the x and y components. However, as demonstrated in Tables 18 and 19, the percentage accuracy of the model can be seen to vary greatly based upon the importance of the z coordinate in the angle calculations.

Orientations that placed connected coordinates along the x and y planes with limited changes in the z coordinate had notably higher accuracy values than orientations that had large changes in the z coordinate. In other words, if the two vectors from which an angle is calculated had large differences in their respective z values, i.e., the angle being observed lay upon the z, y plane, the resultant angle would have a low accuracy. A good example of this observation is a comparison between the percentage accuracy of the closed pose when observed from the side and from the front. When observed from the front, the average accuracy of the MPH model was only 67.8%, in comparison to the model's 84.8% accuracy when observed from the side. From this demonstration, it can be inferred that the normalisation method implemented by MPH fails to successfully equalise the relative scales of the x, y, and z planes, causing the aforementioned instability to rotation. This observation raises a concerning disadvantage for MPH. If this disadvantage is not mitigated by future components, it will have to be mitigated by the operator. Specifically, to ensure that Governing Criterion 1 is met, the operator will have to keep their hands in a constant orientation with respect to the camera. This both limits the manoeuvrability of the operator and makes the algorithm more difficult to use, jeopardising the final algorithm's conformity to Governing Criteria 3 and 4.

The third key finding was that despite the drawbacks of the 2.5D modelling system, the depth component did have some notable benefits over a traditional 2D model. This observation came from the direct comparison between the 2D angle calculations which ignored the depth component and the 3D calculations which used the depth component. While both systems struggled to handle rotation, the three-dimensional system was notably more stable than the two-dimensional system. This can be directly observed by the overall percentages stated in Section 3.2.3, whereby the three-dimensional system's minimum observed accuracy only dropped to 41.5% whilst the two-dimensional system fell to 9.8%. This key finding was practically relevant when informing the selection of future components. Future components were selected to avoid reductions in MPH model complexity, such as the use of joint angles directly, or the removal of the z coordinate, instead favouring the use of the full 21-landmarks, x, y, z coordinate system.

The final key finding was the quantitative validation of MPH resistance to self-occlusion. This key finding validated MPH's ability to handle one of the key limitations of single camera RGB data acquisition, reinforcing its selection as the final gesture-identification algorithm. Evidence for this key finding can be observed in Tables 18 and 19, whereby the average and minimum joint accuracies calculated from the back and side viewpoints remain comparable, if not favourable to angles calculated from the front on viewpoints. As can be seen in Appendices A and B, many of the images taken from the side and back viewpoint had fingers that were not able to be directly observed because they were obscured by other parts of the hand. The fact that the angles calculated from these images produced results similar, if not superior, to their non-obscured counter parts proves MPH's robustness against self-occlusion. This key finding again supports MPH's selection, as it demonstrates how it conforms to Governing Criteria 2 and 4.

However, during the Stage 4 investigation, two limitations were encountered. These were the limited fidelity of the goniometer measurements and the limited number of poses used. The goniometer used for this experiment had a measuring fidelity of 5° intervals. These intervals meant that in almost every measurement, the joint angle was being rounded to the nearest 5° mark as the observed angle often fell between these said marks. This rounding could have contributed to the error percentages recorded in Stage 4. However, as non-digital goniometers remain the industry standard for joint measurements [36], this source of error had to be accepted. This leads into the second limitation that impacted the Stage 4 investigation. Due to the specification to only use three governing poses to keep consistency among all participants, these poses could not be adjusted to set the goniometer to its nearest 5° mark. This prevented the immediate resolution of the error source above. If a more flexible posing regime had been implemented, the poses could have been altered on

a participant-by-participant basis to ensure that the goniometer was reading whole values, rather than having to round to the nearest value.

4.2.3. Gesture-Classifier Selection Analysis

As stated in the methodology for Stage 5, the primary purpose of this stage was to select a gesture classifier that best complimented the functionality of the selected gesture-identification algorithm. The results shown in Stage 5 achieved this purpose through the selection of a bounds-based classifier based on the reasoning shown in Section 3.3.4. Aside from achieving its key purpose, this section produced another key finding which supported the selection of MPH as the gesture-identification component. However, this investigation stage also had three key limitations: the impact of gesture set reduction on pre-trained classifiers, the tests' limited data set, and the unexplained variations in the computational performance data.

The most important finding produced by this investigatory stage, aside from the selection of the classification algorithm, was the demonstration that MPH's disadvantages are surmountable. As can be seen from the confusion matrices shown in Figure 7, despite the inherent variations in the MPH model, the bounds-based classifier was able to achieve an extremely high level of accuracy.

This shows that when MPH is combined with the appropriate auxiliary systems, its disadvantages can be minimised, maximising its effectiveness as a gesture-identification algorithm. This was a key finding that served to both validate the selection of MPH within this overall solution and look favourably on the implementation of MPH in future solutions.

The first key limitation that Stage 5 faced was the impact of gesture set reduction on a pre-trained classifier. As the classifiers could not be re-trained to recognise the eight newly defined gestures, a set of external controls had to be implemented to reduce their classification ranges. This reduction came in the form of a simple cascading set of 'if-else' statements that forced the classifiers to return the highest probability gesture out of the eight-gesture subset, even if the classifier would have otherwise returned a different gesture. This reduction had to be made to produce a proper confusion matrix. To prevent this reduction from impacting upon the overall validity of this experiment, the time take to complete this reduction was removed from the overall classification time. Ideally, the classifiers would have been re-trained to only recognise the eight new defined gestures.

The second key limitation was the size and scope of the data set. Given that this was one of the two secondary investigations performed over the course of this paper, it was not deemed practical to develop a test data set with more than a thousand images. This limited the level of testing that could be performed on the four classifiers. While it does not invalidate the classifier selection made above, in an ideal setting, a larger data set would have been used, with more edge-case gestures being included to really test the limits of the classification algorithms.

The final limitation centred around the unexpected behaviour displayed toward the end of Figure 11. Initially, it was assumed that each of the algorithm's classification times would remain largely consistent across the 80 input images. This was an assumption based on the idea that the classification time would only fluctuate greatly if the size and/or complexity of the input data changed. As the input data remained constant, in both size and complexity across the 80 input models, the fluctuation remained unexplained. However, as the selected classifier was consistently faster than the other classifiers even in the regions of fluctuation, the validity of the final solution was not compromised by this limitation.

5. Conclusions

This project achieved its primary purpose by developing a functioning alternative-control algorithm that extended the usability of a quad-rotor drone. This was achieved through the development of an HGR algorithm that combined the functionality of MPH and a bounds-based classifier. The final solution facilitated natural and accessible control while being computationally inexpensive and not requiring the use of specialized camera

equipment. The success of the final solution demonstrated the applicability of modern, single-camera HGR algorithms within the confines of alternative control. Furthermore, the clinical evaluation of MPH demonstrated MPH's inherent advantages and disadvantages. The success of the developed alternative-control algorithm shows that when handled appropriately, MPH can be a powerful HGR tool that has applications within clinical and control settings. However, future projects seeking to apply MPH must be mindful of the algorithm's limitations or risk failure. There are three main areas of future work related to this paper, these being an extension of the applied methodology, the application of MPH's model validation data, and the application of the developmental framework used in this paper alongside the final solution itself.

The first proposed area of future work is an extension of the method applied in this paper. One area of this extension is the completion of a broader comparative review of gesture-identification components, specifically aiming to extend upon the works of this paper by including a quantitative comparison of landmark-localization accuracy in its analysis of algorithms such as MPH and InterHands2.6M. Furthermore, another proposed extension of this paper's methodology is a data-based analysis of the final solution's performance. This could be achieved by grading the final solution and the TELLO drone's standard control mechanism against the governing criteria of this paper using quantitative metric-based testing.

The second proposed area of future work is the application of the MPH model's validation data and joint-localization accuracy observations. The first area of application is in clinical diagnoses, focusing on using MPH to generate joint angles accurate enough to diagnose illness and injury. As per the findings of this paper, MPH cannot do this directly due to its modelling method's instability; however, it could be achieved using sensor fusion. One possible avenue would be using MPH's joint-localisation accuracy to efficiently locate points of interest and then modelling these joints in a true three-dimensional environment using, for example, a stereo camera. The second setting is in rehabilitation, focusing on using MPH's current level of accuracy to observe the general motion of the human hand and then acting upon this motion in a gamified environment.

The final proposed area of future work is the application of the developed alternative-control algorithm and the framework used to construct it. The most immediate application for the developed alternative-control algorithm is using it to extend the accessibility of drone control to operators that cannot operate the standard control medium due to having impaired dexterity. Another application for the developed algorithm is seeking to optimise its computationally lightweight nature to foster its use in either embedded or edge computing environments. The framework used to develop the final alternative-control algorithm can be re-applied to a multitude of other alternative-control tasks. With the now fully developed HGR component, all that would be required to re-tune the current solution to control a new agent is the reconstruction of the command mapping component.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/s23125462/s1>, Video S1: Video demos for the paper.

Author Contributions: Conceptualization, I.M. and S.K.; data curation, L.C. and S.K.; formal analysis, S.K. and L.C.; investigation, S.K., L.C. and B.B.; methodology, S.K. and I.M.; software, L.C. and S.K.; supervision, I.M. and S.K.; validation, S.K., L.C. and B.B.; visualization, S.K., L.C. and B.B.; original draft, S.K., L.C. and B.B.; review and editing, I.M., L.C., S.K. and B.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This study was conducted in accordance with the Declaration of Helsinki and approved by the Institutional Review Board (or Ethics Committee) of Curtin University Ethics Committee (approval number HRE2021-0047, granted on 3 February 2021) and (approval number HRE2022-0583, granted on 18 October 2022).













Informed Consent Statement: Informed consent was obtained from all subjects involved in this study.

Data Availability Statement: Not applicable.


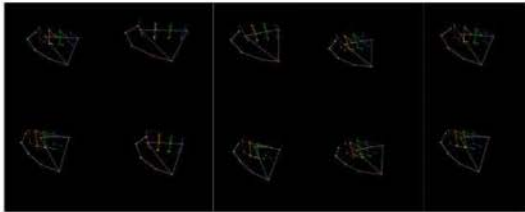

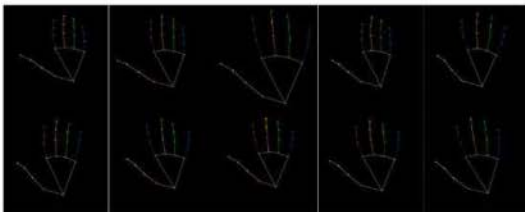
Acknowledgments: The authors would like to acknowledge Jayden Balestra from Climbit Physio who supported the clinical validation of this paper's algorithms.


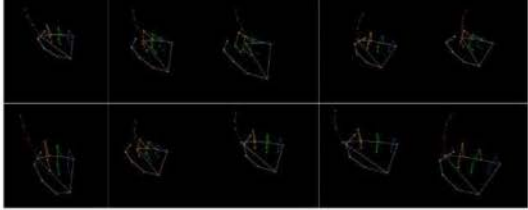

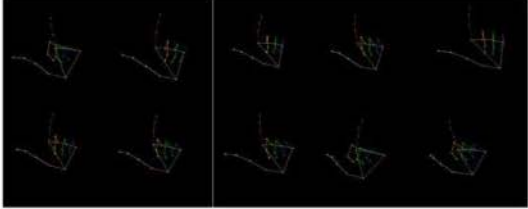

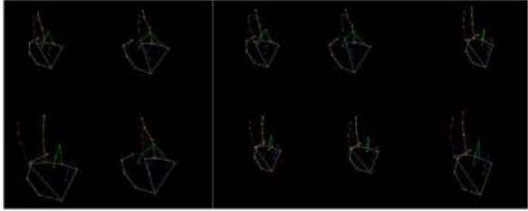

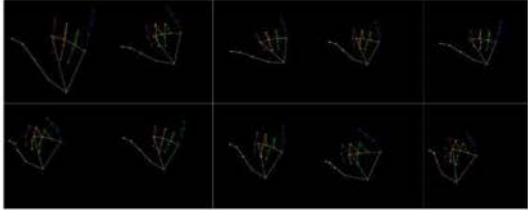

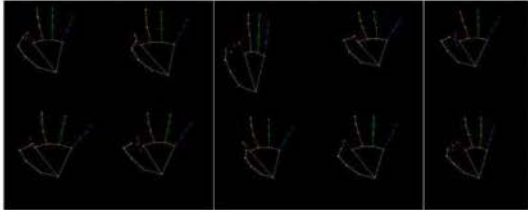
Conflicts of Interest: The authors declare no conflict of interest.


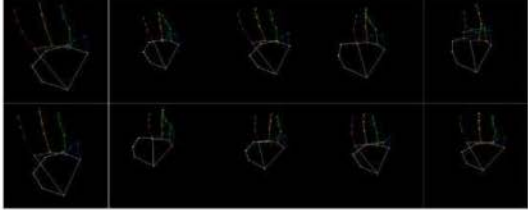
Appendix A. Stage 4—Viewpoint and Pose Example

Pose	Viewpoint Offset			
	0° (Front)	45° (Forty)	90° (Side)	180° (Back)
Open				
Partial				
Closed				

Appendix B. Stage 5—Classification Analysis Data Set

Gesture Identifier	Gesture Reference Image	Models
1		
2		

Gesture Identifier	Gesture Reference Image	Models
3		
4		
5		
6		
7		

Gesture Identifier	Gesture Reference Image	Models
8		

Appendix C. Stage 5—Classification Raw Confusion Data

Linear							
4	1	1	2	0	0	2	0
0	10	0	0	0	0	0	0
0	0	6	4	0	0	0	0
0	0	0	10	0	0	0	0
0	1	0	1	7	0	0	1
0	0	0	7	0	3	0	0
0	0	0	0	0	0	10	0
1	0	3	0	0	0	0	6
ANN							
10	0	0	0	0	0	0	0
0	5	0	0	0	5	0	0
1	0	5	1	0	0	3	0
0	0	0	10	0	0	0	0
0	0	0	0	10	0	0	0
0	0	0	0	0	10	0	0
0	0	0	0	4	0	6	0
0	4	0	0	0	0	0	6
SVM							
10	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0
2	0	6	1	0	1	0	0
0	0	0	10	0	0	0	0
0	0	0	0	9	1	0	0
0	0	0	0	0	10	0	0
0	0	0	0	0	0	2	8
0	2	0	0	0	0	0	8
Bounds-Based							
10	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0
0	0	10	0	0	0	0	0
0	0	0	10	0	0	0	0
0	0	0	0	10	0	0	0
0	0	0	0	0	10	0	0
0	0	0	0	0	0	10	0
2	0	0	0	1	0	0	7

Appendix D. Code Methods

This displays the specific python calculation methods used by various aspects of the report:

```
def calcJointAngles_3D(landmarkArray):
    jointAngles = [0]*15
    jointAngleIndex = 0

    for i in range(5):
        for j in range(3):
            pInit = landmarkArray.landmark[0]
            print(landmarkArray.landmark[0])
            if (j != 0):
                pInit = landmarkArray.landmark[i*4+j]
            pMid = landmarkArray.landmark[i*4 + j + 1]
            pFinal = landmarkArray.landmark[i*4 + j + 2]

            v1 = [pMid.x - pInit.x, pMid.y - pInit.y, pMid.z - pInit.z]
            v2 = [pFinal.x - pMid.x, pFinal.y - pMid.y, pFinal.z - pMid.z]

            theta_num = v1[0]*v2[0] + v1[1]*v2[1] + v1[2]*v2[2]
            theta_den = calcMagnitude(v1) * calcMagnitude(v2)

            theta = math.acos(theta_num / theta_den)

            jointAngle = (math.pi - theta) * 180/math.pi
            jointAngles[jointAngleIndex] = jointAngle
            jointAngleIndex += 1

    return jointAngles
```

Figure A1. Three-dimensional joint-calculation method.

```
def calcJointAngles_2D(landmarkArray):
    jointAngles = [0]*15
    jointAngleIndex = 0

    for i in range(5):
        for j in range(3):
            pInit = landmarkArray.landmark[0]
            print(landmarkArray.landmark[0])
            if (j != 0):
                pInit = landmarkArray.landmark[i*4+j]

            pMid = landmarkArray.landmark[i*4 + j + 1]
            pFinal = landmarkArray.landmark[i*4 + j + 2]

            a = np.array([pInit.x, pInit.y]) # First coord
            b = np.array([pMid.x, pMid.y]) # Second coord
            c = np.array([pFinal.x, pFinal.y]) # Third coord

            radians = np.arctan2(c[1] - b[1], c[0] - b[0]) - np.arctan2(a[1] - b[1], a[0] - b[0])
            angle = np.abs(radians*180.0/np.pi)

            if angle > 180.0:
                angle = 360-angle

            jointAngles[jointAngleIndex] = round(angle)
            jointAngleIndex += 1

    return jointAngles
```

Figure A2. Two-dimensional joint-calculation method.

Appendix E.

This appendix displays the specifications for the mechanism used to demonstrate the functionality of the final alternative-control algorithm. The selected mechanism was a TELLO Drone. Table A1 includes the general technical specifications for the drone. Figure A3 is an image of the TELLO drone.

Table A1. TELLO specifications. Values reproduced from [37].

Aircraft	Flight Weight	80 g
	Dimensions	98 × 92 × 41 mm
	Propeller Diameter	76.2 mm
	Built in Functions	Range Finder Barometer LED Vision System 2.4 GHz 802.11n Wi-Fi
	Electrical Interface	Micro USB Charging Port
Flight Performance	Maximum Flight Range	100 m
	Maximum Flight Time	13 min
	Maximum Speed	8 m/s
	Maximum Height	30 m
Battery	Detachable Battery	3.8 V–1.1 Ah
Camera	Photo	5 MP (2592 × 1936)
	FOV	82.6°
	Video	HD720P30
	Format	JPG (Photo), MP4 (Video)
	EIS	Yes

**Figure A3.** TELLO drone [38].

References

1. Mahmood, M.; Rizwan, M.F.; Sultana, M.; Habib, M.; Imam, M.H. Design of a Low-Cost Hand Gesture Controlled Automated Wheelchair. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSymp), Dhaka, Bangladesh, 5–7 June 2020; pp. 1379–1382. [CrossRef]
2. Posada-Gomez, R.; Sanchez-Medel, L.H.; Hernandez, G.A.; Martinez-Sibaja, A.; Aguilar-Laserre, A.; Lei-ja-Salas, L. A Hands Gesture System of Control for an Intelligent Wheelchair. In Proceedings of the 2007 4th International Conference on Electrical and Electronics Engineering, Bursa, Turkey, 5–7 September 2007; pp. 68–71. [CrossRef]
3. Hu, B.; Wang, J. Deep Learning Based Hand Gesture Recognition and UAV Flight Controls. *Int. J. Autom. Comput.* **2020**, *17*, 17–29. [CrossRef]
4. Lavanya, K.N.; Shree, D.R.; Nischitha, B.R.; Asha, T.; Gururaj, C. Gesture Controlled Robot. In Proceedings of the 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysore, India, 15–16 December 2017; pp. 465–469. [CrossRef]

5. Premaratne, P.; Nguyen, Q.; Premaratne, M. Human Computer Interaction Using Hand Gestures. In *Advanced Intelligent Computing Theories and Applications*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 381–386. [\[CrossRef\]](#)
6. Mitra, S.; Acharya, T. Gesture Recognition: A Survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 311–324. [\[CrossRef\]](#)
7. Guo, L.; Lu, Z.; Yao, L. Human-machine interaction sensing technology based on hand gesture recognition: A review. *IEEE Trans. Hum.-Mach. Syst.* **2021**, *51*, 300–309. [\[CrossRef\]](#)
8. Liu, H.; Wang, L. Gesture recognition for human-robot collaboration: A review. *Int. J. Ind. Ergon.* **2018**, *68*, 355–367. [\[CrossRef\]](#)
9. Damaneh, M.M.; Mohanna, F.; Jafari, P. Static hand gesture recognition in sign language based on convolutional neural network with feature extraction method using ORB descriptor and Gabor filter. *Expert Syst. Appl.* **2023**, *211*, 118559, ISSN: 0957-4174. [\[CrossRef\]](#)
10. Ma, Y.; Liu, Y.; Jin, R.; Yuan, X.; Sekha, R.; Wilson, S.; Vaidyanathan, R. Hand Gesture Recognition with Convolutional Neural Networks for the Multimodal UAV Control. In Proceedings of the 2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Cranfield, UK, 25–27 November 2019; pp. 198–203. [\[CrossRef\]](#)
11. Yoo, M.; Na, Y.; Song, H.; Kim, G.; Yun, J.; Kim, S.; Moon, C.; Jo, K. Motion Estimation and Hand Gesture Recognition-Based Human-UAV Interaction Approach in Real Time. *Sensors* **2022**, *22*, 2513. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Yeh, Y.-P.; Cheng, S.-J.; Shen, C.-H. Research on Intuitive Gesture Recognition Control and Navigation System of UAV. In Proceedings of the 2022 IEEE 5th International Conference on Knowledge Innovation and Invention (ICKII), Hualien, Taiwan, 22–24 July 2022; pp. 5–8. [\[CrossRef\]](#)
13. Tsai, C.-C.; Kuo, C.-C.; Chen, Y.-L. 3D Hand Gesture Recognition for Drone Control in Unity. In Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), Hong Kong, China, 20–21 August 2020; pp. 985–988. [\[CrossRef\]](#)
14. Lee, J.-W.; Yu, K.-H. Wearable Drone Controller: Machine Learning-Based Hand Gesture Recognition and Vibrotactile Feedback. *Sensors* **2023**, *23*, 2666. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Jiang, S.; Kang, P.; Song, X.; Lo, B.P.L.; Shull, P.B. Emerging wearable interfaces and algorithms for hand gesture recognition: A survey. *IEEE Rev. Biomed. Eng.* **2022**, *15*, 85–102. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Rautaray, S.S.; Agrawal, A. Vision Based Hand Gesture Recognition for Human-Computer Interaction: A survey. *Artif. Intell. Rev.* **2015**, *43*, 1–54. [\[CrossRef\]](#)
17. Aggarwal, J.K.; Ryoo, M.S. Human Activity Analysis: A Review. *ACM Comput. Surv.* **2011**, *43*, 16:1–16:43. [\[CrossRef\]](#)
18. Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.L.; Grundmann, M. MediaPipe Hands: On-device real-time hand tracking. *arXiv* **2020**, arXiv:2006.10204. [\[CrossRef\]](#)
19. Moon, G.; Yu, S.-I.; Wen, H.; Shiratori, T.; Lee, K.M. InterHand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. In *Computer Vision—ECCV 2020 (Lecture Notes in Computer Science)*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 548–564. [\[CrossRef\]](#)
20. Ge, L.; Ren, Z.; Li, Y.; Xue, Z.; Wang, Y.; Cai, J.; Yuan, J. 3D hand shape and pose estimation from a single RGB image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10833–10842. [\[CrossRef\]](#)
21. Jindal, M.; Bajal, E.; Sharma, S. A Comparative Analysis of Established Techniques and Their Applications in the Field of Gesture Detection. In *Machine Learning Algorithms and Applications in Engineering*; CRC Press: Boca Raton, FL, USA, 2023; p. 73.
22. Yasen, M.; Jusoh, S. A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Comput. Sci.* **2019**, *5*, e218. [\[CrossRef\]](#) [\[PubMed\]](#)
23. American Sign Language. Wikipedia. Available online: https://en.wikipedia.org/wiki/American_Sign_Language (accessed on 3 April 2023).
24. Oudah, M.; Al-Naji, A.; Chahl, J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *J. Imaging* **2020**, *6*, 73. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Balestra, J.; (Climbit Physio, Belmont, WA, Australia). Personal communication, 2022.
26. Xu, X.; Zhang, X.; Fu, H.; Chen, L.; Zhang, H.; Fu, X. Robust Passive Autofocus System for Mobile Phone Camera Applications. *Comput. Electr. Eng.* **2014**, *40*, 1353–1362. [\[CrossRef\]](#)
27. Bhushan, S.; Alshehri, M.; Keshla, I.; Chakraverti, A.K.; Rajpurohit, J.; Abugabah, A. An Experimental Analysis of Various Machine Learning Algorithms for Hand Gesture Recognition. *Electronics* **2022**, *11*, 968. [\[CrossRef\]](#)
28. Gadekallu, T.R.; Srivastava, G.; Liyanage, M.; Iyapparaja, M.; Chowdhary, C.L.; Koppu, S.; Maddikunta, P.K.R. Hand Gesture Recognition Based on a Harris Hawks Optimized Convolution Neural Network. *Comput. Electr. Eng.* **2022**, *100*, 107836. [\[CrossRef\]](#)
29. Katsuki, Y.; Yamakawa, Y.; Ishikawa, M. High-speed human/robot hand interaction system. In Proceedings of the HRIACM/IEEE International Conference on Human-Robot Interaction System, Portland, OR, USA, 2–5 March 2015; pp. 117–118. [\[CrossRef\]](#)
30. MediaPipe. MediaPipeHands [SourceCode]. Available online: <https://github.com/google/mediapipe/tree/master/mediapipe/python/solutions> (accessed on 21 April 2022).
31. CVZone. HandTrackingModule [SourceCode]. Available online: <https://github.com/cvzone/cvzone/blob/master/cvzone/HandTrackingModule.py> (accessed on 23 May 2022).
32. Soumotanu Mazumdar. Sign-Language-Detection [SourceCode]. Available online: <https://github.com/FortunateSpy5/sign-language-detection> (accessed on 21 August 2022).

33. Dmitry Manoshin. Gesture_Recognition [SourceCode]. Available online: https://github.com/manosh7n/gesture_recognition (accessed on 9 August 2022).
34. Halder, A.; Tayade, A. Real-time vernacular sign language recognition using mediapipe and machine learning. *Int. J. Res. Publ. Rev.* **2021**, *2*, 9–17. Available online: https://scholar.google.com/scholar?as_q=Real-time+vernacular+sign+language+recognition+using+mediapipe+and+machine+learning&as_occt=title&hl=en&as_sdt=0%2C31 (accessed on 16 August 2022).
35. Damia F Escote. DJITelloPy [SourceCode]. Available online: <https://github.com/damiafuentes/DJITelloPy> (accessed on 2 April 2022).
36. Hamilton, G.F.; Lachenbruch, P.A. Reliability of Goniometers in Assessing Finger Joint Angle. *Phys. Ther.* **1969**, *49*, 465–469. [CrossRef] [PubMed]
37. TELLO SPECS. RYZE. Available online: <https://www.rzyzerobotics.com/tello/specs> (accessed on 25 May 2023).
38. Tello. Wikipedia. Available online: [https://de.wikipedia.org/wiki/Tello_\(Drohne\)](https://de.wikipedia.org/wiki/Tello_(Drohne)) (accessed on 25 May 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

4.6 SUMMARY OF USE CASE EXAMPLES AND THEIR IMPLEMENTATION

This chapter provided 4 different use cases which have been published in peer reviewed journals. The CP use cases were selected as example of use of IMUs in clinical settings. IMUs were used to capture movement associated with CP in two experiments. One for measuring the effectiveness of wrist orthotics for two groups of children and young adults in CP via randomized controlled trials. Another example utilised IMUs to measure HCA for children with CP. Both instances demonstrated how IMU sensors can be utilised in rehabilitation environments.

The posture monitoring paper provided an example of measuring full posture utilising the developed framework. Different posture correction exercises were implemented where the user was able to get feedback on their movement while wearing the sensors within their PPE. This paper also provided brief look at how the joint angle measurements were calculated and validated against Curtin University's Motion Analysis Lab (MAL) and goniometer. Chapter 5 will provide a detailed look system testing and validation process.

The alternative control paper provided the user case and implementation of HGR and detailed the methodology for implementing this technology within the framework. The drone control mentioned in this chapter was utilised and a physical game activity where a free form finger movement exercise can fly the drone in different directions. The control was also implemented in the Aeroplane flying game that was discussed Chapter 3.

After successful validation of different use cases for the different elements of the framework, a focus group was conducted with experts in the field of rehabilitation. This focus group discussion was analysed utilising narrative analysis method of qualitative research. The details of this qualitative research have been provided in Chapter 5.

Chapter 5: Analysis of Results and System Testing

5.1 INTRODUCTION OF SYSTEM TESTING AND VERIFICATION

This chapter provides details of how different aspects of the framework were tested, evaluated, and verified. Both quantitative and qualitative methods of research have been applied to evaluate the engineering and clinical perspectives respectively.

For the engineering perspectives, a mix of clinical trials were conducted so the information provided by the framework can be evaluated against gold standards of human movement capture such as Vicon Motion Analysis Laboratories at Curtin University as well as against clinical tools like goniometers. In addition to the clinical trials and tests, four separate use cases as discussed in Chapter 4 were utilised where different elements of the framework were tested and evaluated.

This second half of this chapter provides a detailed qualitative analysis in from a focus group discussion that was conducted with experts in field of rehabilitation. The feedback provided in this focus group was analysed utilising narrative analysis methodology. The participants for this focus group provided signed consent for their names to be published in this thesis and for quotes to be attributed to them if necessary.

5.2 ANALYSIS AND SYSTEM TESTING – ENGINEERING SPECIFICATIONS

5.2.1 Evaluation of Sensors Against Vicon Video Capture System

As mentioned in Chapter 2, the most accurate form of human movement analysis is believed to be analysis via video capture technology. There are number of studies that have demonstrated the accuracy of IMU sensors for human movement as demonstrated in the different use cases in Chapter 4. In additional to the evidence provide in the literature, Curtin University’s Vicon motion analysis lab (MAL) was utilised as a comparison point.

Reflectors were connected to the IMU sensors while at Curtin University’s MAL [16]. The IMU sensors were then connected to a user’s hand and slightly above the

wrist as seen in Figure 92. The user then took part in three rehabilitation exercises which were:

- Flexion and extension exercises
- Small object pickup exercise
- Pressing a stop sign exercise.

The IMU data and Vicon data was collected at the same time and the same sampling rate. After completion of the activity, the data for both activities was drawn as seen in Figure 93 and Figure 94. Please note that the data has been slightly shifted so the similarities of the peaks and valleys of the signal can be visible.



Figure 92: Placing reflectors on a user for Vicon Validation [16]

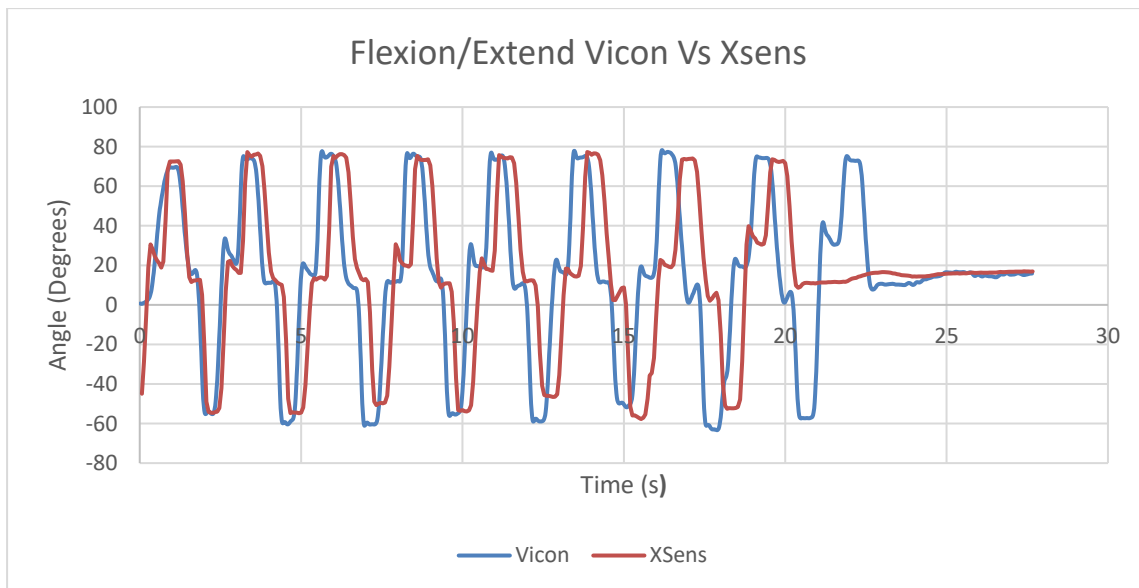


Figure 93: IMU vs Vicon for flexion and extension exercise



Figure 94: IMU vs Vicon for object pick up and stop sign exercises.

5.2.2 Evaluation of Sensor Joint Measurements Against Goniometers

As discussed in Chapter 2, goniometers are reportedly the most used method of calculating joints in clinical settings. This means the next step in evaluating the components of the framework was to evaluate the accuracy of the IMU calculation against a goniometer. To achieve this, IMUs were connected to a goniometer and multiple angle measurements were taken. This test was done once with just the IMU

connected to the legs of a goniometer with legs moving without it connected to a user and another time with the IMU connected to different parts of the body while interacting with the framework. The IMU connection to the goniometer can be seen in Figure 95. For the first test, IMU data was compared to joint measurements via Xsens Dot mobile application and Xsens Dot KineXYZ application that allowed for direct stream of IMU data for evaluation. Additionally, MetaMotionR (MMR) IMU sensor was used to demonstrate the fact that joint angle measurements are independent on the IMU brand as long as the based specifications are met in accordance with hardware requirements mentioned in Chapter 3. The comparison of these calculations can be seen in Figure 96.

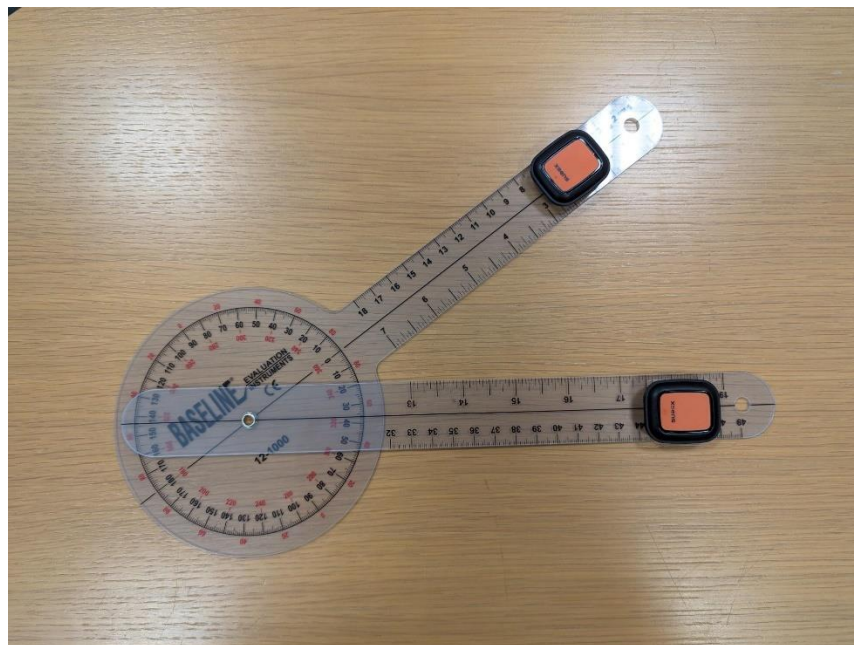


Figure 95: IMUs connected directly to a goniometer

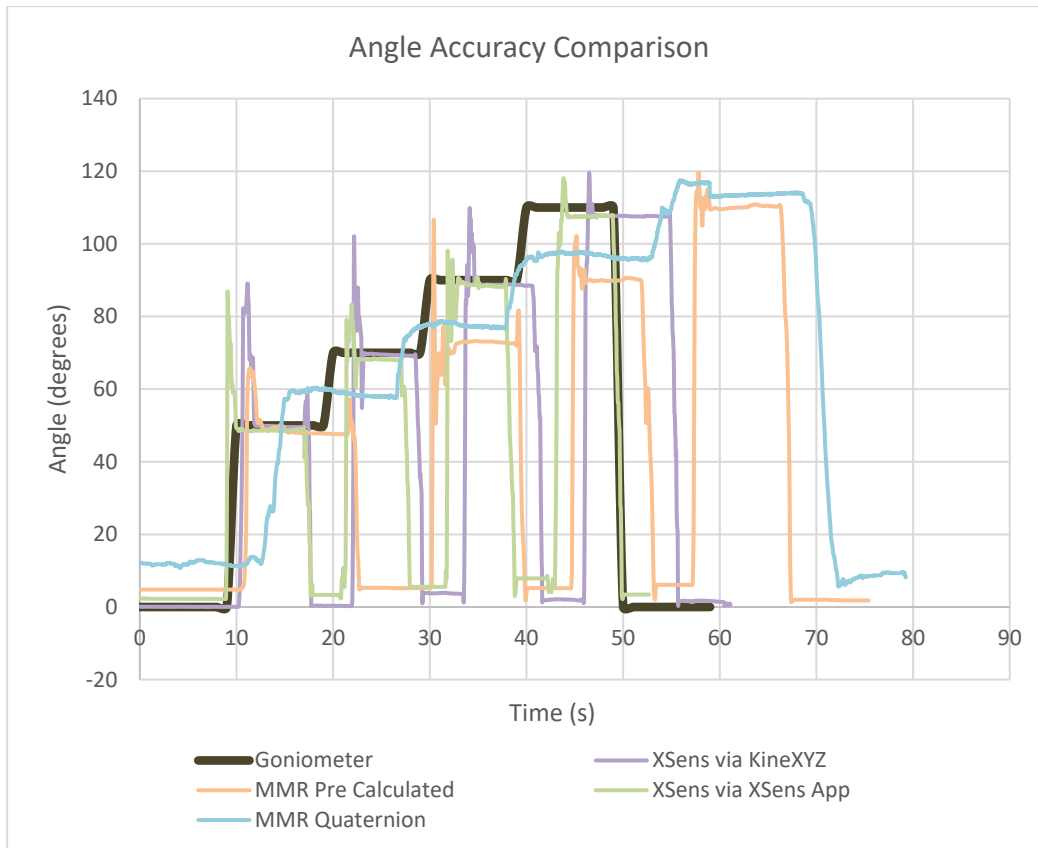


Figure 96: Comparison of Different IMUs with a goniometer

The process of testing IMU data connected directly to a user was explained in detail as part of the posture monitoring use cases found in Chapter 4 and published in [16]. Figure 97, Figure 98, and Figure 99 demonstrates an example of these tests for wrist flexion/extension. The data in Table 6 and Table 7 demonstrate a comparison of IMU data and the user joint measurements via the framework. In these tests the validation was done against a goniometer at different angles using the three joint measurement techniques mentioned in Chapter 3. More example of these tests can be found in Chapter 4 and published in [16].

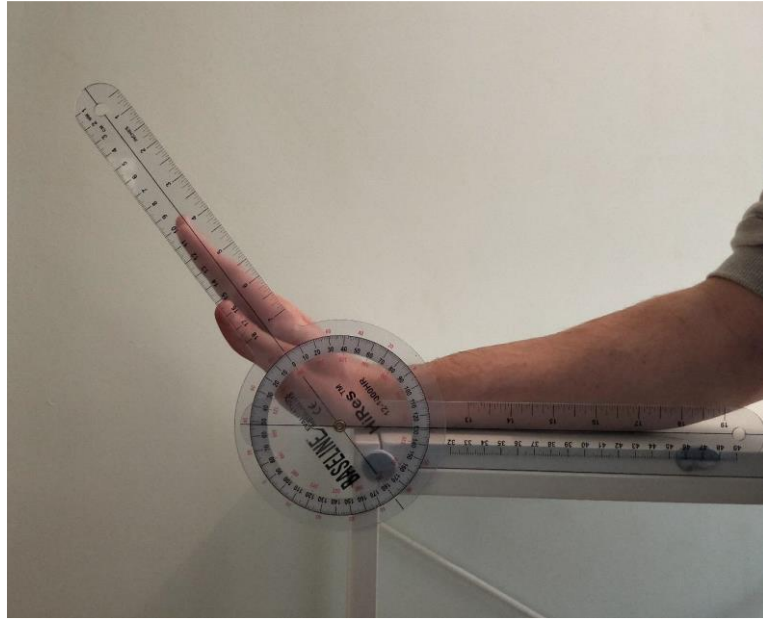


Figure 97: Use of goniometer for calculating wrist joints (Baseline measurement).

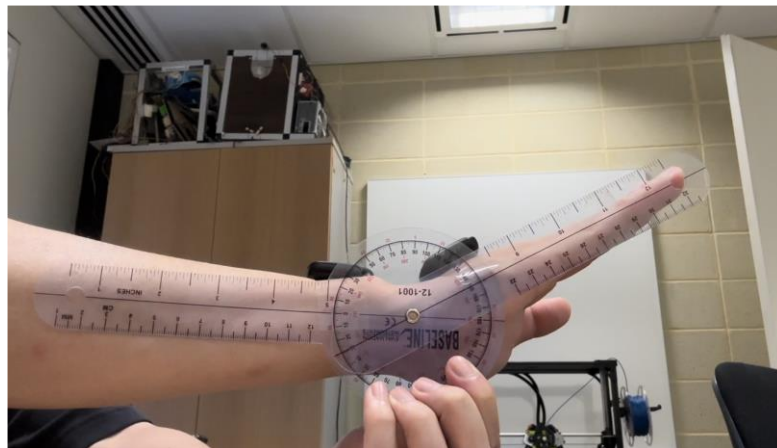


Figure 98: Use of goniometer for calculating wrist joints (Extension measurement)

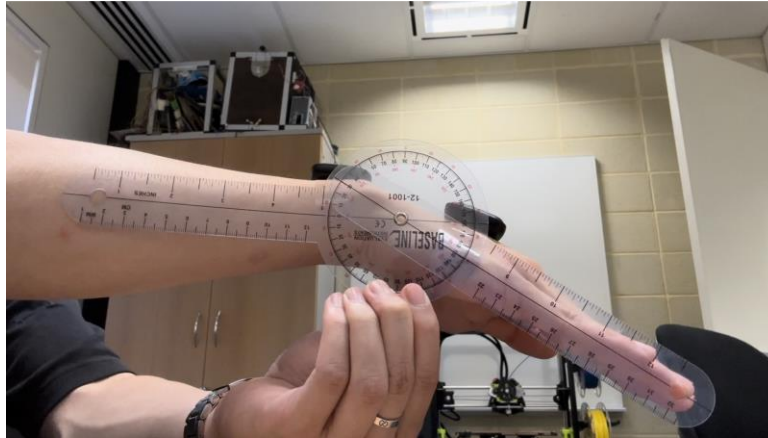


Figure 99: Use of goniometer for calculating wrist joints (Flexion measurement)

Table 6 : Joint angle measurement against goniometer for wrist flexion

Goniometer	Method 1			Method 2			Method 3		
	X	Y	Z	X	Y	Z	X	Y	Z
0	0	1	1	-1	0	0	-7	-83	-35
10	3	11	11	10	3	0	3	-87	-35
20	2	22	22	21	2	0	11	-93	-36
30	1	29	29	29	1	0	19	-100	-37
40	1	39	39	39	1	1	26	-106	-39
50	2	50	49	48	3	1	34	-113	-43

Table 7: Joint angle measurement against goniometer for wrist extension

Goniometer	Method 1			Method 2			Method 3		
	X	Y	Z	X	Y	Z	X	Y	Z
0	16	NAN	17	-1	16	1	2	-69	-35
10	19	10	22	-11	19	1	-4	-61	-35
20	21	18	29	-20	21	1	-10	-54	-35
30	23	31	37	-29	22	1	-18	-47	-36
40	24	39	45	-41	23	1	-25	-40	-38
50	25	49	54	-50	24	2	-32	-32	-41

Once it was determined that method 2 of joint angle calculation was the most accurate, another set of tests were run to measure the absolute error of this method against a goniometer. These error calculations can be seen in Table 8, Table 9, Table 10, and Table 11.

Table 8: Error calculation for flexion angles

Goniometer Angle	Xsens DOT Measured Angle	%Error	Absolute Error
5	4.8	4	-0.2
15	16.2	8	1.2
25	24.8	0.8	-0.2
35	36.7	4.86	1.7
45	47.5	5.56	2.5
55	56.4	2.5	1.4

Table 9: Error calculation for extension angles

Goniometer Angle	Xsens DOT Measured Angle	%Error	Absolute Error
5	4.8	4	-0.2
15	15.4	2.67	0.4
25	25.8	3.2	0.8
35	34.5	1.4	-0.5
45	45.6	1.3	0.6
55	55.2	0.36	0.2

Table 10: Error calculation for radial deviation angles

Goniometer Angle	Xsens DOT Measured Angle	%Error	Absolute Error
5	4.5	10	-0.5
10	9.75	2.5	-0.25
15	15.9	6	0.9
20	21.5	7.5	1.5
25	25.5	2	0.5
30	28.7	4.3	-1.3

Table 11: Error calculation for ulnar deviation angles

Goniometer Angle	Xsens DOT Measured Angle	%Error	Absolute Error
5	5.2	4	0.2
10	10.3	3	0.3
15	15.5	3.3	0.5
20	20.6	3	0.6
25	24.2	3.2	-0.8
30	30.5	1.7	0.5

5.2.3 Evaluation of Sensor Smoothness Measurements

The reliability of the smoothness measurements via LDLJ and SPARC needed to be evaluated. To achieve this, a healthy participant engaged in the Aeroplane flying game going through 5 levels of difficulty. The same user then engaged with the same game with inclusion of Jerky movements in their interaction. The result of these calculations can be seen in Table 12. As it can be seen, the results of SPARK for level 1 vary greatly to other levels due to the rest times available in this level. This means that the results need to be analysed by removing level 1 data as outlier. This led to the observation that normal movements were $\sim 0.75 - 1.25$ lower than jerky movements. If LDLJ is considered, normal movement is $\sim 0.40 - 1.55$ lower than those of jerkier movements.

Table 12: Smoothness measurement comparing normal vs jerky movements.

Healthy User	SPARC (Linear)	SPARC (Angular)	LDLJ (Linear)	LDLJ (Angular)
Level 1				
Normal	-15.0648	-16.5873	-16.4036	-17.1783
Jerky	-7.97966	-8.23135	-14.0585	-14.1818
Difference	7.08514	8.35595	2.3451	2.9965
Level 2				
Normal	-8.8484	-8.90046	-16.3254	-16.0477
Jerky	-10.0893	-11.8307	-17.4618	-17.4405
Difference	-1.2409	-2.903024	-1.1364	-1.3928
Level 3				
Normal	-10.3873	-9.21525	-17.4934	-17.0546
Jerky	-11.1552	-11.8074	-18.6294	-18.6323
Difference	-0.7679	-2.59215	-1.136	-1.5777
Level 4				
Normal	-10.5619	-10.2303	-17.548	-17.1992
Jerky	-11.3361	-11.0971	-15.9417	-16.023
Difference	-0.7742	-0.8668	1.6063	1.1762
Level 5				
Normal	-10.5947	-9.95636	-17.8519	-17.8753
Jerky	-11.5976	-13.3031	-19.1289	-18.2584
Difference	-1.0029	-3.34674	-1.277	-0.3831

After the initial test, a study was run with a total of 5 participants where 2 were healthy and 3 had varying levels of spinal cord injury. The users with spinal cord injury

participated in the same activity and the results were compared to the healthy participant. The three participants in this study (presented as User A, User B and User C) had varying levels of spinal cord injury. Table 13 provide a summary of the user's capability as part of this study. The healthy participants were both 20 years of age at the time of study. The data captured for the study can be seen in Table 14, Table 15, and Table 16 . The results demonstrate that SPARC and LDLJ can both be utilised to monitor smoothness measurements, where the variance in smoothness of the movements can clearly be seen in the provided data below. Additionally, the improvement overtime can be seen for user B as there is a 1-month gap in the data presented for this user which shows the effects of the gamified rehabilitation exercises. For user A, SPARC shows a difference range of ~0.20 – 1.20 and for user B the SPARC values were 2.15 – 10.60. The improvements for user B can be seen in their improved SPARC values of 1.45 – 3.20. This data proves that quantitative measures of SPARC and LDLJ have demonstrated smoothness measurements when compared to qualitative analysis of the user's performances.

Table 13: Information regarding participants of the trial

User Letter Code	Injury Type	Left hand or right hand	Age in 2022	Time of Injury
A	C5 incomplete ASIA C, 50% normal function in upper limb	Right-handed but prefers left after injury	70	January 2021
B	C4 incomplete ASIA C, 30% normal function in upper limb	Right-handed	30	When they were a teenager
C	C4 complete ASIA A, 14% normal function in upper limb	Right-handed	30	At 12 years of age

Table 14: Smoothness measurement comparing healthy user vs user with minor spinal cord injury. (User A)

User A	SPARC (Linear)	SPARC (Angular)	LDLJ (Linear)	LDLJ (Angular)
Level 1				
Healthy	-15.0648	-16.5873	-16.4036	-17.1783
Spinal Cord Injury	-9.49496	-10.2899	-15.915	-16.4052
Difference	5.56984	6.2974	0.4886	0.7731
Level 2				
Healthy	-8.8484	-8.90046	-16.3254	-16.0477
Spinal Cord Injury	-9.22404	-10.6375	-16.9769	-17.5328
Difference	-0.37564	-1.73704	-0.6515	-1.4851
Level 3				
Healthy	-10.3873	-9.21525	-17.4934	-17.0546
Spinal Cord Injury	-10.5755	-10.6786	-18.1615	-18.6581
Difference	-0.1882	-1.46335	-0.6681	-1.6035
Level 4				
Healthy	-10.5619	-10.2303	-17.548	-17.1992
Spinal Cord Injury	-11.7454	-12.9276	-18.7333	-19.3095
Difference	-1.1835	-2.6973	-1.1853	-2.1103
Level 5				
Healthy	-10.5947	-9.95636	-17.8519	-17.8753
Spinal Cord Injury	-11.0921	-11.8562	-18.5024	-19.0432
Difference	-0.4974	-1.89984	-0.6505	-1.1679

Table 15: Smoothness measurement comparing healthy user vs user with major spinal cord injury (User B trial 1)

User B	SPARC (Linear)	SPARC (Angular)	LDLJ (Linear)	LDLJ (Angular)
Level 1				
Healthy	-15.0648	-16.5873	-16.4036	-17.1783
Spinal Cord Injury	-19.685	-26.5254	-16.658	-16.6885
Difference	-4.6202	-9.9381	-0.2544	0.4898
Level 2				
Healthy	-8.8484	-8.90046	-16.3254	-16.0477
Spinal Cord Injury	-19.4293	-23.8923	-17.6483	-17.3666
Difference	-10.5809	-14.9918	-1.3229	-1.3189
Level 3				
Healthy	-10.3873	-9.21525	-17.4934	-17.0546
Spinal Cord Injury	-12.5405	-12.1977	-18.3131	-18.2126
Difference	-2.1532	-2.98245	-0.8197	-1.158
Level 4				
Healthy	-10.5619	-10.2303	-17.548	-17.1992
Spinal Cord Injury	-9.06415	-11.1813	-18.0392	-18.3019
Difference	1.49775	-0.951	-0.4912	-1.1027

Table 16: Smoothness measurement comparing healthy user vs user with major spinal cord injury (User B trial 2)

User B	SPARC (Linear)	SPARC (Angular)	LDLJ (Linear)	LDLJ (Angular)
Level 1				
Healthy	-15.0648	-16.5873	-16.4036	-17.1783
Spinal Cord Injury	-10.3873	-9.21525	-17.4934	-17.0546
Difference	4.6775	7.37205	-1.0898	0.1237
Level 2				
Healthy	-8.8484	-8.90046	-16.3254	-16.0477
Spinal Cord Injury	-10.3152	-12.6184	-17.3166	-17.459
Difference	-1.4668	-3.71794	-0.9912	-1.4113
Level 3				
Healthy	-10.3873	-9.21525	-17.4934	-17.0546
Spinal Cord Injury	-13.5655	-15.1098	-18.1474	-18.0009
Difference	-3.1782	-5.89455	-0.654	-0.9463
Level 3 Repeat				
Healthy	-10.5619	-10.2303	-17.548	-17.1992
Spinal Cord Injury	-12.8679	-13.0156	-17.9631	-17.6056
Difference	-2.4806	-3.80035	-0.4697	-0.551
Level 4				
Healthy	-10.5947	-9.95636	-17.8519	-17.8753
Spinal Cord Injury	-12.3168	-12.4267	-19.0337	-18.7518
Difference	-1.7549	-2.1964	-1.4857	-1.5526

To demonstrate how close each participant was to an ideal range of smooth movements the graphs in Figure 100, Figure 101, and Figure 102 were drawn for linear SPARC. As it can be seen in Figure 99, User A's data is very close to the idea range due to higher percentage of functional movement. Improvements can be observed for user again going from 18/11/21 to 14/01/2021 dates with the SPARC range of 9.57-9.67 being close to non-injured of 8.85. Figure 101 shows the SPARC for User B where the results are further from the idea line as expected due to lower percentage of movement compared to User A. This can be seen in the SPARC result of 10.27 to 14.63 for User B versus the 8.56 to 10.16 of User A. Improvement can again be seen

for User B trending towards 10.32. Finally, results of User C are the furthest away from the ideal range as expected due to the limited functional movement compared to Users A and B. The improvement can be seen with results changing from 21.85 to 17.69 to a better range of 15.51. Similar analysis can be conducted for linear LDLJ and angular SPARC measurements which can be seen in Appendix C. The result of this study demonstrated that both SPARC and LDLJ can show movement smoothness with SPARC capable of showing the difference on a greater spectrum.

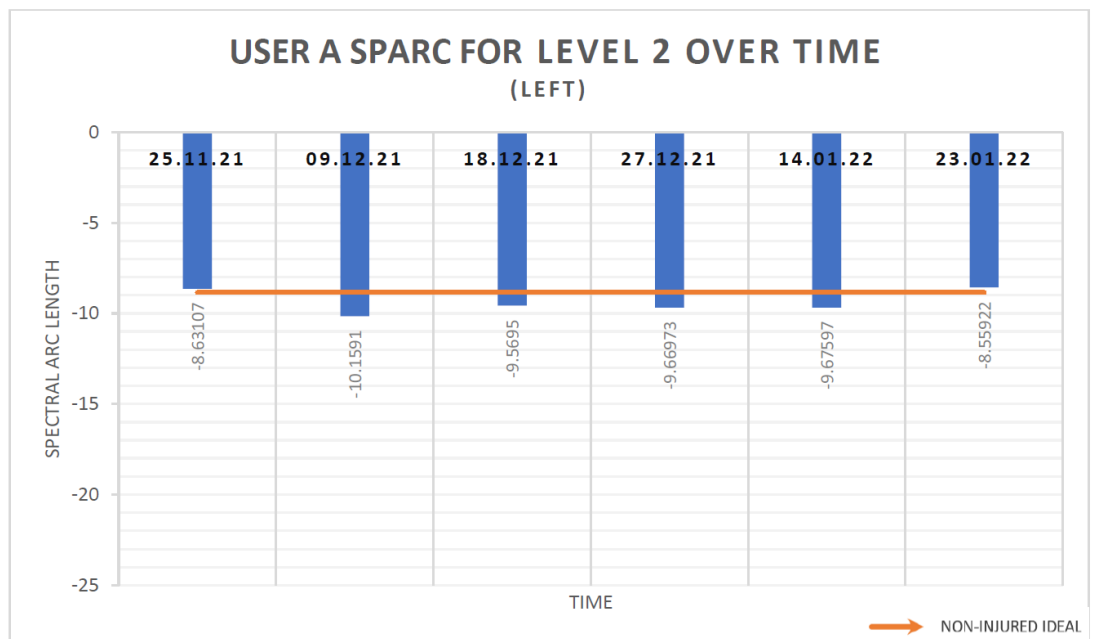


Figure 100: Linear SPARC for User A

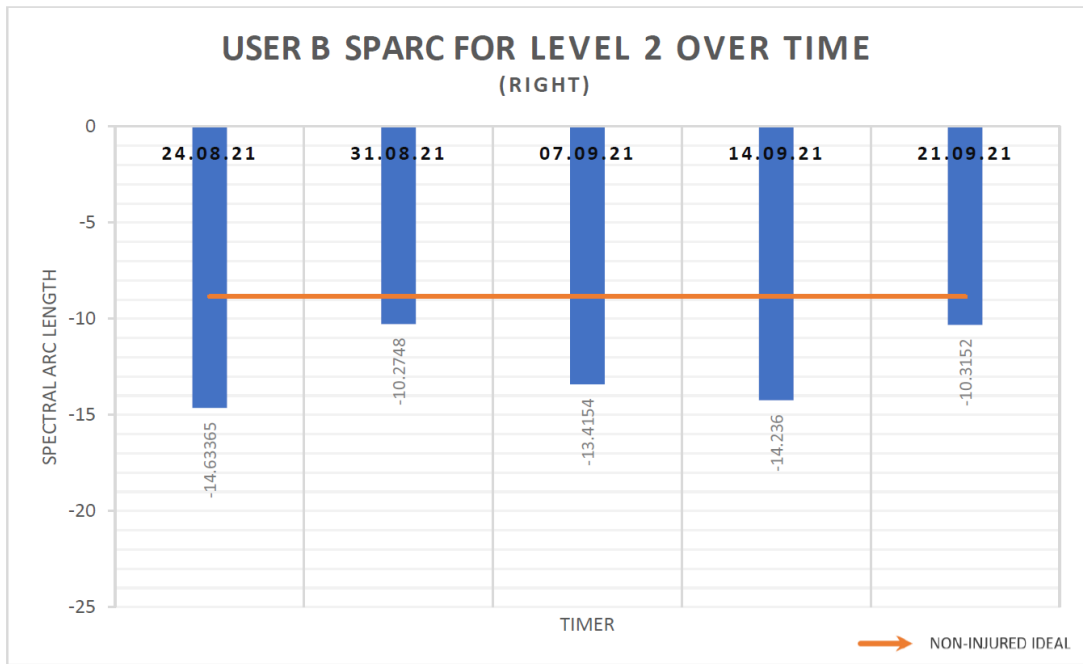


Figure 101: Linear SPARC for User B

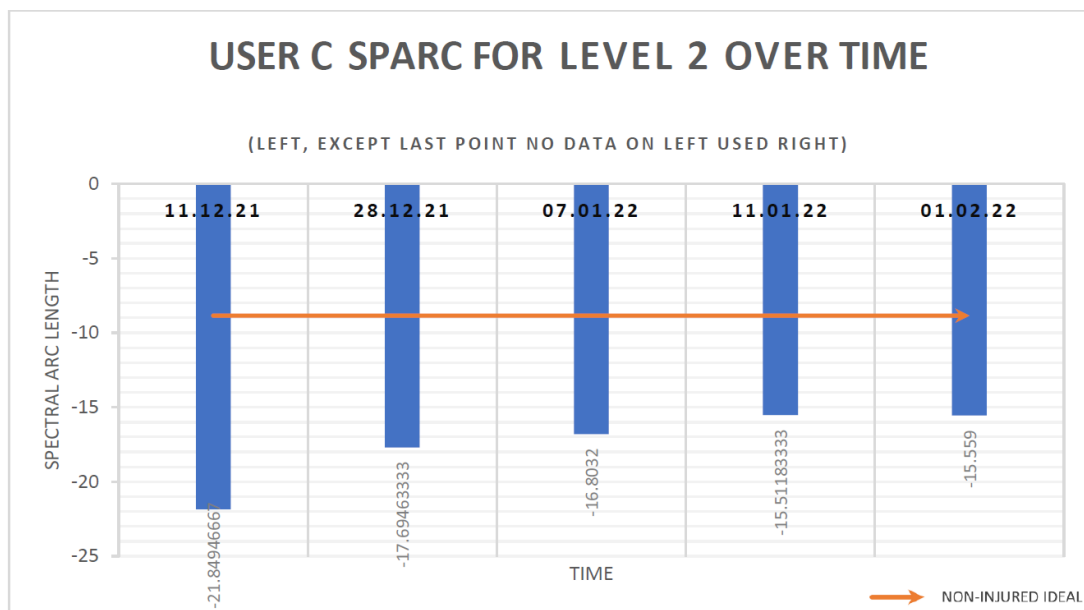


Figure 102: Linear SPARC for User C

5.2.4 Evaluation of HGR Elements

The evaluation process for HGR elements of the framework has been previously discussed in Chapter 4 as part of the alternative control use case and has been published in [43]. In summary, data from HGR methods were validated against a goniometer with consultation from clinical collaborators. As cited in (Khaksar et al.) [43], an

example of this evaluation process was to first capture open palm joint angles using goniometers for each of the following joints of each finger as seen in Table 17:

- Metacarpophalangeal (J1)
- Proximal interphalangeal (J2)
- Distal interphalangeal (J3)

Table 17: Goniometer measurements for open palm [43]

Joint Number	Thumb	Index	Middle	Ring	Pinkie
J1	162	178	175	178	178
J2	177	172	172	170	172
J3	180	177	180	180	180

The same joints were then measured using MPH models and their accuracy was calculated. These measurements were done in categories of open, partially open, and closed palm divided by front, 45-degree, side, and back views. The accuracy of the measurements was then calculated in categories of 2D data and 3D data depending on the viewpoint. Table 18 shows an example of MPH measurements for open palm, Table 19 shows example accuracy by finger, and Table 20 shows the accuracy by viewpoint for the 3D data [43].

Table 18: MPH measurements for open palm (front view) [43]

Joint Number	Thumb		Index		Middle		Ring		Pinkie	
	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D
J1	159	165	163	166	168	172	171	179	166	170
J2	175	175	168	175	169	178	171	173	171	171
J3	166	166	177	178	178	179	174	177	172	174

Table 19: Accuracy of measurements from 3D data by finger [43]

Hand Position	Percentage Accuracy									
	Thumb		Index		Middle		Ring		Pinkie	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Open	96.2%	92.0%	93.8%	78.7%	95.7%	84.0%	95.7%	91.0%	94.4%	85.4%
Partial	92.2%	63.6%	82.0%	66.1%	82.6%	54.2%	86.2%	68.2%	82.6%	63.6%
Closed	89.6%	60.7%	72.3%	50.9%	77.0%	53.1%	78.5%	41.5%	82.3%	44.4%

Table 20: Accuracy of measurements from 3D data by viewpoint [43]

Hand Position	Percentage Accuracy							
	Front		Forty-Five		Side		Back	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Open	96.8%	91.6%	94.5%	84.8%	93.2%	78.7%	96.0%	85.4%
Partial	88.6%	71.0%	90.0%	81.3%	76.1%	63.6%	85.7%	54.2%
Closed	67.8%	41.5%	86.3%	64.3%	84.8%	60.7%	80.7%	51.8%

5.3 ANALYSIS AND SYSTEM TESTING - CLINICAL PERSPECTIVES

5.3.1 Use of Focus Group for Qualitative Feedback

Several clinical trials were conducted as quantitative analysis of engineering elements of the developed framework. Once all the foundations were analysed, a focus group comprising seven experts in the field of rehabilitation was conducted followed by use of narrative analysis as qualitative method of evaluating the framework.

The focus group included seven academics and professionals who work and research in rehabilitation and recovery. Informed consents were provided by each participant when engaging with the focus group discussions. The small number of participants may have implications for the breadth, representation, subjectivity, and uniformity of the discussions, however, due to the participants' expert knowledge and relevant academic qualifications, the discussions were highly focused and specialised in relevant areas. Potential issues related to uniformity arising from differences in academic settings and clinical matters were addressed, as all participants had practical clinical experience in the field of rehabilitation. It is important to note that the focus group was conducted to comment on the gamification aspects of the framework and non-invasive sensor interaction; not the

provided by the focus group can also be related to broader spectrum of emergent technologies.

5.3.2 Research Questions Used in the Focus Group Discussion

The full question list utilised to guide the focus group discussion can be found in Appendix D. There were four set of questions that helped guide the conversation during the focus group. The purpose of each question set has been provided below.

Question Set 1: Introductory Questions

This question asks the participants to introduce themselves and provide information regarding their occupation. The answers would provide the foundation knowledge of the expertise and qualifications available during the discussion. The responses provided facilitated the following information:

- **Establishing background for the participants:**

By requesting participants to introduce themselves and share details about their occupation, this question effectively established the diversity and breadth of professional expertise within the participants.

- **Identifying relevant expertise:**

Through this question, the understanding of participants' professional backgrounds facilitated the specialised knowledge and experience. This was particularly useful when seeking input or insights from specific professional domains.

- **Exploring different perspectives:**

By divulging the participants' professional occupations, as prompted by this question, a deeper insight was gained into their professional roles, responsibilities, and viewpoints. This provided the perspective of the attendant when providing feedback.

- **Promote credibility:**

This question promoted the credibility of the focus group and established the participant's expertise. This aspect holds particular importance when weighing the feedback given by the focus group participants.

Question Set 2: Clinical Implementation Questions

This question gathers feedback on the familiarity of participants with the concept of gamification of rehabilitation and gain insights, opinions, and feedback regarding the implementation of the framework. The first question in this set provided the base understanding of the participants experience in this area as well as feedback on the participants opinions regarding the effects of gamification in rehabilitation. This provided an opportunity for participants to share their perspectives on the benefits and drawbacks of using game elements in rehabilitation based on their experiences. This information helped identify common themes, concerns, or misconceptions related to impact of gamification.

The second question in this set aimed to understand the first impression of the participants of the developed framework right after being presented with a brief demo. These initial thoughts allowed for spontaneous feedback and opinions, leading to valuable insight into first reactions. The question allowed the participants to express positive aspects such as potential advantages or benefits they perceived from the framework, as well as any negative aspects or concerns. This feedback informed refinements and adjustments to the framework to address concerns and capitalize on strengths.

The final question in this set explored participant's perspectives on the practical implementation of the framework within clinical settings and rehabilitation exercise programs. The question also allowed reflection on feasibility and challenges of integrating the framework into existing clinical practices. The participants were also able to provide feedback on potential roadblocks and obstacles that would impede successful implementations. The feedback received here provided guidance in further development of the framework, strategies for overcoming barriers, and maximizing effectiveness of implementation in clinical scenarios.

Question Set 3: Questions Regarding the Usability of the Framework

This question set explores different aspects of usability of the framework. The answers to this question set provided insights and opinions regarding the implementation and use of framework in clinical settings from both user and clinician point of views.

The first question gathered opinions and impressions of the interaction methods to the framework which involved the non-invasive sensors and the RGB camera as the input methods. The effectiveness, accuracy, and level of accessibility of the technologies was also discussed as part of this question.

The second question identified potential challenges or concerns when using sensors in cases where the user has some disabilities. This question also provided an opportunity to gather insights into practical issues, concerns of compatibility, and accessibility requirements.

The third question explored opinions on the benefits and adequacy of remote viewing of live exercises sessions. Participants were able to share their perspectives on advantages of real time monitoring, potential benefits for healthcare providers, and the level of engagement provided by the remote viewing aspects of the framework. Participants were also able to provide their preference for viewing summary of activities with relevant statistic and indicate if this type of information would be sufficient for assessment of user's progress through the rehabilitation program.

The fourth question provided understanding of participant's preferences for frequency of viewing user engagement with the framework. This allowed for receiving insight into how often patient data needs to be monitored for example after each session or on a periodic basis. Additionally, the questions allowed for receiving feedback on benefits of the engagement reports such as progress tracking and trends identification, which could lead to creating tailored interventions based on the gathered data.

The fifth question provided feedback and opinions on the ease of navigational aspects of the framework. The responses to this question provided insight into user friendly and intuitive navigational system aspects of the framework. This allowed to find areas that may be challenging to navigate based on the participants experience in use of technology in clinical settings.

The sixth question provided feedback and opinion on the level of customization provided to the users within the framework. The participants were able to discuss whether more restriction needed to be added to the customization aspects of the framework. This feedback also helped gather insight into different level of freedom that should be given to users depending on their rehabilitation requirements, level of

familiarity with the technology, and the safety requirements for setting boundaries defined by the clinicians.

The seventh question explored participant's preferences regarding how baseline information for rehabilitation exercises would be provided to the user. There participants were presented with the option of inputting the baseline settings in the same unity application as the user or provide the information in a separate simpler application, designed specifically for baseline settings. This question helped with understanding how the framework can be used as part of a normal rehabilitation program.

The eighth question requested participants to provide their gather perspectives on potential use of AI for increase exercise attributes for the user. The goal was to get an understanding of potential parts of the rehabilitation program that could be automated if with use of AI. This question also related to the customisability of the framework.

Question Set 4: Final Thoughts and Wrap up Questions

The final question invited the participants to provide suggestions and recommendations for future developments and improvements of the current framework. This question helped with wrapping up the discussions within the focus group and get a sense of how much of their needs are addressed via the developed framework. These responses helped in defining future directions for implementation of gamification within rehabilitation settings and helped with providing guidelines for future researchers to continue working in this area.

5.3.3 Setting, Population, and Participants of the Focus Group

This focus group was conducted in a meeting room at Curtin University's Bentley campus and provided the options for participants to join in the discussion via a video conference call. The room contained two large tables in the centre where participants and facilitators were seated. A demonstration of the human computer interface aspects was given to the participants, and they were able to physically interact with the gamified exercises before engaging in the discussion portion of the focus group session. After this activity was complete, the author gave a short 10-minute presentation on the overall aims and objectives of the framework which was followed by in depth discussion within the group. The audio for the session was recorded to help

with transcribing the participant responses. After the sessions, narrative analysis was utilised as a qualitative analysis method to analyse the information provided in the focus group discussions.

There was a total of seven participants who were experts in field of rehabilitation. The participants all provided signed consent at the beginning of the session and provided permission for their name and occupation to be fully published as a part of this thesis. Table 21 contains a summary of the participants area of expertise, occupation, qualifications, and academic backgrounds.

Table 21: Information regarding discussion group participants

Participant Name	Occupation and Academic Background
Group One	
Professor Warren Mansell	Professor of Mental Health at the School of Population Health at Curtin University. BA in Natural Sciences, DPhil, DClinPsy in Clinical Psychology
Dr Welber Marinovic	Senior Lecturer, School of Population Health, Discipline of Psychology at Curtin University, BS, PhD Psychology
Professor Kylie Hill	Professor at School of Allied Health at Curtin University. BSc Physiotherapy, PhD
Dr Meg Harrold	Senior Lecturer School of Allied Health at Curtin University BSc Physiotherapy, PhD
Dr Dale Edwick	Senior Physiotherapist at Fiona Stanley Hospital, Lecturer Curtin school of allied health at Curtin University, Senior Physiotherapist at Fiona Stanley Hospital. BS, Hon, PhD in Physiotherapy
Group Two	
Dr Dave Parsons	Lecturer, Curtin School of Allied Health, Accredited Hand Therapist, Allied Health Research Lead at St John of God Health Care. BSc. (Occupational Therapy), MBA, PhD in Occupational Therapy
Ms Eliza Becker	Project Manager at East Metro Health Services. BS, Hon in Physiotherapy

5.3.4 Summary of Responses to the Research Questions

The summary of participant's responses and their implications have been provided below. As mentioned before, narrative analysis was used as the qualitative analysis method for the focus group.

Clinical Implementation Questions: Familiarity with Gamification

Based on the responses, all participants were familiar with gamification in the rehabilitation space, and this was not a new concept to them. Many were already using gamification concepts their work such as putty playing exercise as part of rehabilitation programs. The participants generally agreed that gamification can be effective to drive specific interactions and increase engagement. Dr Parsons noted concerns about previous solutions taking up space in the clinical laboratories and not being efficiently utilised due to the long set up process. He commented positively on the relatively quick set up time of the HMI. Dr Edwick mentioned previous experience with use of virtual reality to help with pain management of people recovering from burn injuries. Other participants commented on use of virtual reality and augmented reality in rehabilitation and most agreed that these technologies present a physical barrier and are too cumbersome for people recovering from injuries such as spinal cord injuries or people with existing conditions such as cerebral palsy. The weight of the current headset technology and effects of it on the rehabilitation program was mentioned such as use for people with spinal cord injuries. Professor Hill commented that physical games are already being used in rehabilitation settings and mentioned that the transition to digital version of the activities could be interesting. This comment led to a discussion about using technology so the physical rehabilitation object could be implemented in an augmented reality type game where the physical game activity gets the added benefit of more information and feedback being provided to the user.

Clinical Implementation Questions: Initial Impressions

Most participants had generally positive feedback on the HMI of the framework. There were initially some concerns regarding the example cases being focused on wrist rehabilitation which were addressed by presenting the full body tracking mode within Unity and demonstration of the posture monitoring use case. Dr Parsons commented that during a rehabilitation visit it would be faster for a clinician to get the joint angles using goniometers, but he could see the benefits of the framework in remote settings where users are partaking in rehabilitation exercises outside the clinical setting. Ms

Becker had concerns about users becoming withdrawn or overzealous if they are constantly being monitored.

Clinical Implementation Questions: Thoughts on Implementing in Rehabilitation Plans

There was a bit of scepticism regarding the logistics of implementing digital gamified exercises in clinical settings. Professor Hill commented that service costs need to become cheaper, or resources need to be freed up to allow for the technology to be fully implemented. Dr Edwick had the same thoughts and noted that if the system is successfully implemented, it could lead to increase rate of patient assessment which would lead to freeing up hospital beds leading to cost savings. Dr Parsons questioned the funding source for implementing this type of system and suggested that it could be spun from value added perspective for private sectors but had concerns about government healthcare systems.

Another interesting discussion point was the comments by Dr Harrold and Dr Parsons that generally physiotherapist and occupational therapist are not usually “Tech Savvy”. Most participants agreed with this comment and mentioned training required for the clinicians to become familiar with the technology aspects of the framework. Dr Parsons commented that old technology is still being used and new system are usually left untouched to alleviate security concerns. The consensus was that the framework can greatly assist telehealth and remote health. Ms Becker commented that usually, similar systems are initiated by clinicians and the inclusion of the data provided by the framework and circulating this information in academic spaces would lead to the technology being more receptive in clinical settings. Ms Becker also noted that the inclusion of progress over time data to both the user and clinician could be use as the evidence of benefits in clinical settings.

Usability Questions: Thoughts on Interaction Method with the Framework

There was a consensus that the use of Camera as the interaction method with the exercises would be more accepted compared to use of IMUs as they add a physical barrier to entry. They further elaborated concerns regarding the physical sensors getting lost when borrowed by the patients. Dr Marinovic and several others suggested that if physical interaction methods were accepted, use of technology such as haptic feedback to measure force could be implemented. Professor Hill commented that there are three main considerations when assessing the effectiveness of a rehabilitation

exercise. These considerations were mentioned as range of movement, quality of movement, and control of movement. It was discussed that the use of IMUs would address all these considerations since joint measurements and smoothness measurements provided with the framework covers ROM and Control considerations. The quality consideration is also addressed by using the reference IMU. Dr Harrold commented that in rehabilitation the use would be required to engage in short but frequent exercise sessions which means the exercise session should limit longer engagement with the framework. Dr Harrold gave the example of a user doing all their exercises repetitions within the span of one hour at the end of the day which can be physically damaging and dangerous. Several other participants mentioned the same requirements and suggested limiting the engagement time with the system so injuries are not overstressed.

Usability Questions: Thoughts on Use of Sensors with People with Disabilities

The participants outline three main concerns in use of sensors with regards to people with disabilities:

1. Small size of sensors leading to them being misplaced.
2. Sanitation requirements of the sensors to prevent infection.
3. Concerns about use cases if patients have open wounds or acute trauma.

This question also opened the discussion regarding other use cases such as aging populations and geriatric users. Professor Hill and Dr Harrold commented the lack of technology skills with this group which requires extensive training to troubleshoot problems and resolve issues. This response led to discussion regarding the importance of proper training materials both included in the framework as a guided modes as well as other training materials to both the user and clinicians.

Usability Questions: Thoughts on Telehealth Implications and Live Viewing of Exercise Sessions

The Telehealth implications and the data provided by the framework outside clinical settings were the most popular aspects of the framework. The participants were excited about the capability to see engagement metrics to see how many times the user engaged with the exercises. The participants highlighted the fact that this data could provide an accountability to the patients claims on the engagement time with the

framework. It was commented that this aspect can help with evaluation of how the exercise program is working. There could be a situation where the engagement data shows the user participating in exercises without any physical improvements. This means any potential problems can be detected early which helps the clinician provide guidelines on correct engagement with the program or investigate any other potential issues. Professor Hill commented that the “Good Patients” are the ones that engage with the program and have continuous improvement overtime whereas more complex cases would take more time in finding out potential issues. Professor Hill commented that the data provided by the framework would help free up clinician time so they can spend more time with the complex cases.

Usability Questions: Thoughts on Frequency Data Provided to the Clintons

The consensus about the frequency of receiving information from the user trended towards weekly. However, it was noted that rehabilitation requirements can vary greatly from patient to patient and there were positive comments that more frequent data was collected if they clinician ever needed to go back and access more information if required. Dr Parsons stated that the system needs to be flexible enough that the frequency can be modified depending on the case. Professor Mansell commented the importance of assigning exercises that are specific to a patient. He also commented that his experience shows that some users will engage with gamification more than others. Professor Mansell was interested in identifying why some users are not engaging with the gamification such as avoiding or fear of pain and discomfort.

Usability Questions: Thoughts on Ease or Difficulty of Navigating the Menus and Settings

All participants were happy with the implemented navigation control, but the importance of a guided mode and tutorials were mentioned again. One of the suggestions was to allow the clinicians to limit the number of choices provided to the user if needed so they can have more control about what aspects of the exercises are modifiable. It was highlighted that different familiarity with technology could lead to more what options are presented to the user. Professor Hill had positive comments regarding the animations and transitions within the framework and mentioned the interface would be engaging to younger users. The possibility of providing touch input for menu selection and navigation was also discussed.

Usability Questions: Thoughts on Level of Customization Provided to the Patients

Generally, most participants preferred to limit the amount of control users would have in changing baselines and attributes within the framework. It was mentioned that clinicians would need direct oversight when changes are made to threshold settings. The suggested example was to only allow the threshold settings to be increment if the required number of repetitions have been met. Which means if a user has not completed their repetitions they should not be allowed to increase the threshold as it may cause further injury. Many were happy to allow the user to pick which mini game they want to engage with as long as it represented a particular rehabilitation exercise. Professor Mansel commented that the games would need to be specialised to the need of the patients and commented positively on the customizable nature of the games within the HMI.

Usability Questions: Thoughts on Requirement for Separate Baseline Setting Application versus a Single All in One Application

The responses were generally centred around a simple method of data entry so the baseline and threshold settings can be provided to the participants without too much time spent with visual menus. The participants suggested context menu, boxes with information and inputs, and slider bars for navigational options of the Settings application. There were some suggestions around having some threshold angles represented on a goniometer since the clinicians would already be familiar with values on the device.

Final Question Regarding Closing Thoughts and Future Development Suggestions

Most participants in group one stated that AI should not make any decisions or changes to the rehabilitation exercise. The second group were more open to the idea but mentioned that their experience shows that the current state of technology does not support too much reliance on AI. Ms Becker and Dr Parsons suggested that legislative changes and extensive research in AI decision making processes need to take place prior to implementation of an automated system. The participants also highlighted the fact that rehabilitation exercises need to be adjusted to user's needs which means a lot of customizability and oversight needs to be considered when making decisions regarding changes in a rehabilitation program.

There was a lot of interest regarding a centralized data system that could promote social engagement. One of the suggestions was to implement systems that can forecast recovery utilising historical data to encourage further engagement with the framework. Dr Parsons suggested that the success rate of rehabilitation programs provided by the framework could potentially be utilised by private practice as advertisement of the effectiveness of their techniques. Dr Parsons also suggested potential use case of the framework in preventing injury such as using the data provided by the framework to track potential cause of injury such as finding causes of Carpal Tunnel Syndrome and providing exercises to help with offsetting any potential injuries.

There were some concerns surrounding data safety and security and suggestion that some clients may not want their data tracked or recorded. The example of workers compensations was brought up and the potential hesitation of engaging with this type of technology.

Professor Hill has some suggestions regarding the integration of this technology in general society much like wearable fitness tracking and tech-based lifestyle applications. She suggested including options for users to compare their progress and engagement with other users in similar rehabilitation environments as a means of increased engagement.

5.4 SUMMARY OF THE ANALYSIS OF RESULTS AND SYSTEM TESTING

5.4.1 Summary of the Evaluation of Engineering Aspects

The first part of this chapter demonstrated quantitative methods that have been utilised to evaluate the technology selected for the development of the framework. Initially Curtin University's Motion Analysis Laboratory (MAL) was utilised a method of comparing the accuracy of the sensors against a gold standard of human movement capture.

Once the sensor data was evaluated, different use cases were defined where engineering aspects of the framework were put to test in clinical environments. As part of these use cases several clinical trials were summarised below:

- **IWHOT and MIT Randomized Control Trials:** These trials provided an opportunity to demonstrate the accuracy of use of IMUs in clinical settings as published in [14].

- HCA Trial: This trial provided an opportunity to evaluate the use of IMUs for capturing head movement of children with CP where the data was compared with a Goniometer as published in [103].
- Smoothness Calculation Trial: This trial provided the opportunity to test and evaluate the results of the smoothness calculations via tests run with participants who had spinal cord injury.

The usability of the framework for posture monitoring as well as evaluation of joint angle measurement technique using the IMUs were demonstrated in the posture monitoring user case found in Chapter 4 and [16].

Finally, HGR elements of the framework were validated against goniometers and utilised as an alternative method of control for a drone system as found in Chapter 4 and [43].

5.4.2 Summary of the Evaluation of Clinical Aspects

The second part of this chapter included details of a qualitative analysis of the framework in form of a focus group discussion with experts in field of rehabilitation. In summary, the participants were interested in the integration and simplification of healthcare through technology and saw strong uses in Telehealth and remote healthcare applications. It was noted that there is a requirement for the framework to cover all three aspects of rehabilitation, namely range, control, and quality of movement. The data captured by the framework caused a lot of excitement where the following use cases were mentioned by the participants.

- The data captured by the framework provided insight into how effective the rehabilitation program is and if improvements to the range of movement is being achieved.
- The data captured by the framework can provide accountability to the user and inform the clinician of how often a patient has engaged with the rehabilitation program.
- The data captured by the framework can provide insight in situations where improvement is being made even though the exercise program is being followed. This would help with early detection of any potential issues.

As mentioned above, there is requirement to include more tutorials and guided modes for both users and clinicians. It was noted that rehabilitation exercises may vary greatly going from user to user which means individual patient requirements needs to be carefully considered when defining gamified rehabilitation programs. This also affect the amount of control and customisation option is available to a user meaning baseline and threshold settings need to be carefully defined by the clinicians. It was also highlighted that the developed framework is usable for different age groups due to ease of navigation in menus and visual representation of different aspects of the framework.

Chapter 6: Conclusions

This thesis presented a framework for gamification of rehabilitation exercises utilising non-invasive sensors. To achieve this, an extensive review of the state-of-the-art human movement capture technologies as well as review of recent trends in rehabilitation and gamified approaches were conducted. It was found that in recent years, physical games have been used to increase engagement with rehabilitation programs and encourage patients to complete the program. Examples of gamification of exercises were found, but the main issue was the lack of clinical validation of data provided by the framework. Most virtual gamified exercises focus on increasing engagement or utilise expensive specialised equipment such as motion analysis labs, or virtual/augmented reality headsets. Review of the literature showed that most of these virtual gamified solutions offer data attributed to joint measurement but there was a lack of clinical validity in their implementation. Moreover, there is no unified method of applying virtual gamification in existing exercises so that the clinicians are not able to rely on the data with high levels of confidence. That is why the research for this thesis started by defining hardware and software components in such a way that real time, clinically validated data can be collected to the benefit of patients and clinicians.

The first step in implementation of the system was the selection of IMUs and ML facilitated by MediaPipe as the data acquisition methods and main form of interaction with the gamified exercises. After defining the data acquisition techniques, algorithms for evaluating the effectiveness of the rehabilitation session were developed so the user can get feedback on the correctness of their exercises outside clinical settings. To achieve this, three main pillars of rehabilitation were considered which are Range of Movement, Control of Movement, and Quality of Movement.

The range of movement, typically assessed using goniometers in clinical environments, aids in establishing a foundational point for rehabilitation programs. Control and quality of movement is usually observed by the clinicians during an in-person patient visit. The challenge lies outside clinical settings where patients do not get any feedback regarding the accuracy of their participation in the exercise regimen. Additionally, the clinicians are not able to continuously monitor the patients' progress

through the rehabilitation program without them physically present which causes challenges if the patient lives in a remote area and is not able to get regular checks at a rehabilitation centre. To be able to provide this information remotely to the clinician and provide feedback to the user, IMUs and ML technologies were implemented. The data from the framework provides joint angle utilising Quaternion orientation frames; provides smoothness measurements through implementation of LDLJ and SPARC; and provides feedback on quality of movement by a mix of IMU data and HGR.

To create an engaging gamified environment, an HMI was developed in Unity game engine that was capable of providing audiovisual feedback to the user. To achieve this, several mini game templates were developed that are mapped to specific rehabilitation exercises. Guidelines are provided on how to apply game theory in building a gamified rehabilitation exercises using the provided templates.

To test the systems, several use cases were defined to demonstrate different elements of the developed framework. The CP use cases demonstrated the use of IMUs and ML in clinical settings. The posture monitoring use case demonstrated the use of the developed framework in full body movement capture and provided information regarding validation of IMU data against Curtin University's motion analysis laboratory and goniometers. Finally, the alternative control system use case demonstrated the use of the HGR elements of the framework for interacting with a physical drone can be mapped to a physical gamified exercise. This use case also contained the validation method for the HGR elements of the framework. These use cases were all published in open access, peer review journals as documented in Chapter 4.

After all the engineering elements were tested and validated, a focus group was conducted with experts in field of rehabilitation to gather qualitative feedback on the developed framework. The participants of the focus group highlighted the level of customization required for rehabilitation exercises. It was noted that the data captured by the framework outside clinical settings can provide insight into how well a patient is progressing through rehabilitation. This data can also facilitate early detection of any potential areas for when a rehabilitation program is not working as expected and can add accountability to the user on how well they have engaged with the exercise program.

At the completion of this research, the main aims and objectives were to design, develop, test, and validate a framework for virtual gamification of rehabilitation exercises which has been achieved. There are opportunities for further development of this research already taking place at Curtin university. One of these potential directions would be use of technology in physical equipment that is currently being used in rehabilitation such as exoskeletons such that the physical interaction can be demonstrated and recorded. Addition of haptic devices could also be a potential future direction which adds another feedback method that can increase the immersion of the virtual games. The author believes that once the physical headsets in AR and VR technology become light weigh, low cost, and more readily available there is great potential in use of this technology as an element for virtual gamification of rehabilitation. As of 2023, physical equipment in VR and AR technology is reportedly still too cumbersome for rehabilitation but has shown great potential in areas such as cognitive abilities or in the psychological therapy space for exposure therapy. However, due to popularity and fast growth of video game industry, these technologies are progressing at a fast paste and soon the barrier to entry will be lowered further. This paints a bright future for virtual gamification of rehabilitation and opens the door for further development in this space.

Bibliography

- [1] WHO. "Rehabilitation by World Health Organization " <https://www.who.int/news-room/fact-sheets/detail/rehabilitation> (accessed 2023).
- [2] J. W. Burke, M. McNeill, D. Charles, P. Morrow, J. Crosbie, and S. McDonough, "Serious Games for Upper Limb Rehabilitation Following Stroke," in *2009 Conference in Games and Virtual Worlds for Serious Applications*, 23-24 March 2009 2009, pp. 103-110, doi: 10.1109/VSGAMES.2009.17.
- [3] J. Janssen, O. Verschuren, W. J. Renger, J. Ermers, M. Ketelaar, and R. van Ee, "Gamification in Physical Therapy: More Than Using Games," (in eng), *Pediatr Phys Ther*, vol. 29, no. 1, pp. 95-99, Jan 2017, doi: 10.1097/pep.0000000000000326.
- [4] J. Melin, "Patient Participation in Physical Medicine and Rehabilitation: A Concept Analysis," *International Physical Medicine & Rehabilitation Journal*, vol. 3, 01/24 2018, doi: 10.15406/ipmrj.2018.03.00071.
- [5] R. L. Gajdosik and R. W. Bohannon, "Clinical measurement of range of motion. Review of goniometry emphasizing reliability and validity," (in eng), *Phys Ther*, vol. 67, no. 12, pp. 1867-72, Dec 1987, doi: 10.1093/ptj/67.12.1867.
- [6] Y.-J. Chang, S.-F. Chen, and J.-D. Huang, "A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities," *Research in Developmental Disabilities*, vol. 32, no. 6, pp. 2566-2570, 2011/11/01/ 2011, doi: <https://doi.org/10.1016/j.ridd.2011.07.002>.
- [7] A. D. Gama, T. Chaves, L. Figueiredo, and V. Teichrieb, "Guidance and Movement Correction Based on Therapeutics Movements for Motor Rehabilitation Support Systems," in *2012 14th Symposium on Virtual and Augmented Reality*, 28-31 May 2012 2012, pp. 191-200, doi: 10.1109/SVR.2012.15.
- [8] P. Herrero, P. Carrera, E. García, E. Gómez-Trullén, and B. Oliván-Blázquez, "Reliability of goniometric measurements in children with cerebral palsy: A comparative analysis of universal goniometer and electronic inclinometer. A pilot study," *BMC Musculoskelet Disord*, vol. 12, p. 155, 07/10 2011, doi: 10.1186/1471-2474-12-155.
- [9] Physiopedia. "Goniometer." <https://www.physio-pedia.com/Goniometer> (accessed 10 October, 2020).
- [10] S. Shultz, P. Houglum, and D. Perrin. "Examination of Physiologic Range of Motion." Human Kinetics, Inc. <https://us.humankinetics.com/blogs/excerpt/measuring-range-of-motion> (accessed 2023).
- [11] P. Rosenbaum *et al.*, "A report: The definition and classification of cerebral palsy April 2006," *Developmental medicine and child neurology. Supplement*, vol. 109, pp. 8-14, 03/01 2007, doi: 10.1111/j.1469-8749.2007.tb12610.x.
- [12] C. P. A. R. Foundation. "How does cerebral palsy affect people " <https://cerebralpalsy.org.au/our-research/about-cerebral-palsy/what-is-cerebral-palsy/how-cerebral-palsy-affects-people/> (accessed 10 October, 2020).

- [13] K. McConnell, L. Johnston, and C. Kerr, "Upper limb function and deformity in cerebral palsy: a review of classification systems," (in eng), *Dev Med Child Neurol*, vol. 53, no. 9, pp. 799-805, Sep 2011, doi: 10.1111/j.1469-8749.2011.03953.x.
- [14] S. Khaksar *et al.*, "Application of Inertial Measurement Units and Machine Learning Classification in Cerebral Palsy: Randomized Controlled Trial," *JMIR Rehabil Assist Technol*, vol. 8, no. 4, p. e29769, 2021/10/20 2021, doi: 10.2196/29769.
- [15] J. H. Medicine. "Back and Neck Pain." Johns Hopkins Medicine. <https://www.hopkinsmedicine.org/health/conditions-and-diseases/back-pain> (accessed).
- [16] S. Khaksar *et al.*, "Posture Monitoring and Correction Exercises for Workers in Hostile Environments Utilizing Non-Invasive Sensors: Algorithm Development and Validation," *Sensors*, vol. 22, no. 24, p. 9618, 2022. [Online]. Available: <https://www.mdpi.com/1424-8220/22/24/9618>.
- [17] Y. Sato, Y. Kobayashi, and H. Koike, "Fast tracking of hands and fingertips in infrared images for augmented desk interface," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, 28-30 March 2000 2000, pp. 462-467, doi: 10.1109/AFGR.2000.840675.
- [18] U. Leap. "Leap Motion Controller " <https://www.ultraLeap.com/product/leap-motion-controller/> (accessed 22.09, 2020).
- [19] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," (in eng), *Sensors (Basel)*, vol. 13, no. 5, pp. 6380-93, May 14 2013, doi: 10.3390/s130506380.
- [20] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "FingerIO: Using Active Sonar for Fine-Grained Finger Tracking," presented at the Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, California, USA, 2016. [Online]. Available: <https://doi.org/10.1145/2858036.2858580>.
- [21] Photonics. "Fiber Optics: Understanding the Basics." https://www.photonics.com/Articles/Fiber_Optics_Understanding_the_Basics/a25151 (accessed 2022).
- [22] R. W. World. "Advantages and Disadvantages of Fiber Optic Sensors." <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Fiber-Optic-Sensor.html> (accessed).
- [23] J. M. Williams, I. Haq, and R. Y. Lee, "Dynamic measurement of lumbar curvature using fibre-optic sensors," *Medical Engineering & Physics*, vol. 32, no. 9, pp. 1043-1049, 2010/11/01/ 2010, doi: <https://doi.org/10.1016/j.medengphy.2010.07.005>.
- [24] M. Stoppa and A. Chiolerio, "Wearable electronics and smart textiles: a critical review," (in eng), *Sensors (Basel)*, vol. 14, no. 7, pp. 11957-92, Jul 7 2014, doi: 10.3390/s140711957.
- [25] A. Fathi and K. Curran, "Detection of spine curvature using wireless sensors," *Journal of King Saud University - Science*, vol. 29, no. 4, pp. 553-560, 2017/10/01/ 2017, doi: <https://doi.org/10.1016/j.jksus.2017.09.014>.
- [26] E. Sardini, M. Serpelloni, and V. Pasqui, "Wireless Wearable T-Shirt for Posture Monitoring During Rehabilitation Exercises," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 439-448, 2015, doi: 10.1109/TIM.2014.2343411.

- [27] F. Caputo, A. Greco, E. D'Amato, I. Notaro, and S. Spada, "Imu-based motion capture wearable system for ergonomic assessment in industrial environment," in *Advances in Human Factors in Wearable Technologies and Game Design: Proceedings of the AHFE 2018 International Conferences on Human Factors and Wearable Technologies, and Human Factors in Game Design and Virtual Environments, Held on July 21–25, 2018, in Loews Sapphire Falls Resort at Universal Studios, Orlando, Florida, USA 9*, 2019: Springer, pp. 215-225.
- [28] S. S. Bidabadi, I. Murray, and G. Y. F. Lee, "The clinical application of inertial measurement unit in identification of foot drop symptoms," in *2017 IEEE 15th Student Conference on Research and Development (SCORED)*, 13-14 Dec. 2017 2017, pp. 183-186, doi: 10.1109/SCORED.2017.8305397. [Online]. Available: <https://ieeexplore.ieee.org/document/8305397/>
- [29] N. Abhayasinghe and I. Murray, "Human activity recognition using thigh angle derived from single thigh mounted IMU data," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 27-30 Oct. 2014 2014, pp. 111-115, doi: 10.1109/IPIN.2014.7275474.
- [30] Xsense. "Xsenser overview of IMUs." <https://www.xsens.com/inertial-sensor-modules> (accessed April 23, 2020).
- [31] Movella. "What is Motion Capture." <https://www.xsens.com/motion-capture> (accessed 28 October, 2020).
- [32] M.-S. Academy. "What is motion capture and how does it work?" MO-SYS Academy <https://www.mo-sys.com/what-is-motion-capture-and-how-does-it-work/> (accessed 2022).
- [33] D. Thewlis, C. Bishop, N. Daniell, and G. Paul, "A comparison of two commercially available motion capture systems for gait analysis: high-end vs low-cost," presented at the Congress of the International Society of Biomechanics, Brussels, Belgium, 2011. [Online]. Available: <https://eprints.qut.edu.au/49083/>.
- [34] Vicon. "Vicon Product Page." <https://www.vicon.com/> (accessed 07 September, 2020).
- [35] A. Kostas. "Frame Rotations and Representations." <http://www.kostasalexis.com/frame-rotations-and-representations.html> (accessed 2022).
- [36] Á. López Revuelta, "Orientation estimation and movement recognition using low cost sensors," ed, 2017.
- [37] N. Surla, "Fall Detection for Elderly using IMU Based Gait Analysis," Bachelor of Engineering in Electrical Engineering, Curtin University, 2021.
- [38] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami, and E. Burdet, "On the analysis of movement smoothness," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, no. 1, p. 112, 2015/12/09 2015, doi: 10.1186/s12984-015-0090-9.
- [39] A. Melendez-Calderon, C. Shirota, and S. Balasubramanian, "Estimating Movement Smoothness From Inertial Measurement Units," (in eng), *Front Bioeng Biotechnol*, vol. 8, p. 558771, 2020, doi: 10.3389/fbioe.2020.558771.
- [40] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet, "A Robust and Sensitive Metric for Quantifying Movement Smoothness," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 8, pp. 2126-2136, 2012, doi: 10.1109/TBME.2011.2179545.

- [41] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311-324, 2007, doi: 10.1109/TSMCC.2007.893280.
- [42] P. Prashan, *Human Computer Interaction Using Hand Gestures*: Springer Singapore.
- [43] S. Khaksar, L. Checker, B. Borazjan, and I. Murray, "Design and Evaluation of an Alternative Control for a Quad-Rotor Drone Using Hand-Gesture Recognition," *Sensors*, vol. 23, no. 12, p. 5462, 2023. [Online]. Available: <https://www.mdpi.com/1424-8220/23/12/5462>.
- [44] M. Yasen and S. Jusoh, "A systematic review on hand gesture recognition techniques, challenges and applications," (in eng), *PeerJ Comput Sci*, vol. 5, p. e218, 2019, doi: 10.7717/peerj-cs.218.
- [45] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, p. Article 16, 2011, doi: 10.1145/1922649.1922653.
- [46] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355-367, 2018/11/01/ 2018, doi: <https://doi.org/10.1016/j.ergon.2017.02.004>.
- [47] L. Guo, L. Zongxing, and L. Yao, "Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review," *IEEE Transactions on Human-Machine Systems*, vol. PP, pp. 1-10, 06/23 2021, doi: 10.1109/THMS.2021.3086003.
- [48] I. El Naqa and M. J. Murphy, "What Is Machine Learning?," in *Machine Learning in Radiation Oncology: Theory and Applications*, I. El Naqa, R. Li, and M. J. Murphy Eds. Cham: Springer International Publishing, 2015, pp. 3-11.
- [49] V. Nath and J. Mandal, *Nanoelectronics, Circuits and Communication Systems Proceeding of NCCS 2018: Proceeding of NCCS 2018*. 2020.
- [50] S. Jiang, P. Kang, X. Song, B. Lo, and P. Shull, "Emerging Wearable Interfaces and Algorithms for Hand Gesture Recognition: A Survey," (in eng), *IEEE Rev Biomed Eng*, vol. 15, pp. 85-102, 2022, doi: 10.1109/rbme.2021.3078190.
- [51] M. Oudah, A. Al-Naji, and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," *Journal of Imaging*, vol. 6, no. 8, p. 73, 2020. [Online]. Available: <https://www.mdpi.com/2313-433X/6/8/73>.
- [52] S. Jiang, P. Kang, X. Song, B. P. L. Lo, and P. B. Shull, "Emerging Wearable Interfaces and Algorithms for Hand Gesture Recognition: A Survey," *IEEE Reviews in Biomedical Engineering*, vol. 15, pp. 85-102, 2022, doi: 10.1109/RBME.2021.3078190.
- [53] S. Rautaray and A. Agrawal, "VISION-BASED APPLICATION-ADAPTIVE HAND GESTURE RECOGNITION SYSTEM," *International Journal of Information Acquisition*, vol. 09, 04/03 2013, doi: 10.1142/S0219878913500071.
- [54] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial intelligence review*, vol. 43, pp. 1-54, 2015.
- [55] F. Zhang *et al.*, "Mediapipe hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.
- [56] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," in *Computer Vision–ECCV 2020: 16th European Conference*,

- Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, 2020: Springer, pp. 548-564.
- [57] Y. Katsuki, Y. Yamakawa, and M. Ishikawa, "High-speed Human / Robot Hand Interaction System," *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, 2015.
- [58] L. Ge *et al.*, *3D Hand Shape and Pose Estimation From a Single RGB Image*. 2019, pp. 10825-10834.
- [59] Indriani, H. Moh, and A. Ali Suryaperdana, "Applying Hand Gesture Recognition for User Guide Application Using MediaPipe," in *Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021)*, 2021/11/23 2021: Atlantis Press, pp. 101-108, doi: 10.2991/aer.k.211106.017. [Online]. Available: <https://doi.org/10.2991/aer.k.211106.017>
- [60] J. Sanalohit and T. Katanyukul, "TFS Recognition: Investigating MPH}{Thai Finger Spelling Recognition: Investigating MediaPipe Hands Potentials," *ArXiv*, vol. abs/2201.03170, 2022.
- [61] M. Jindal, E. Bajal, and S. Sharma, "A Comparative Analysis of Established Techniques and Their Applications in the Field of Gesture Detection," *Machine Learning Algorithms and Applications in Engineering*, p. 73, 2023.
- [62] S. Bhushan, M. Alshehri, I. Keshta, A. K. Chakraverti, J. Rajpurohit, and A. Abugabah, "An Experimental Analysis of Various Machine Learning Algorithms for Hand Gesture Recognition," *Electronics*, vol. 11, no. 6, p. 968, 2022. [Online]. Available: <https://www.mdpi.com/2079-9292/11/6/968>.
- [63] RACP. "Paediatrics & Child Health Division (PCHD)" <https://www.racp.edu.au/about/college-structure/paediatrics-child-health-division> (accessed 2022).
- [64] M. J. Stoner and M. D. Lanker, "Five Key Injuries of the Pediatric Wrist and Hand," *Pediatric Emergency Medicine Reports*, 2012. [Online]. Available: <https://www.reliasmedia.com/articles/77364-five-key-injuries-of-the-pediatric-wrist-and-hand>.
- [65] N. Barrett, I. Swain, C. Gatzidis, and C. Mecheraoui, "The use and effect of video game design theory in the creation of game-based systems for upper limb stroke rehabilitation," (in eng), *J Rehabil Assist Technol Eng*, vol. 3, p. 2055668316643644, Jan-Dec 2016, doi: 10.1177/2055668316643644.
- [66] L. A. Koman and B. P. Smith, "Surgical management of the wrist in children with cerebral palsy and traumatic brain injury," (in eng), *Hand (N Y)*, vol. 9, no. 4, pp. 471-7, Dec 2014, doi: 10.1007/s11552-014-9636-8.
- [67] B. Barbara. "Understanding Active vs Passive Exercises to Customize Rehabilitation to Your Ability Level." <https://www.flintrehab.com/active-vs-passive-exercises-during-rehab/> (accessed).
- [68] J. Janssen, O. Verschuren, W. J. Renger, J. Ermers, M. Ketelaar, and R. van Ee, "Gamification in Physical Therapy: More Than Using Games," *Pediatric Physical Therapy*, vol. 29, no. 1, pp. 95-99, 2017, doi: 10.1097/pep.0000000000000326.
- [69] F. Corona *et al.*, "Serious Games for Wrist Rehabilitation in Juvenile Idiopathic Arthritis," in *2018 IEEE Games, Entertainment, Media Conference (GEM)*, 15-17 Aug. 2018 2018, pp. 35-42, doi: 10.1109/GEM.2018.8516458.
- [70] B. Bonnechère *et al.*, "Can serious games be incorporated with conventional treatment of children with cerebral palsy? A review," *Research in*

- developmental disabilities*, vol. 35, 04/30 2014, doi: 10.1016/j.ridd.2014.04.016.
- [71] B. Steiner, L. Elgert, B. Saalfeld, and K. H. Wolf, "Gamification in Rehabilitation of Patients With Musculoskeletal Diseases of the Shoulder: Scoping Review," (in eng), *JMIR Serious Games*, vol. 8, no. 3, p. e19914, Aug 25 2020, doi: 10.2196/19914.
- [72] I. Afyouni *et al.*, "A therapy-driven gamification framework for hand rehabilitation," *User Modeling and User-Adapted Interaction*, vol. 27, 06/01 2017, doi: 10.1007/s11257-017-9191-4.
- [73] M. Pirovano, R. Mainetti, G. Baud-Bovy, P. L. Lanzi, and N. A. Borghese, "Self-adaptive games for rehabilitation at home," in *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, 11-14 Sept. 2012 2012, pp. 179-186, doi: 10.1109/CIG.2012.6374154.
- [74] C. See, "Gamification in Anatomy Education," in *Teaching Anatomy: A Practical Guide*, L. K. Chan and W. Pawlina Eds. Cham: Springer International Publishing, 2020, pp. 63-71.
- [75] J. Schell, *The Art of Game Design: A Book of Lenses, Third Edition*. A K Peters/CRC Press, 2019, p. 652.
- [76] M. S. Cameirão, S. B. i. Badia, E. D. Oller, and P. F. M. J. Verschure, "Neurorehabilitation using the virtual reality based Rehabilitation Gaming System: methodology, design, psychometrics, usability and validation," *Journal of NeuroEngineering and Rehabilitation*, vol. 7, no. 1, p. 48, 2010/09/22 2010, doi: 10.1186/1743-0003-7-48.
- [77] M. E. Nixon and A. M. Howard, "Applying Gaming Principles to Virtual Environments for Upper Extremity Therapy Games," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 13-16 Oct. 2013 2013, pp. 3430-3435, doi: 10.1109/SMC.2013.585.
- [78] S. García-Martínez, F. Orihuela-Espina, L. E. Sucar, A. L. Moran, and J. Hernández-Franco, "A design framework for arcade-type games for the upper-limb rehabilitation," in *2015 International Conference on Virtual Rehabilitation (ICVR)*, 9-12 June 2015 2015, pp. 235-242, doi: 10.1109/ICVR.2015.7358573.
- [79] T. V. V. Batista, L. d. S. Machado, A. M. G. Valença, and R. M. d. Moraes, "FarMyo: A Serious Game for Hand and Wrist Rehabilitation Using a Low-Cost Electromyography Device," *International Journal of Serious Games*, vol. 6, no. 2, pp. 3 - 19, 06/24 2019, doi: 10.17083/ijsg.v6i2.290.
- [80] A. Gouaich, N. Hocine, L. Dokkum, and D. Mottet, *Digital-Pheromone Based Difficulty Adaptation in Post-Stroke Therapeutic Games*. 2012.
- [81] J. W. Krakauer, "Motor learning: its relevance to stroke recovery and neurorehabilitation," (in eng), *Curr Opin Neurol*, vol. 19, no. 1, pp. 84-90, Feb 2006, doi: 10.1097/01.wco.0000200544.29915.cc.
- [82] R. Magill and D. Anderson, *Motor Learning and Control: Concepts and Applications*. McGraw-Hill Education, 2020.
- [83] R. A. Wise, "Dopamine, learning and motivation," *Nature Reviews Neuroscience*, vol. 5, no. 6, pp. 483-494, 2004/06/01 2004, doi: 10.1038/nrn1406.
- [84] C. Cruz, M. D. Hanus, and J. Fox, "The need to achieve: Players' perceptions and uses of extrinsic meta-game reward systems for video game consoles," *Computers in Human Behavior*, vol. 71, pp. 516-524, 2017/06/01/ 2017, doi: <https://doi.org/10.1016/j.chb.2015.08.017>.

- [85] K. Lohse, N. Shirzad, A. Verster, N. Hodges, and H. F. Van der Loos, "Video games and rehabilitation: using design principles to enhance engagement in physical therapy," (in eng), *J Neurol Phys Ther*, vol. 37, no. 4, pp. 166-75, Dec 2013, doi: 10.1097/npt.000000000000017.
- [86] N. Lazzaro, "Why We Play : Affect and the Fun of Games—Designing Emotions for Games, Entertainment Interfaces, and Interactive Products," 2012.
- [87] E. Adams. "Designer's Notebook: A Symmetry Lesson." <https://www.gamedeveloper.com/disciplines/designer-s-notebook-positive-feedback> (accessed 2022).
- [88] M. Csikszentmihalyi, "Flow: The Psychology of Optimal Experience," 1990.
- [89] S. Jones. "5 WAYS BLUETOOTH 5 MAKES WIRELESS AUDIO BETTER." <https://pro.harman.com/insights/author/sjones/> (accessed 2022).
- [90] M. H. Song and R. I. Godøy, "How Fast Is Your Body Motion? Determining a Sufficient Frame Rate for an Optical Motion Tracking System Using Passive Markers," (in eng), *PLoS One*, vol. 11, no. 3, p. e0150993, 2016, doi: 10.1371/journal.pone.0150993.
- [91] J. Antonio Santoyo-Ramón, E. Casilari, and J. Manuel Cano-García, "A study of the influence of the sensor sampling frequency on the performance of wearable fall detectors," *Measurement*, vol. 193, p. 110945, 2022/04/01/ 2022, doi: <https://doi.org/10.1016/j.measurement.2022.110945>.
- [92] Xsense. "Xsense DOT Sensor." <https://www.xsens.com/xsens-dot> (accessed April 23, 2020).
- [93] Vicon. "Blue Trident." <https://www.vicon.com/hardware/blue-trident/> (accessed 2022).
- [94] Shimmer. "Shimmer IMU." <https://shimmersensing.com/wearable-sensor-products/development-kits/> (accessed 2022).
- [95] B. Systems. "QuantiMotion Full Body Set." <https://store.bonsai-systems.com/en/motion-capturing/9-quantimotion-full-body-set.html> (accessed 2022).
- [96] W. Yan, Q. Zhang, L. Wang, Y. Mao, A. Wang, and C. Zhao, "A modified kalman filter for integrating the different rate data of gyros and accelerometers retrieved from android smartphones in the GNSS/IMU coupled navigation," *Sensors*, vol. 20, no. 18, p. 5208, 2020.
- [97] S. J. Hall, "Linear Kinematics of Human Movement," in *Basic Biomechanics*, 7e. New York, NY: McGraw-Hill Education, 2015.
- [98] K. L. Whitcroft, L. Massouh, R. Amirfeyz, and G. Bannister, "Comparison of Methods of Measuring Active Cervical Range of Motion," *Spine*, vol. 35, no. 19, pp. E976-E980, 2010, doi: 10.1097/BRS.0b013e3181cd6176.
- [99] Blender. "What is Blender? ." <https://www.blender.org/about/> (accessed 2022).
- [100] Google. "MediaPipe Hand Landmarks Dection Guide." https://developers.google.com/mediapipe/solutions/vision/hand_landmarker (accessed 2023).
- [101] Adobe. "Adobe Maximo Free Avatar Package " <https://www.mixamo.com/> (accessed).
- [102] T. S. Kim *et al.*, "A study on the measurement of wrist motion range using the iPhone 4 gyroscope application," (in eng), *Ann Plast Surg*, vol. 73, no. 2, pp. 215-8, Aug 2014, doi: 10.1097/SAP.0b013e31826eabfe.
- [103] S. S. Al-azzawi, S. Khaksar, E. K. Hadi, H. Agrawal, and I. Murray, "HeadUp: A Low-Cost Solution for Tracking Head Movement of Children with Cerebral

Palsy Using IMU," *Sensors*, vol. 21, no. 23, p. 8148, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/23/8148>.

Appendices

APPENDIX A: AUTHOR ATTRIBUTIONS

This appendix includes the contribution statement for all the publications related to this PhD. The contributions for journal papers were published in the paper and have been brought here for ease of access.

Contribution Statement for Paper 1

Statement: Conceptualization of the study was by IM and SK. Data curation was conducted by CE, CI, AC, and CW. Formal analysis was conducted by SK. Investigation was conducted by SK and HA. Methodology was devised by SK and IM. Resources were provided by CE and CI. Software provision was by SK and HP. Supervision was conducted by IM and WL. Validation was carried out by SK. Visualization was conducted by SK and BB. Original draft was written by SK. BB, HA, CE, and CI were involved in reviewing and editing the paper.

Authors: Khaksar, S., H. Pan, B. Borazjani, I. Murray, H. Himanshu, W. Liu, C. Elliott, C. Imms, A. Campbell, and C. Walmsley. 2021. "Application of Inertial Measurement Units and Machine Learning Classification in Cerebral Palsy: Randomized Controlled Trial." *JMIR Rehabilitation and Assistive Technologies* 8 (4)

Table 22: Author Signatures for Paper 1

Author Name	Author Signature	Date
1. Siavash Khaksar		02.12.2021
2. Huizhu Pan		02.12.2021
3. Bitra Borazjani		02.12.2021
4. Iain Murray		02.12.2021
5. Himanshu Agrawal		02.12.2021
6. Wanquan Liu		02.12.2021
7. Catherine Elliott		02.12.2021
8. Christine Imms		02.12.2021
9. Amity Campbell		02.12.2021
10. Corrin Walmsley		02.12.2021

Contribution Statement for Paper 2

Statement: S.S.A.-a. contributed to the design, data curation and implementation. S.S.A.-a., S.K., and E.K.H. contributed to formal analysis and review of results. S.S.A.-a., S.K., I.M., and H.A. contributed to supervision and conceptualization of this paper. S.S.A.-a. and E.K.H. contributed to data collection, measurement validation. The article was written and edited by S.S.A.-a., reviewed by S.S.A.-a., S.K., H.A., and I.M. All authors discussed the results and reviewed and commented on the published version of the manuscript.

Authors: Sabah Al-azzawi, S., S. Khaksar, E. Khdhair Hadi, H. Himanshu, and I. Murray. 2021. "HeadUp: A Low-Cost Solution for Tracking Head Movement of Children with Cerebral Palsy Using IMU." MDPI Sensors 21 (23)

Table 23: Author Signatures for Paper 2

Author Name	Author Signature	Date
1. Sana Sabah Al-azzawi		08.08.2022
2. Siavash Khaksar		08.08.2022
3. Emad Khdhair Hadi		
4. Himanshu Agrawal		08.08.2022
5. Iain Murray		08.08.2022

Contribution Statement for Paper 3

Statement: Conceptualization, I.M. and S.K.; data curation, S.P. and S.K.; formal analysis, S.K. and S.P.; investigation, S.K., S.P. and B.B.; methodology, S.K. and I.M.; software, S.P., S.K., J.H., H.B. and A.K.; supervision, I.M. and S.K.; validation, S.K., S.P., B.B., A.C. and J.H.; visualization, S.K., S.P. and B.B.; original draft, S.K., S.P. and B.B.; review and editing, I.M., A.C., S.K. and B.B. All authors have read and agreed to the published version of the manuscript.

Authors: Khaksar, S., S. Pieters, B. Borazjani, J. Hyde, H. Booker, A. Khokhar, I. Murray, and A. Campbell. 2022. "Posture Monitoring and Correction Exercises for Workers in Hostile Environments Utilizing Non-Invasive Sensors: Algorithm Development and Validation." MDPI Sensors 22 (24,9618)

Table 24: Author Signatures for Paper 3

Author Name	Author Signature	Date
1. Siavash Khaksar		03.04.2023
2. Stefanie Pieters		03.04.2023
3. Bitra Borazjani		03.04.2023
4. Joshua Hyde		31.03.2023
5. Harrison Booker		01.04.2023
6. Adil Khokhar		15.04.2023
7. Iain Murray		03.04.2023
8. Amity Campbell		03.04.2023

Contribution Statement for Paper 4

Statement: Conceptualization, I.M. and S.K.; data curation, L.C. and S.K.; formal analysis, S.K. and L.C.; investigation, S.K., L.C. and B.B.; methodology, S.K. and I.M.; software, L.C. and S.K.; supervision, I.M. and S.K.; validation, S.K., L.C. and B.B.; visualization, S.K., L.C. and B.B.; original draft, S.K., L.C. and B.B.; review and editing, I.M., L.C., S.K. and B.B. All authors have read and agreed to the published version of the manuscript.

Authors: Khaksar, S. L.Checker, B.Borazjani, I.Murray, 2023 “Design and Evaluation of an Alternative Control for a Quad-Rotor Drone using Hand Gesture Recognition” MDPI Sensors 2023

Table 25: Author Signatures for Paper 4

Author Name	Author Signature	Date
1. Siavash Khaksar		18.07.2023
2. Luke Checker		18.07.2023
3. Bita Borazjani		18.07.2023
4. Iain Murray		18.07.2023

Contribution Statement for Invited Conference 1:

Statement: Conceptualization, I.M., S.K., and N.A.; data curation, I.M., S.K., and N.A.; formal analysis, S.K., N.A, B.B., N.M., S.S., and N.F.; investigation, S.K., N.A, B.B., N.M., S.S., and N.F.; methodology, I.M., S.K., A.N, B.B., N.M., S.S., and N.F.; software, S.K., N.A, B.B., N.M., S.S., and N.F. and S.K.; supervision, I.M., A.N., and S.K.; validation, I.M., S.K., and N.A.; visualization, I.M., S.K., and N.A.; original draft, I.M., S.K., A.N, B.B., N.M., S.S., and N.F; review and editing, I.M., S.K., B.B., and A.N. All authors have read and agreed to the published version of the manuscript.

Authors: Khaksar, S, I. Murray, N. Abhayasinghe, B. Borazjani, N. Mohammadi, S. Sharif, N. Fernando. “Engineering Applications in Rehabilitation and Assistive Technology” SLIIT International Conference on Engineering and Technology in Sri Lanka Institute of Information Technology 10th of February 2022

Table 26: Author Signatures for Invited Conference 1

Author Name	Author Signature	Date
1. Siavash Khaksar		14.06.2023
2. Iain Murray		14.06.2023
3. Nimsiri Abhayasinghe		14.06.2023
4. Bitra Borazjani		14.06.2023
5. Nazanin Mohammadi		04.07.2023
6. Shiva Sharif		14.06.2023
7. Nimalika Fernando		30.06.2023

Contribution Statement for Invited Conference 2:

Statement: This conference was presented solely by the author so there are no other author contributions.

Author: Khaksar, S. "A framework for gamification of remote wrist rehabilitation exercises utilising non-invasive sensors" Indo-Asia Digital Health & Telemedicine Summit 22nd March 2023

Table 27: Author Signatures for Invited Conference 2

Author Name	Author Signature	Date
1. Siavash Khaksar		24.03.2023

APPENDIX B: SAMPLE DATA RECORDINGS FROM DEVELOPED GAMES

This appendix provides sample data recordings from games discussed in chapter 3.5.

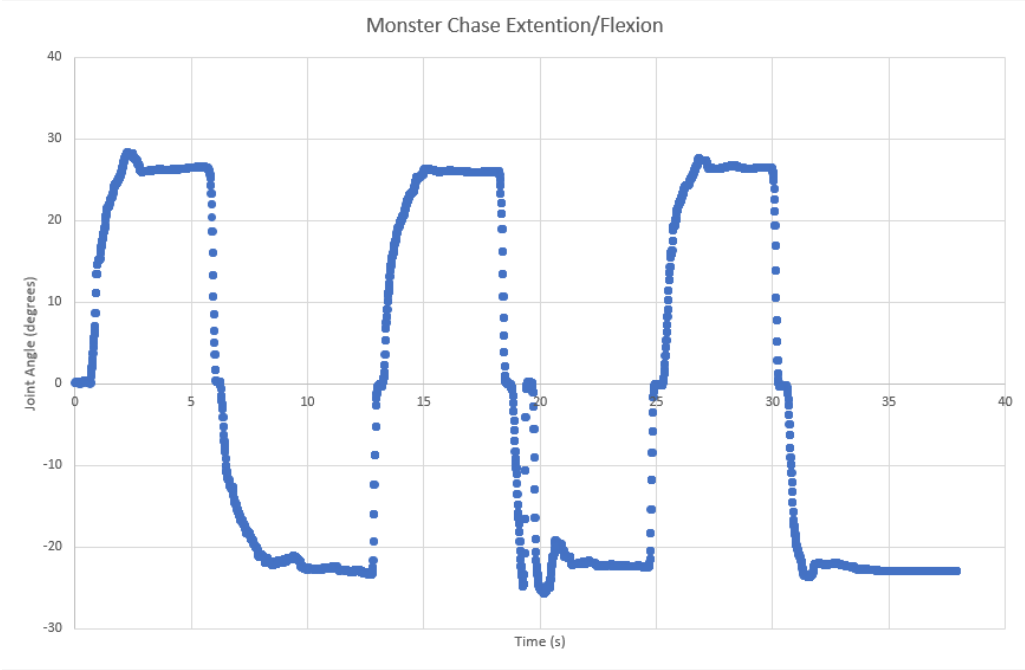


Figure 103: Sample extension/flexion data for the Monster Chase game

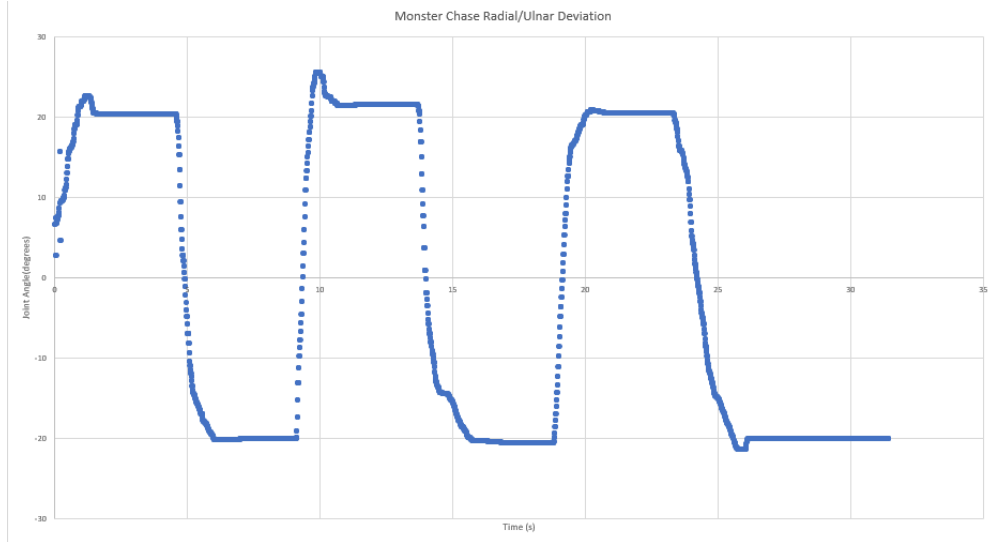


Figure 104: Sample radial/ulnar deviation data for the Monster Chase game

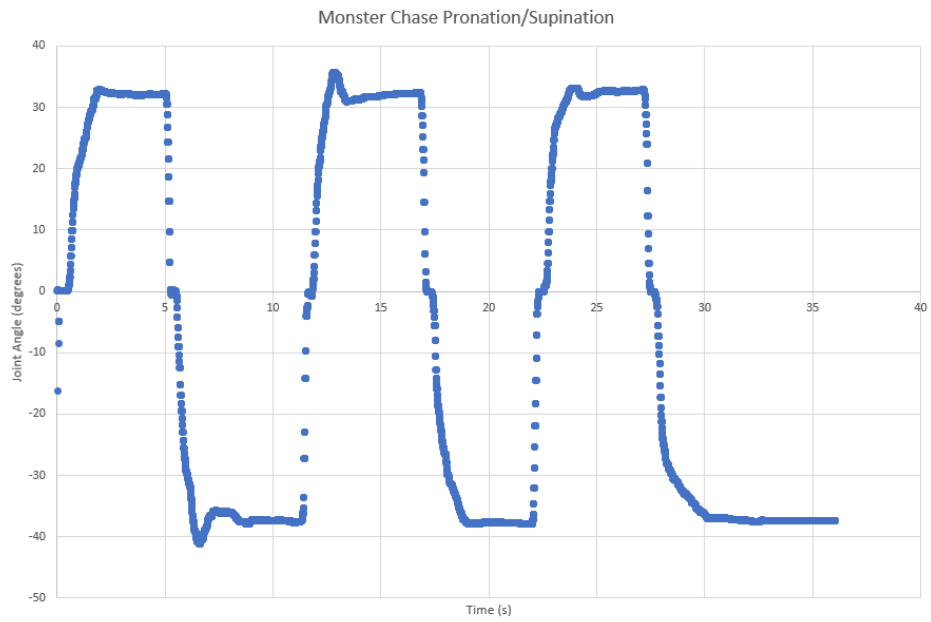


Figure 105: Sample pronation/supination data for the Monster Chase game

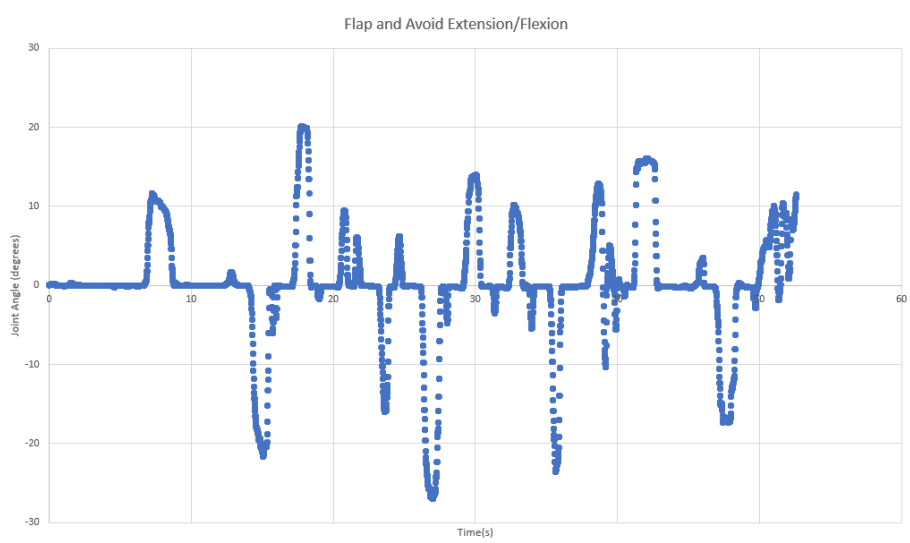


Figure 106: Sample flexion/extension data for the Flap and Avoid game.

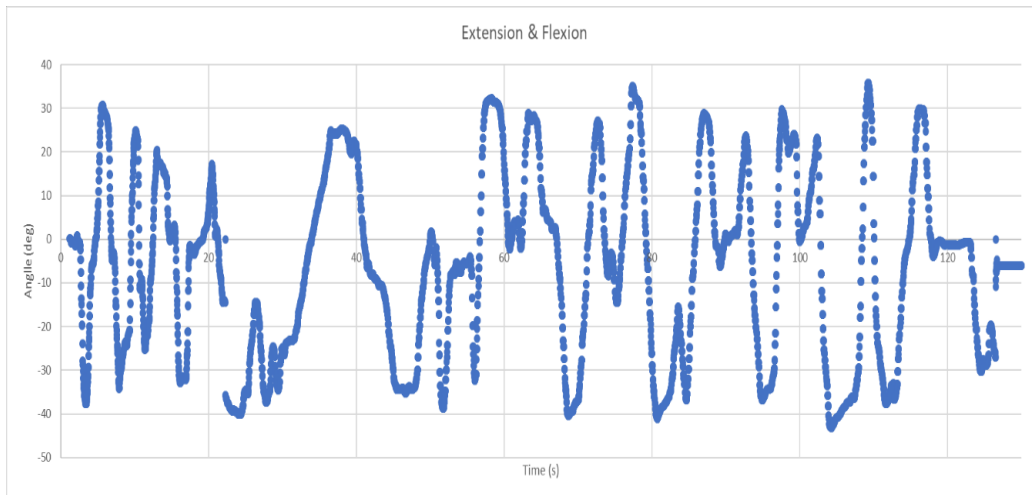


Figure 107: Sample flexion/extension data for the Hovercraft game

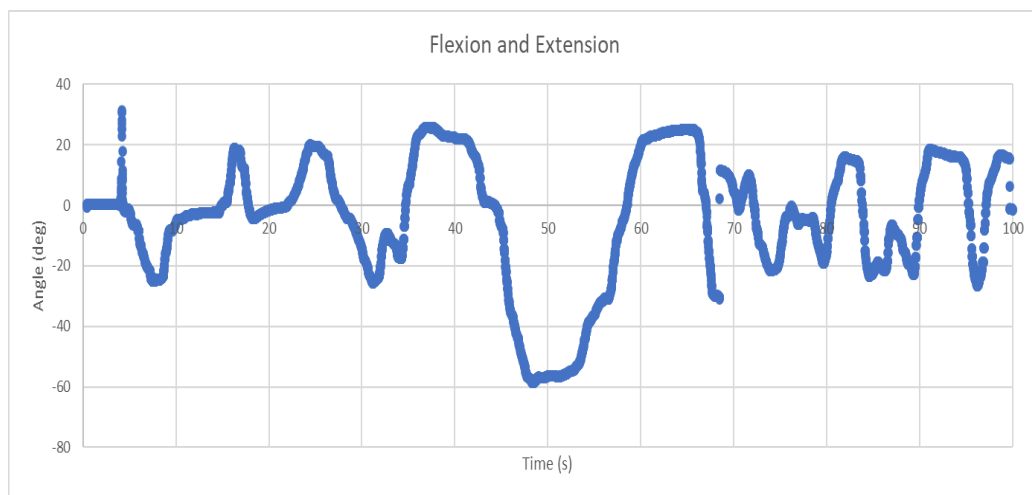


Figure 108: Sample flexion/extension data for the Hovercraft game with larger obstacles

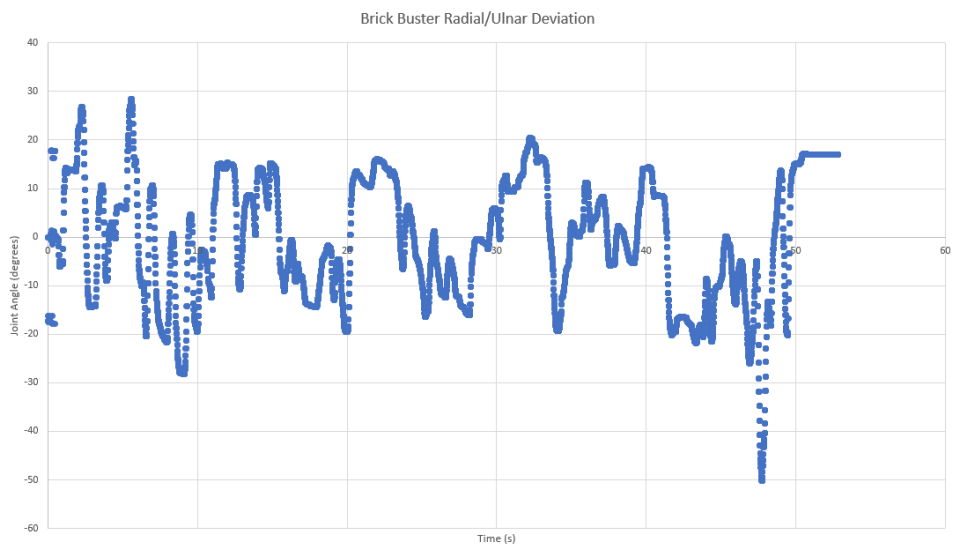
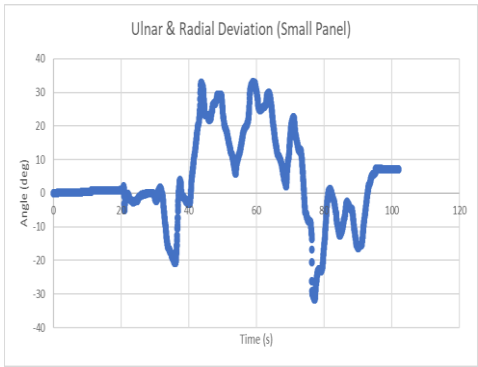
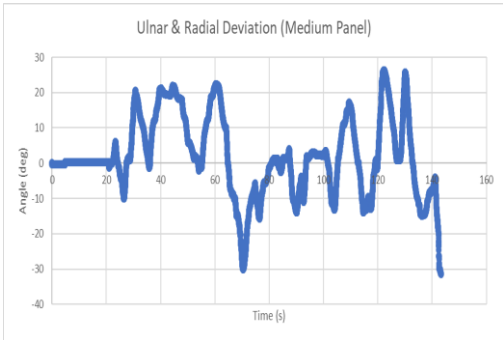


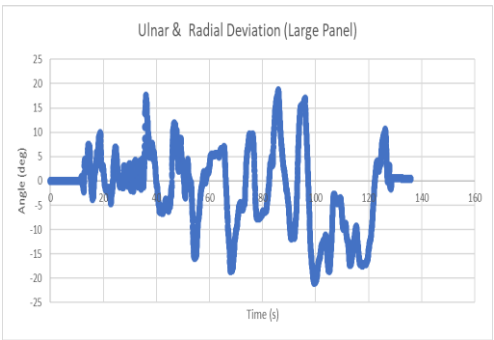
Figure 109: Sample radial/ulnar deviation data for the Brick-Busting game



(a)



(b)



(c)

Figure 110: More radial/ulnar deviation data samples for the Brick-Busting game; (a) Small Panel, (b) Medium Panel,

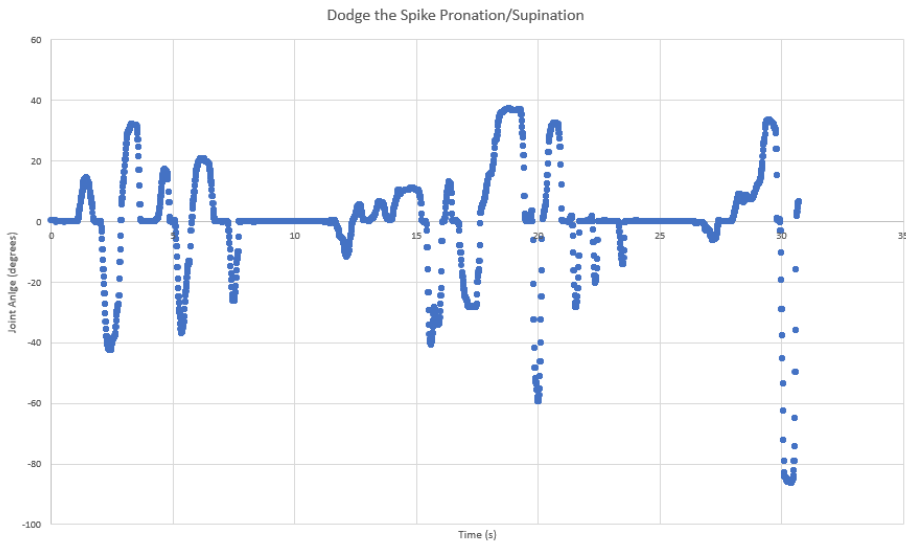


Figure 111: Sample pronation/supination data for the Dodge the Spike game

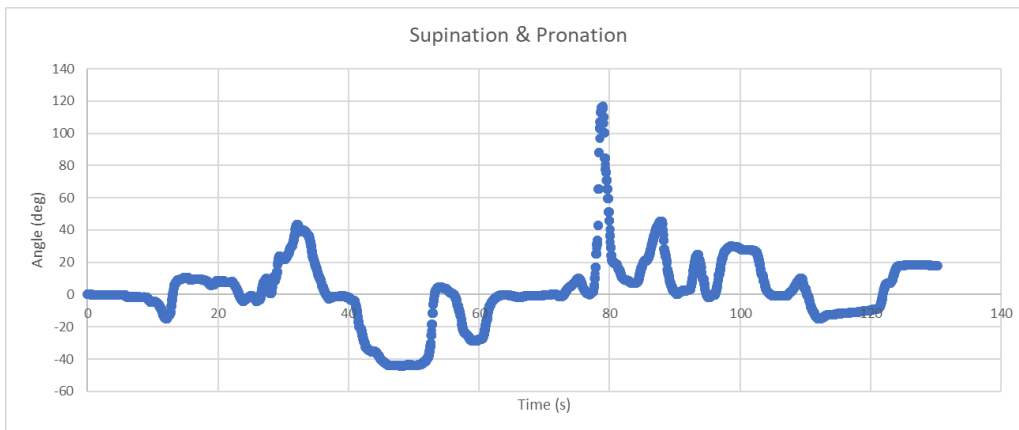


Figure 112: Sample pronation/supination data for the Tilting Maze game

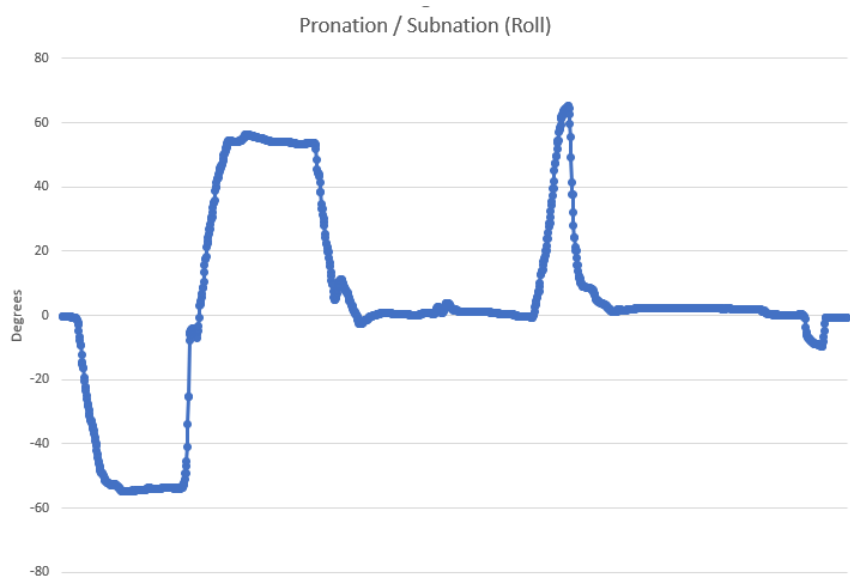


Figure 113: Sample pronation/supination data for the Skiing game

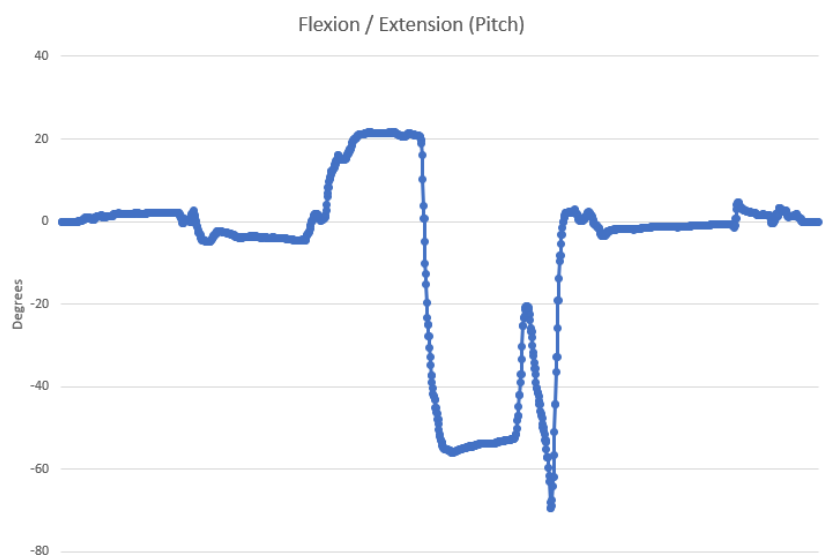


Figure 114: Sample flexion/extension data for the Skiing game

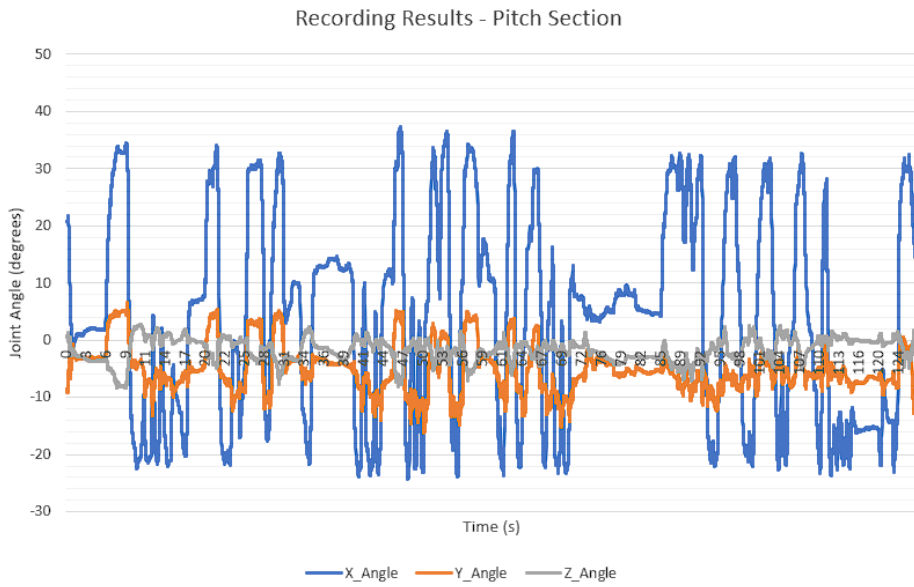


Figure 115: Sample data from Aeroplane flying game representing movement in pitch angles.

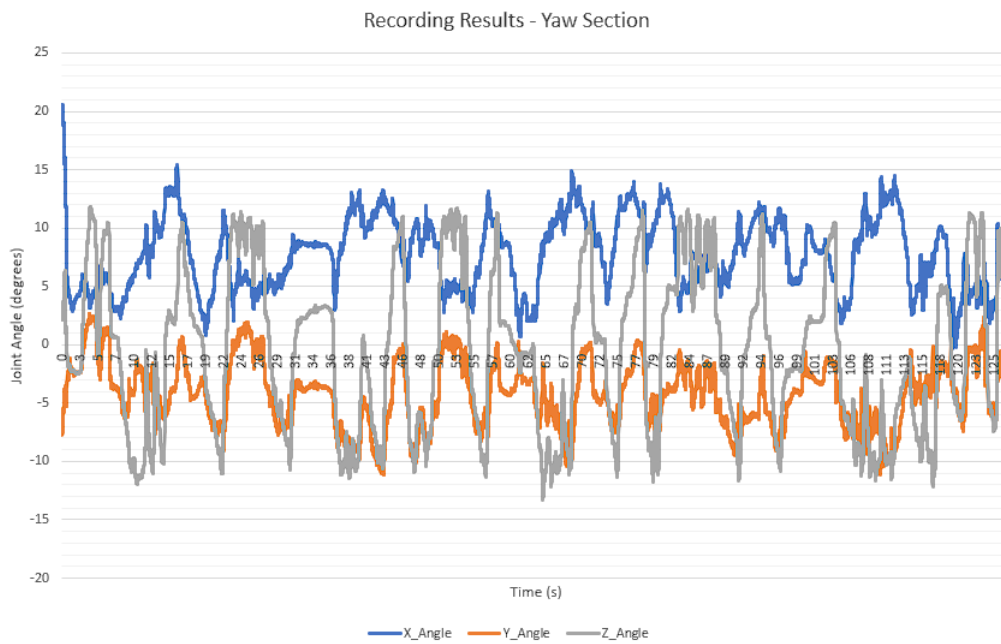


Figure 116: Sample data from Aeroplane flying game representing movement in Yaw angles.

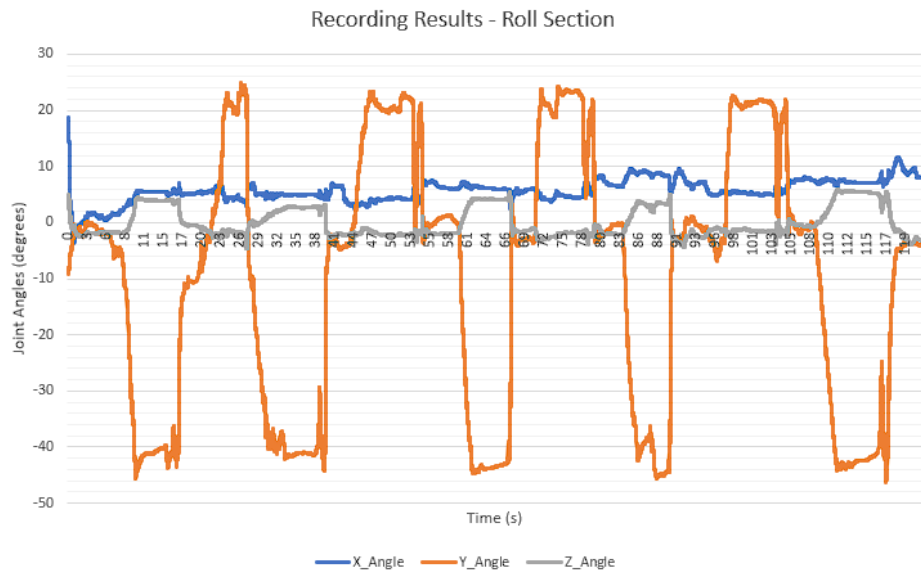


Figure 117: Sample data from Aeroplane flying game representing movement in Roll angles

APPENDIX C: EXAMPLES RESULTS OF LINEAR LDLJ AND ANGULAR SPARC FOR SMOOTHNESS MEASUREMENTS

This appendix provides more sample analysis results for the study run to measure angular and linear LDLJ for evaluation of smoothness measurements of the framework.

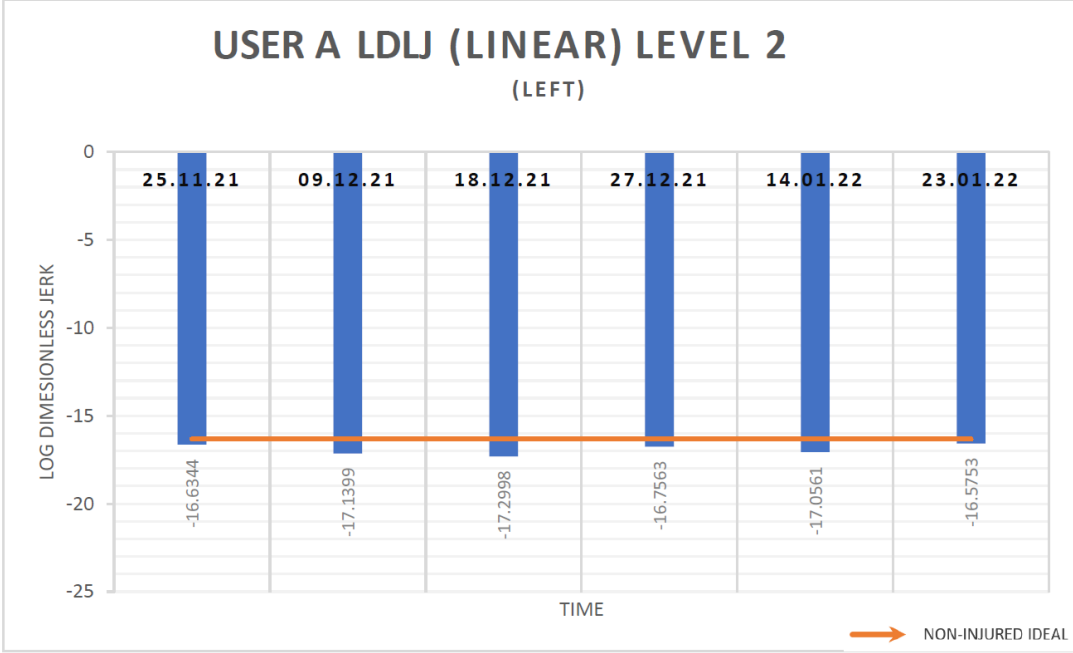


Figure 118: Linear LDLJ for User A

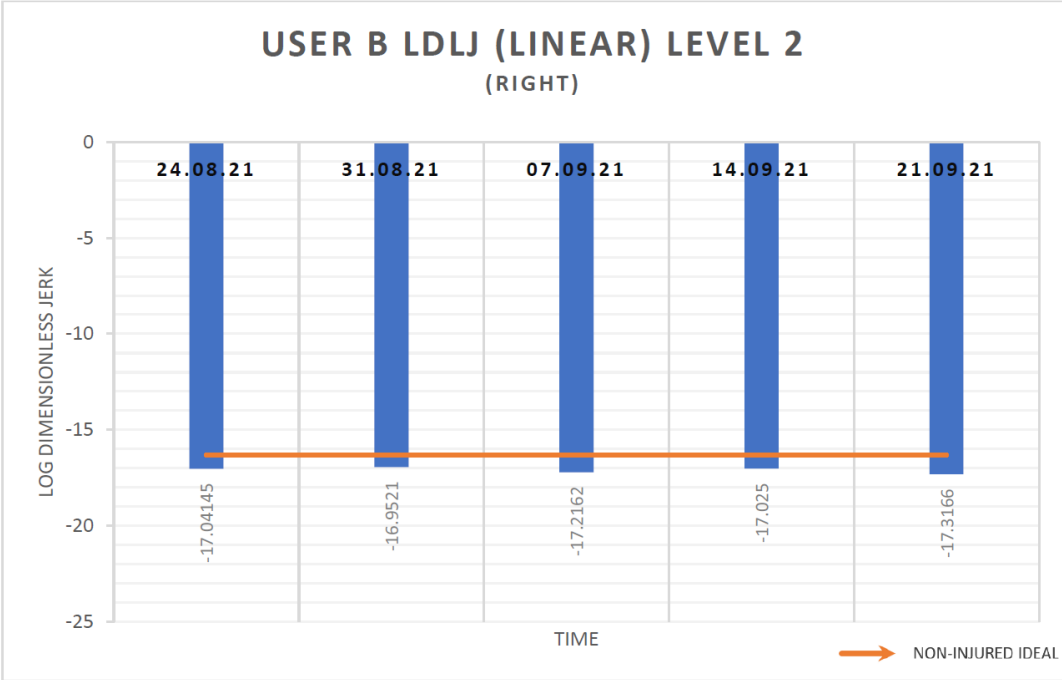


Figure 119: Linear LDLJ for User B

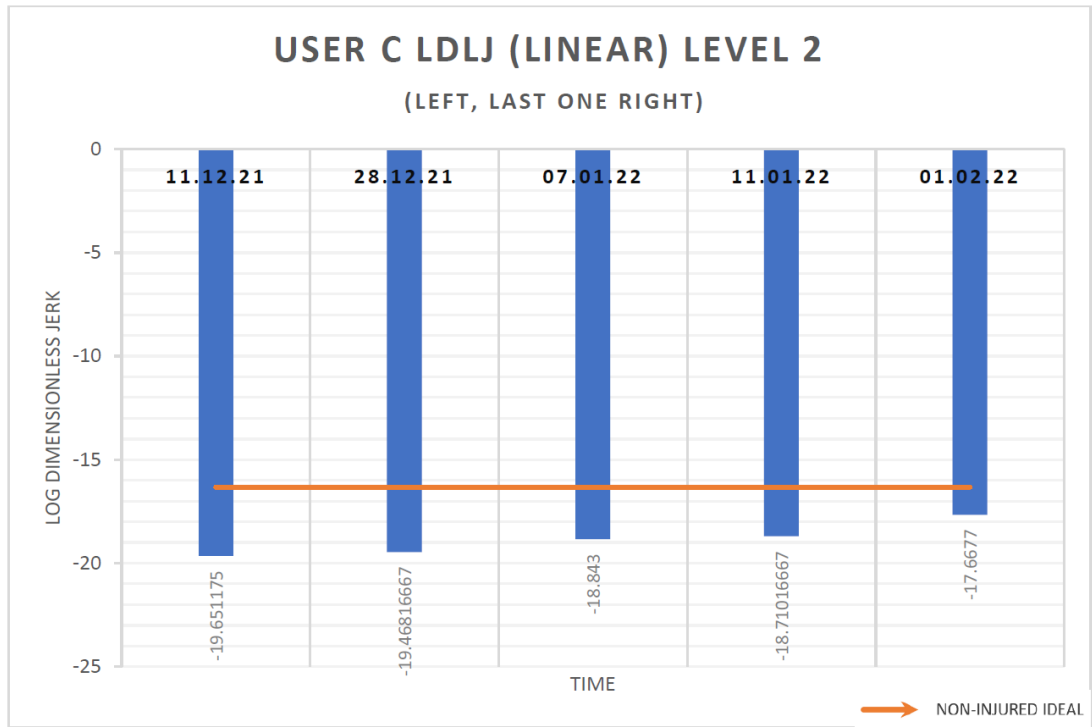


Figure 120: Linear LDLJ for User C

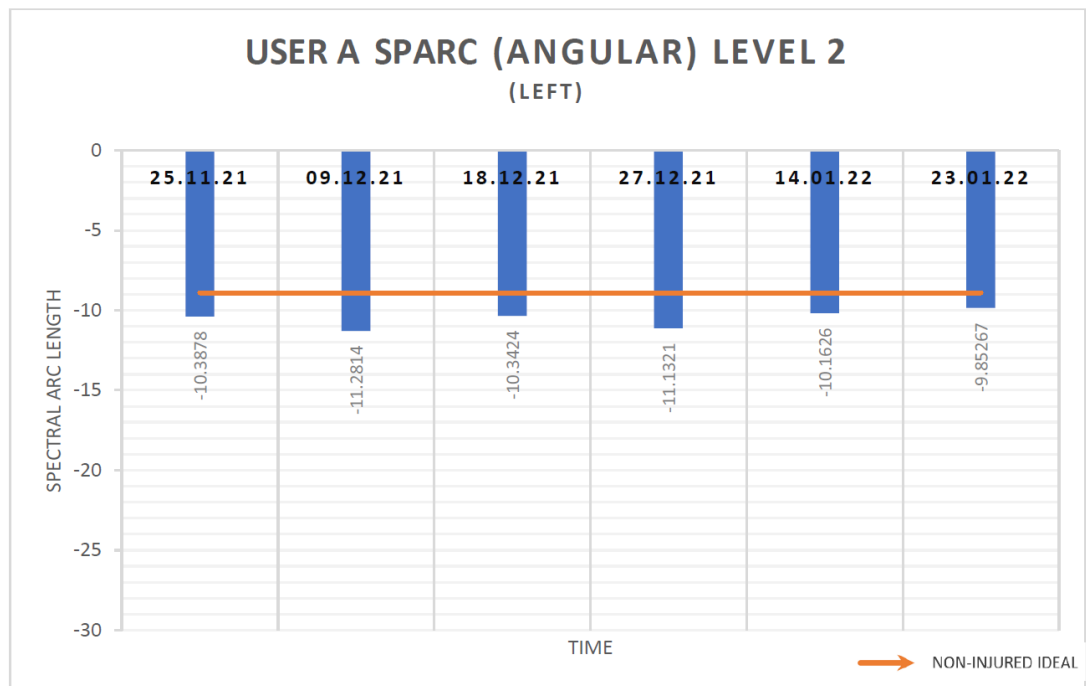


Figure 121: Angular SPARC for User A

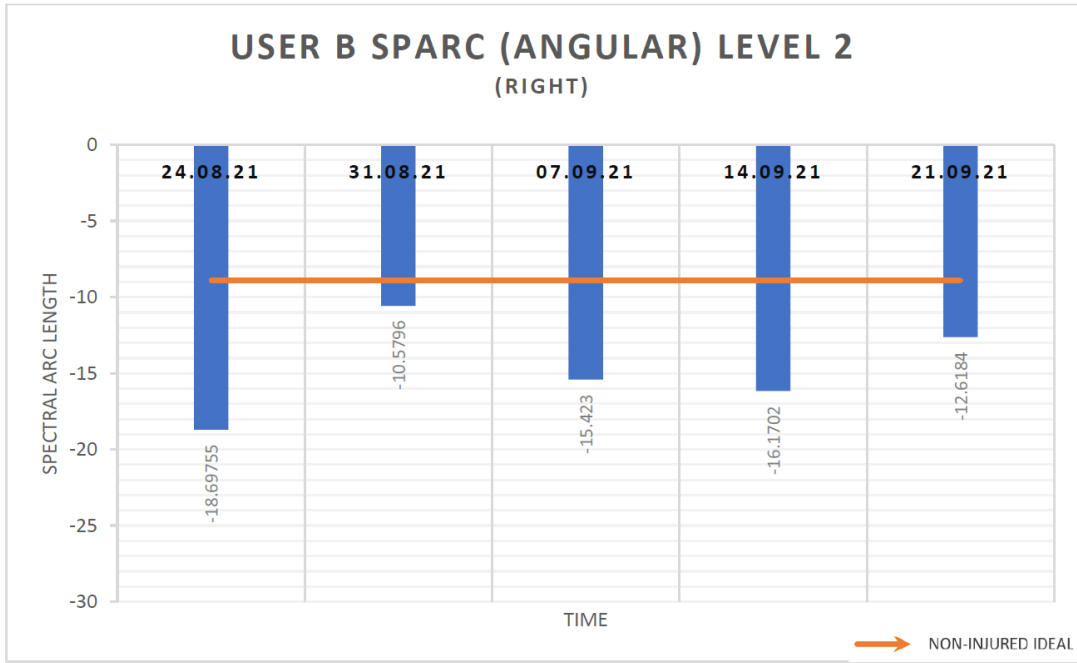


Figure 122: Angular SPARC for User B

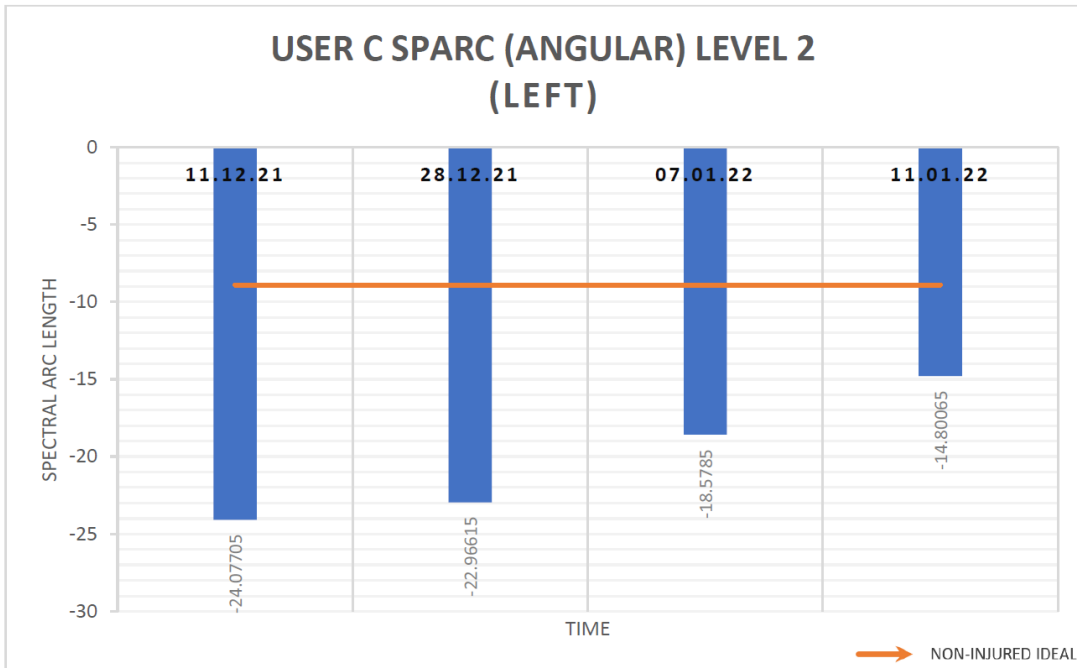


Figure 123: Angular SPARC for User C

APPENDIX D: FULL QUESTION LIST FOR FOCUS GROUP DISCUSSION

This appendix contains the full question list that was utilised for the focus group discussion.



Focus group questions:

1. Please introduce yourself and your occupation

2. Clinical implementation

- 2.1. How familiar are you with gamification of rehabilitation exercises and what is your opinion regarding its effects?
- 2.2. What is your initial impression of this framework? Please share any positive or negative feedback that comes to your mind.
- 2.3. Do you see a pathway for effectively incorporated this framework into clinical settings and patients' treatment plan? What are the roadblocks?

3. Usability questions

- 3.1. What do you think about the interaction method (i.e IMU sensors and the camera) with the framework?
- 3.2. Do you see any issues with connecting IMUs to people with disabilities?
- 3.3. Do you see any benefits of being able to remotely view the live exercise session or would a summary of activities with stats such as Max/Min Range of movements, number of repetitions achieved, etc be adequate
- 3.4. How often would you be interested in reviewing patient's engagement with the framework? (Per session or per week?) What are the benefits of the report?
- 3.5. Do you find the framework easy to navigate? What makes it easy/hard?
- 3.6. What do you think about the level of customization given the patients? i.e should there be more restrictions given to the patients when interacting with the framework.
- 3.7. Do you prefer to have a separate application for providing the baseline for the patients or would you rather have an all-in-one application?
- 3.8. Do you see any benefits in an AI system providing feedback to the user?

4. Final thoughts

- 4.1. Do you have any suggestions on the future development and improvement of the current framework