

Lebanese American University

The Design and Simulation of a Fuzzy
Logic-Controlled Upper-body Exoskeleton
for Lower-Limb Paraplegics Using Swing
Through Gait Ambulation

By

Jad Eid

A thesis submitted in the partial fulfillment of the
requirements for the degree of Master of Science in
Mechanical Engineering

School of Engineering
May 2022

THESIS APPROVAL FORM

Student Name: Jad Eid I.D. #: 201601895

Thesis Title: The Design and Simulation of a Fuzzy Logic-Controlled Upper-body Exoskeleton for Lowe

Program: Masters of Science in Mechanical Engineering

Department: Department of Industrial and Mechanical Engineering

School: _____

The undersigned certify that they have examined the final electronic copy of this thesis and approved it in Partial Fulfillment of the requirements for the degree of:

Master of Science in the major of Mechanical Engineering

Thesis Advisor's Name: Ali Ammouri

Signature: E-signature/Approval Attached. Date: 02 / 05 / 2022
Day Month Year

Committee Member's Name: Evan Fakhoury

Signature: _____ Date: 02 / 05 / 2022
Day Month Year

Committee Member's Name: Barbar Akle

Signature: _____ Date: 02 / 05 / 2022
Day Month Year



THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner) grants the Lebanese American University (LAU) the non-exclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic formats and in any medium, including but not limited to audio or video. You agree that LAU may, without changing the content, translate the submission to any medium or format for the purpose of preservation. You also agree that LAU may keep more than one copy of this submission for purposes of security, backup and preservation. You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant LAU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission. IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN LAU, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT. LAU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Name: Jad Eid

Signature: 

Date: 19 / 03 / 2022

Day

Month

Year

PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that:

1. I have read and understood LAU's Plagiarism Policy.
2. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
3. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: Jad Eid

Signature: 

Date: 19 / 03 / 2022

Day

Month

Year

The Design and Simulation of a Fuzzy Logic Controlled Upper-body Exoskeleton for Lower-Limb Paraplegics Using Swing Through Gait Ambulation

Jad Eid

Abstract

Currently, most of the research activities and the resulting commercial products offer motorized lower-limb exoskeletons that assist paraplegics in ambulating while standing up. However, there exist numerous gaiting techniques that enable the paralyzed to move using the upper extremities. This research aims for the development of an upper limb assistive technology that enables paraplegics to ambulate using a supported Swing Through Gait (STG) technique. The STG technique is known to require significant upper-body strength and superior balancing abilities during the swing phase. The challenge in the swing phase is due to the small contact area with the ground, represented by the ends of the crutches, and the large torques resulting from carrying the body and the swinging of the torso and legs. This article presents a control system that can be deployed onto an upper-limb rehabilitation exoskeleton that supports the STG ambulation technique. The controller is designed to recreate the STG using a rigid body dynamics model with a fuzzy logic controller guiding the motion of the shoulders and elbows. The controller is then calibrated and verified with experimental data measured from a spinal cord injured subject with complete thoracic lesion at T2 that relies on the STG pattern. The controller is further augmented by adding a failure detection and correction mode aimed at preventing falls and injuries in paraplegic users. The developed controller shows that it can control actuators in an exoskeleton to ambulate using the STG technique, detect falls/anomalies in the motion, and correct the falls to restore the intended trajectory. This work demonstrates that the proposed controller can be extended to an upper-limb exoskeleton to help support and rehabilitate paraplegic patients using the STG technique.

Keywords: Exoskeletons, Fuzzy Logic Control, Human-Robot Interaction, Rehabilitation Robotics, Simulation, Swing Through Gait

TABLE OF CONTENTS

Chapter	Page
I - Introduction	1
1.1 Literature Review on Rehabilitation & Exoskeletons.....	1
1.2 Importance of Work.....	2
II – Experimental Validation Procedure	4
2.1 Swing Through Gait Motion.....	5
2.2 Data Collection.....	6
2.3 Swing Through Gait Control & Simulation.....	6
2.4 Regular Motion Control System.....	7
2.5 Simulation Parameters & Methodology.....	8
2.6 Regular Motion Simulation & Results.....	9
III – Emergency Control & Simulation	10
3.1 Sensors.....	10
3.2 Fuzzy Logic Decision Making.....	10
3.3 Emergency Mode Control System.....	11
3.4 Emergency Control Simulation & Results.....	12
IV – Comparison of Subject Gait & Simulated Controller	16
4.1 Regular Gait Validation.....	16
4.1.1 Shoulder Motion Profile.....	16
4.1.2 Ankle Motion Profile.....	18
4.2 Gait Correction Analysis.....	19
V – Conclusion	20
Bibliography	21-23
Appendix	24-34

List of Figures

Figure 1 - The 4 key stages of the STG technique.	5
Figure 2 – Finite state machine that governs the operation of the exoskeleton. The four stages mimic the stages seen in the STG technique...	7
Figure 3 - Exploded view of simulated exoskeleton model.	8
Figure 4 - Side by side comparison of simulation and experimental models. .	9
Figure 5 - The Emergency control state machine that takes effect once the sensors detect a fall event. The state machine attempts to correct the trajectory of the body and avoid the fall.	11
Figure 6 - The application of an external disturbance to the body that triggered the emergency control response.	13
Figure 7 - Motion profile of body when disturbance is applied during regular STG motion. The controller detects a fall, prevents it, corrects the trajectory, and proceeds with the STG motion.....	14
Figure 8 - Variation of torso angle against time during initial angle disturbance. The fall prevention mechanism is enabled during the fall and prevents the fall from occurring. All initial disturbances eventually reach the same approximate final torso angle.	15
Figure 9 - Shoulder joint velocity comparison of simulated and experimental models with each of the stages visualized on the graph.....	17
Figure 10 - Ankle joint velocity comparison of simulated and experimental models with each of the stages visualized on the graph.....	18
Figure 11 - Sample LabVIEW project with the SoftMotion axes and the SolidWorks assembly.	28
Figure 12 - Sensor measurements and first FLC in LabVIEW.	30
Figure 13 - The user interface of the control system. This helped guide the user by displaying the state of the system and its behavior.....	34

List of Tables

Table I.....	6
Table II.....	9
Table III.....	12
Table IV	13
Table V	17
Table VI	18
Table VII	25
Table VIII	31

Chapter One

Introduction

Literature Review on Rehabilitation & Exoskeletons

Living with spinal cord injuries, or damage to the spinal cord nerves, often results in a reduction in mobility leading to a lower quality of life. Spinal cord injuries extend beyond immobility, eventually leading to muscle atrophy, skin complications, infections, and compromised cardiovascular systems [1]. Rehabilitative interventions help in recovering a certain degree of mobility and preventing further complications [2]–[4].

Although some lower-limb paraplegics rely on the *Swing-Through-Gait* (STG) ambulation technique to regain some form of mobility, most research focuses on the rehabilitation of lower-limb ambulation techniques using motorized exoskeletons [5]–[10]. One example exoskeleton operates by reading the wearer's movement through a set of EMG and joint angle sensors, inputting the data into the associated control system, and providing feedback to the rehabilitation center. The device provides assistive forces through knee and hip actuators according to the control system output after data acquisition [11]. Another study implements a similar exoskeleton system for unilateral lower limb disorders. The exoskeleton can analyze the motion of the healthy leg and use the data as control for the dyskinetic leg with a half gate-cycle delay. The control method utilizes torque measurements from the healthy leg to calculate and output assisted torque on the dyskinetic leg [12]. Another research project proposed a fuzzy controller for lower limb exoskeletons on the basis that an artificial neural network (ANN)-based classifier was susceptible to overtraining [13]. It is important to note that their work revolved around the sitting and standing actions rather than a complete gait cycle. While some work involves fuzzy logic control schemes, the majority involves artificial neural networks that learn the patterns and behaviors in the input data and output accordingly to the hip and knee actuators. Much of the literature neglects the use of upper-

limb ambulation techniques such as the STG, and focuses on human augmentation [14], [15] or lower-limb rehabilitation.

Importance of Work

Some paraplegics rely on their upper extremities for locomotion using ambulation techniques such as the STG, the target of this study. These movement patterns demand above average upper limb strength which may stress certain joints such as the shoulders, elbows, and wrists. Upper-limb ambulation techniques have a few key requirements and disadvantages:

- **Strength:** Requires above-average user strength in the serratus anterior, pectoralis major, latissimus dorsi, trapezius, deltoids, forearms, wrists, abdominals, biceps, and triceps.
- **Range of Motion:** Requires above average user knee and hip extension, scapular depression and abduction, shoulder flexion, elbow extension, and scapular elevation.
- **Injury:** Increases risk of injury in the shoulder, wrist, and elbow joints [16].

Given the dependency and difficulty of upper-limb ambulation techniques, this paper aims at initiating the development of a rehabilitation exoskeleton control system for the STG, one of several upper-limb ambulation technique.

As such, the objective of this paper is to develop a control system that can be deployed onto a rehabilitation exoskeleton to help enable the STG. The developed control system is validated by developing a dynamic rigid-body simulation that recreates the motion of a paralyzed person operating according to the decision making of the proposed controller. The model used in the simulation is calibrated and verified using experimental data.

This work introduces an experimental model that facilitates the validation procedure. It follows up by illustrating the controller design and

simulation parameters. It concludes by validating the controller against the experimental model.

Chapter Two

Experimental Validation Procedure

The parameters of the STG motion must be clearly defined to enable its implementation in a controller. Moreover, to develop and verify the STG controller, a series of experiments are performed with the help of a paraplegic subject that traverses using the STG. The motion profiles of certain points on our subject are tracked and compared to the motion profiles of similar points on the simulated model operating using the proposed controller.

Swing Through Gait Motion

The STG motion pattern is illustrated in Figure 1 and can be seen in the supplementary video titled *ExperimentSTG.mp4*. The STG technique starts with the subject in a balanced standing posture (S1 on Figure 1) that is followed by the movement of the crutches to a position ahead of the body. The subject then leans on the crutches (S2), extends the elbows, and protracts the scapulae to ensure that the feet leave the ground. The shoulders act as a pivot point during this stage so that the body can swing through the crutches (S3). Finally, the subject retracts the elbows and scapulae as soon as the heels contact the ground to regain the primary balanced position or to repeat the cycle (S4 - S1).

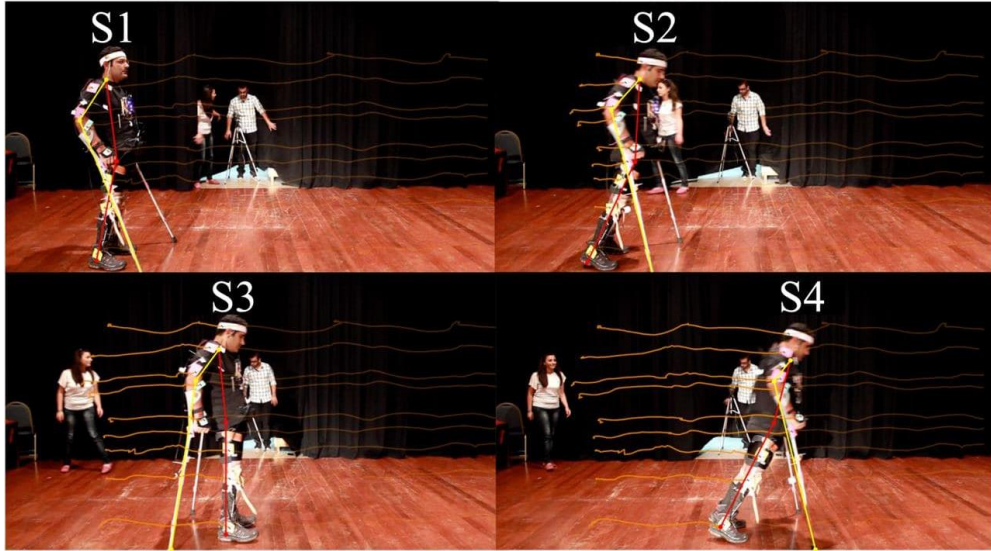


Figure 1 - The 4 key stages of the STG technique.

Thus, STG can be modeled through a set of four sequential stages (S1-S4) that involve the use of the shoulders, arms, crutches, and a rigid body. This sequence of steps can be transferred to the simulations discussed in the control sections of this paper.

Data Collection

The paraplegic subject, shown in Figure 1, is asked to repeat STG cycles over 10m of flat terrain. The subject is a T2-T3 lower-limb paraplegic that regularly ambulates using the STG. The subject's motion is video recorded using a Canon T2i camera positioned perpendicular to the plane of motion at an elevation of 1.5m from the ground. The videos are post-processed using Kinovea, an open-source sports analysis software, to track the motion of the markers placed on the ankle and shoulder joints, areas that are important to assess the STG motion. The collected data of the experimental model is compared to that of the simulated model operating under the developed controller.

Swing Through Gait Control & Simulation

The primary objective of this paper is to develop a controller that can mimic the STG motion when actuating the motors of an upper-limb exoskeleton. This controller is designed to operate in two modes: Regular and emergency control mode. The regular motion control system reported in this section is designed to replicate the STG. The emergency control mode provides a safety net for

users and is discussed in the next section. The controller for the regular mode consists of a control paradigm that actuates the simulated shoulder and elbow joints, as discussed in the next subsection.

Regular Motion Control System

The STG controller consists of a finite state machine (FSM), which is a type of control scheme that can describe the operation of the system. It is characterized by its ability to only be in one state at a given time and can be controlled to transition between different states. The states are programmed to actuate the motors of the simulated model in the case of the paper's proposed controller. The functionality of the state machine can be extended to a physical exoskeleton prototype by implementing control scheme to physical motors in the shoulder and elbow joints.

Regular Motion States	
State	Function
Initialization State	Software Initialization
Stage 1	Balanced Start
Stage 2	Advance Crutches
Stage 3	Swing Legs
Stage 4	Restart
Emergency Control SM	Emergency Mode Activation

An FSM was chosen as the main control scheme as it provides a method to define and transition between unique states that can mimic the four defined stages of the STG (S1-S4). The behavior of the FSM is illustrated using the diagram in Figure 2 where each state represents a certain stage of the STG motion, as shown in *Table 1*.

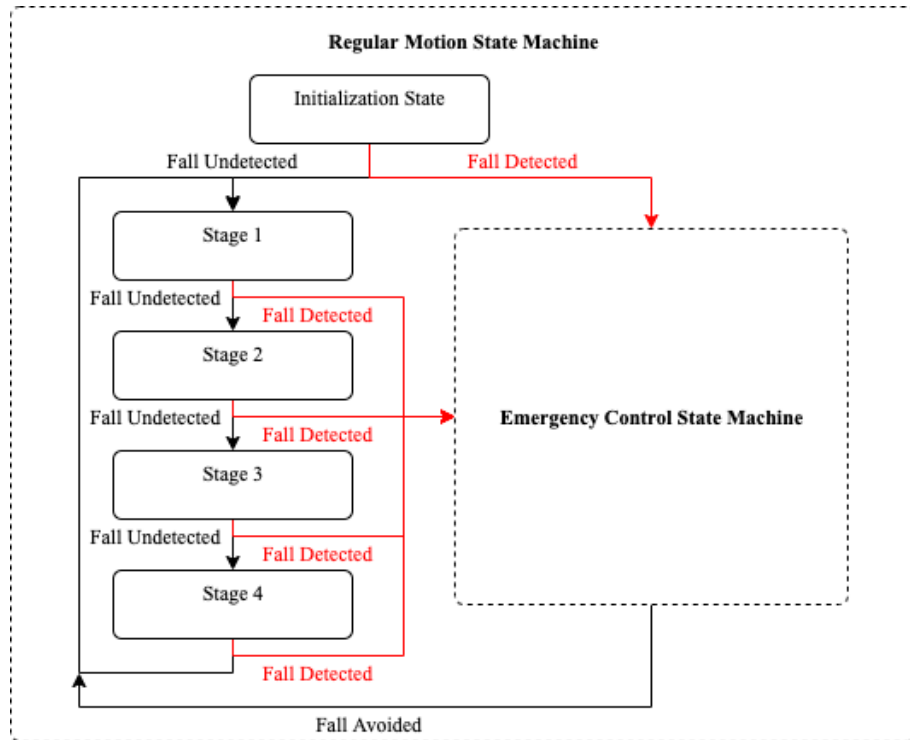


Figure 2 – Finite state machine that governs the operation of the exoskeleton. The four stages mimic the stages seen in the STG technique.

The FSM is programmed to transition between the different stages of the STG given that no disturbances are introduced into the trajectory. The last stage in *Table 1*, the Emergency Control SM, is added to detect and avoid falls. This emergency mode sub-state machine is discussed in the next section.

Simulation Parameters & Methodology

The paper's proposed controller is validated against the experimental data to verify its functionality and performance. SOLIDWORKS, a computer aided design and engineering software, is interfaced with LabVIEW, a graphical programming language, to code the control algorithm and test the performance of the controller using a simulated exoskeleton model. The purpose of the interface is to reliably test the controller during development and fine-tune it to represent the STG motion defined in the literature. The LabVIEW SoftMotion module, an interface module for LabVIEW and third-party software, supports this task by providing a method to interface SOLIDWORKS and LabVIEW, control the actuators, and read sensor data during the motion study. A 3D model that consists of four primary components is developed to simulate the motion. The components include: The torso and lower body as a single rigid structure, the arms, the crutches, and the ground. An exploded view of the model is shown in Figure 3.

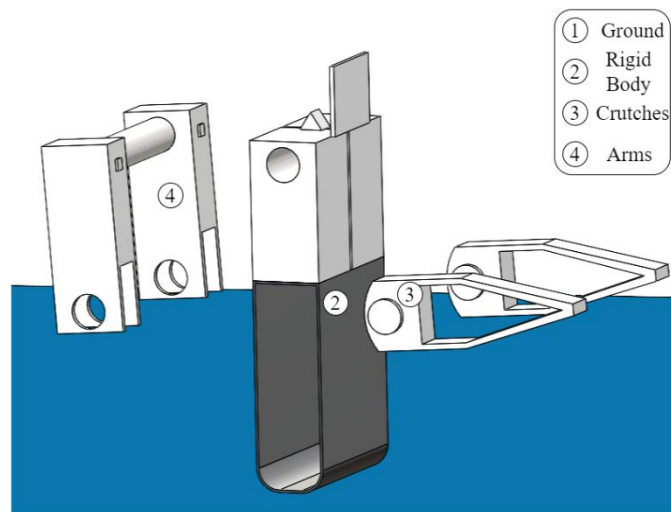


Figure 3 - Exploded view of simulated exoskeleton model.

The simulated body seen in Figure 3 represents the anthropometry of the subject of the experimental model to ensure an accurate comparison. As such, the exoskeleton is modeled according to the data in *Table II*.

Paraplegic Subject Anthropometric Data	
Dimension	Value
Bodyweight	105 Kg
Height	1.8 m
Arm Length	0.6 m
Crutch Length	1.0 m

Actuators were placed on the shoulder and elbow joints to emulate the flexion and extension of the shoulder and triceps muscles. The joints were designed as pin joints as they provide the necessary motion required for the STG. These motors rotate the arm ahead of and behind the body and extend/retract the crutches.

Regular Motion Simulation & Results

The SOLIDWORKS Motion Toolbox, a Multiphysics simulation tool, is used to simulate the designed model ambulating using the STG through the decision making of the state machine. The motion of the ankle and shoulder joints of the simulated model are recorded to be compared to the experimental data. The controller is observed to operate reliably in the regular mode, given that no external disturbances are introduced into the system. The resulting motion is displayed in Figure 4 and can also be seen in the supplementary video titled *RegularMotion.mp4*.



Figure 4 - Side by side comparison of simulation and experimental models.

Chapter Three

Emergency Control & Simulation

Given the complexity of the movement, paraplegic rehabilitation patients are likely to develop the STG ambulation technique through repeated practice. However, novice STG users are generally susceptible to failure or falling during training. This can lead to severe injuries that could demotivate the patients. As such, the paper's proposed controller is designed to provide a safety net by transitioning to an emergency mode after detecting a fall/failure in the motion. This emergency mode determines the current condition of the system using a set of sensors and actuates the motors of the exoskeleton to place the arms in a position that prevents the fall.

Sensors

The decision making is dependent on a set of sensors that analyze the condition of the system. The fuzzy logic decision-making algorithm in the proposed controller, used to detect failures in the motion, requires a set of inputs from motion sensors on specific parts of the body to produce a useable output. This paper's proposed controller is dependent on feedback from three different variables (torso angle relative to the perpendicular plane to the ground, torso angular velocity, torso angular acceleration) and this feedback is analyzed to determine if the state machine should transition from the regular motion to the emergency mode.

Fuzzy Logic Decision Making

The operation of this paper's controller is dependent on the condition of the system as read by the set of sensors. These torso angle, velocity, and acceleration values were inputted into a decision-making algorithm, a fuzzy logic controller (FLC), that decides between the regular STG and the emergency control mode. Fuzzy logic has several advantages over other conventional decision-making algorithms: The configuration is more accessible, it is easier to develop and tune, and it does not require large datasets to train [21]. As such, a multiple input single output FLC is chosen to

detect fails in the trajectory. The input into the FLC is a concatenation of the three signals read by the sensors. The output of the FLC dictates if the trajectory is in an error state or not. If a fall is detected, the fuzzy logic triggers the transition into the emergency control sub-state machine. The functionality of the sub-state machine is discussed in the next subsection.

Emergency Mode Control System

Given the complexity of the movement, a paraplegic subject utilizing the STG often runs the risk of falling. This could be due to muscle fatigue during continuous STG use, obstacles in the ambulation space, and other environmental conditions that can obstruct the STG. In result, a safety mechanism is integrated into the controller as a safety net for patients that use an exoskeleton operating under the proposed controller. This could promote a sense of safety in future exoskeleton wearer to allow them to continue to pursue the STG motion. Once a failure in the motion is detected, the FLC triggers the FSM to transition into the emergency control mode. The emergency control mode is a sub-state machine within the regular motion FSM shown in Figure 2. The FSM diagram of the emergency mode is shown in Figure 5 and the individual states are explained in *Table III*.

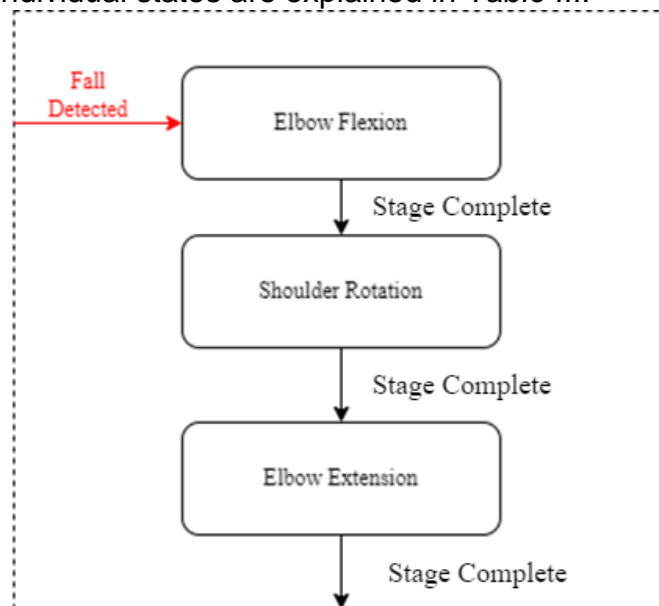


Figure 5 - The Emergency control state machine that takes effect once the sensors detect a fall event. The state machine attempts to correct the trajectory of the body and avoid the fall.

<i>Table III</i>	
Emergency Control States	
State	Function
Elbow Flexion	The elbows retract and create clearance between the crutches and the ground to allow shoulder rotation.
Shoulder Rotation	The shoulders rotate to a position that avoid a fall.
Elbow Extension	The elbows are extended to prevent the fall once the shoulders reach the intended position.

In summary, the emergency control mode analyzes the direction of the fall and prevents it by placing the arms in a position that catches the body.

Emergency Control Simulation & Results

To simulate a failure during the STG motion, various disturbances in the form of forces and initial conditions are systematically introduced into the physics simulation to test the safety mechanism's functionality in this paper's proposed controller. This controller is shown to recover from a step-function force application of 150N that is applied to the static body as shown in Figure 6. This force magnitude is chosen as an initial assessment and controller demonstration as it was the minimum required force to tip over the body in the simulation space. Each of the events of the force application simulation are explained in *Table IV*. The motion can also be seen in the supplementary file titled *FLCSimulation.mp4*.

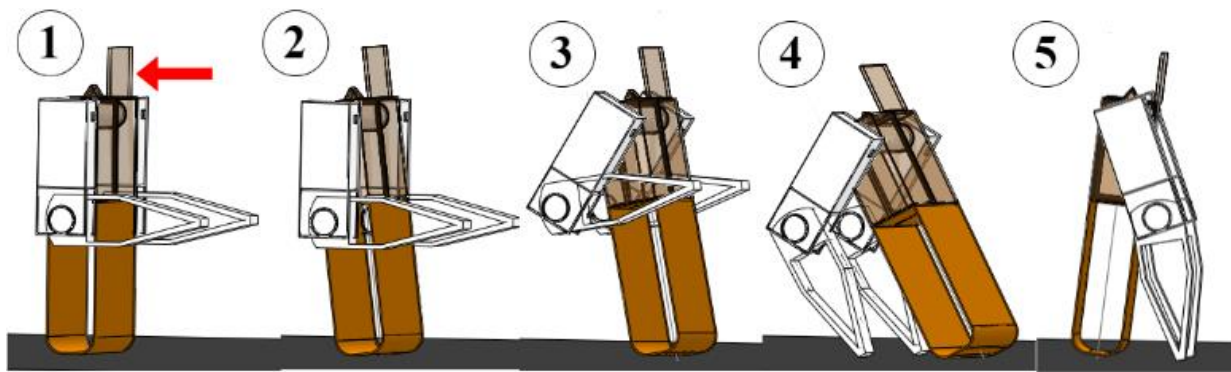


Figure 6 - The application of an external disturbance to the body that triggered the emergency control response.

Table IV

Emergency Control

Event	Function
1	A force application interrupts regular motion.
2	A fall is detected by the FLC, and the emergency mode is triggered.
3	The elbows retract to allow for shoulder rotation.
4	The shoulders rotate to prevent the fall.
5	The elbows extend to return to regular motion.

The controller is also shown to recover from a similar force disturbance during the regular STG motion. The motion profile of the body during the application of this disturbance is shown in Figure 7.

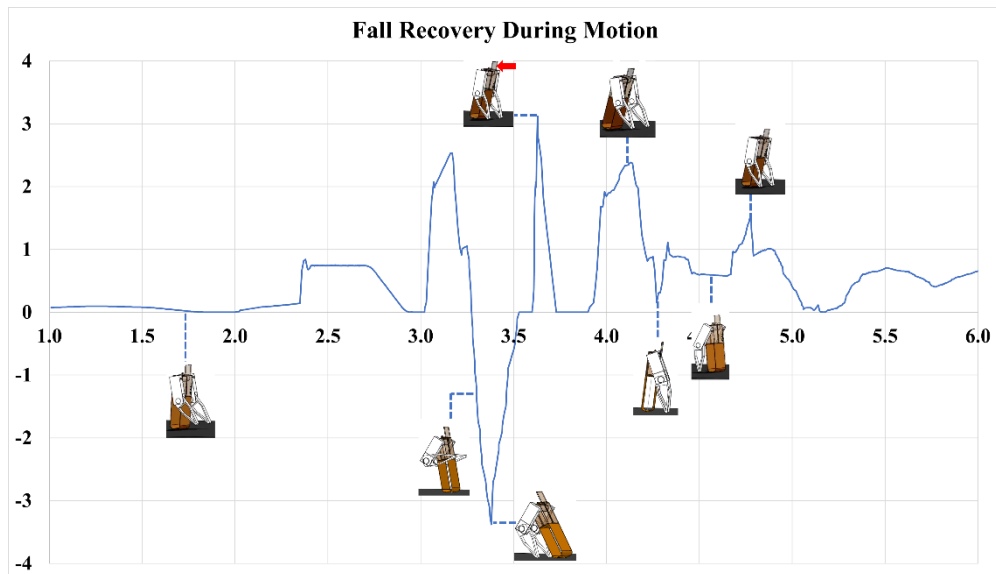


Figure 7 - Motion profile of body when disturbance is applied during regular STG motion. The controller detects a fall, prevents it, corrects the trajectory, and proceeds with the STG motion.

The controller is further tested for a range of initial positions that triggered the emergency control mechanism when the body is stationary. These initial positions varied by the torso angle relative to the perpendicular plane to the ground. The range of values begins at 0° (upright torso position) and is incremented to reach a torso angle that the emergency control mechanism could not recover from. This is done to determine the maximum torso angle, or the furthest tilt from the upright position, that the emergency mechanism could successfully recover from. The controller is shown to recover from falls at torso angles of $\theta_{1i} \geq -10^\circ$ relative to the perpendicular plane to the ground, as measured by the sensors in the simulation. Figure 8 visualizes the variation of torso angle against time during and after fall recovery. The events of the simulation are identical to those seen in *Table IV*, however, the first event can be replaced by an initial torso angle rather than a force application.

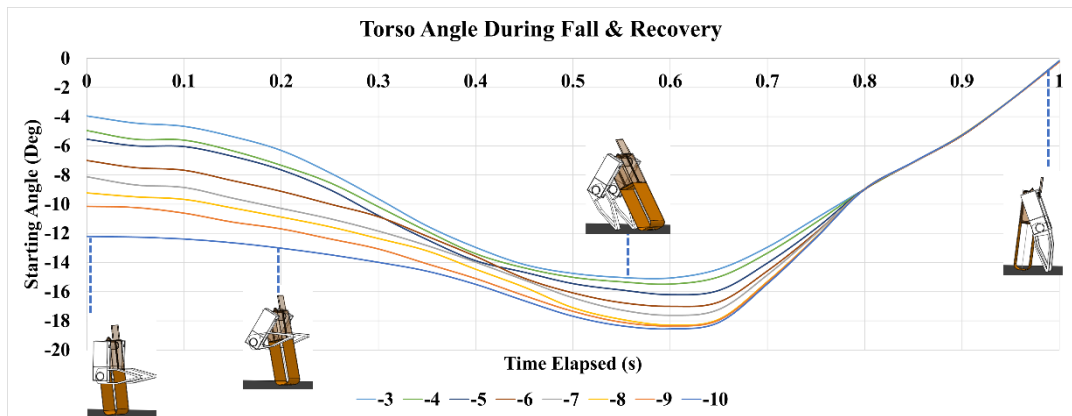


Figure 8 - Variation of torso angle against time during initial angle disturbance. The fall prevention mechanism is enabled during the fall and prevents the fall from occurring. All initial disturbances eventually reach the same approximate final torso angle.

Chapter Four

Comparison of Subject Gait & Simulated Controller

The controller is verified by comparing the recorded motion profile of the simulated body to that of the experimental model. The verification is completed in two stages: Regular gait validation and gait correction analysis.

Regular Gait Validation

The results of the subject's motion profile are compared to those of the regular operation of the simulated body using the proposed controller. The shoulder and ankle points, the two key points of the STG motion, are tracked to analyze the velocity profiles of the exoskeleton during the STG technique. The velocity profiles of the experimental and simulated models are compared on plots where the y-axis represents velocity of the tracked point, and the x-axis represents the elapsed time. The motion profiles similarly show repeating patterns that represent the cycling of the STG with some variations in velocity and timing.

Shoulder Motion Profile

The first notable point is the shoulder joint as it is a critical pivot point during the STG. The velocity profile of the shoulder joint on the simulated body is recorded and compared to the velocity profile of the subject's shoulder joint. The velocity profile of the models is visualized in Figure 9 and explained in *Table V*.

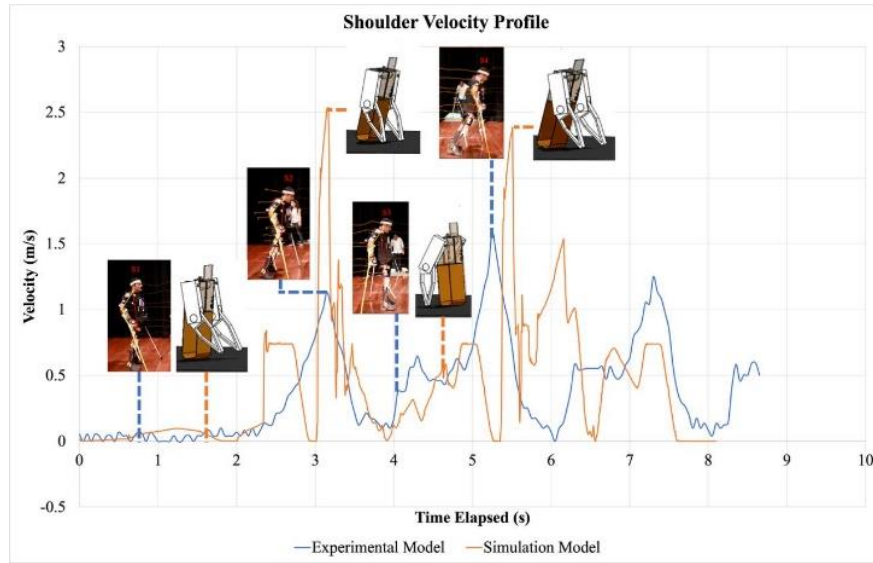


Figure 9 - Shoulder joint velocity comparison of simulated and experimental models with each of the stages visualized on the graph.

Table V

Event	Explanation
Stage 1	Start of motion, shoulder is stationary.
Stage 2	Shoulders move forward during the lean onto the crutches ahead of the body.
Stage 3	Shoulders are stationary while legs swing forward.
Stage 4	Shoulders move forward again during the lean to restart the cycle.

The cyclic behavior of the controller-operated body matches that of the experimental model. There is a discrepancy between the local velocity maxima of the two profiles which may lead to a mismatch in ambulation rate. This could be attributed to the high latency between the simulation and control software.

However, the repeated cyclical behavior verifies the performance of the proposed control system.

Ankle Motion Profile

The second critical point, the ankle joint, is also tracked and compared to further validate the performance of the controller. The data is visualized in Figure 10 and each stage is explained in *Table VI*.

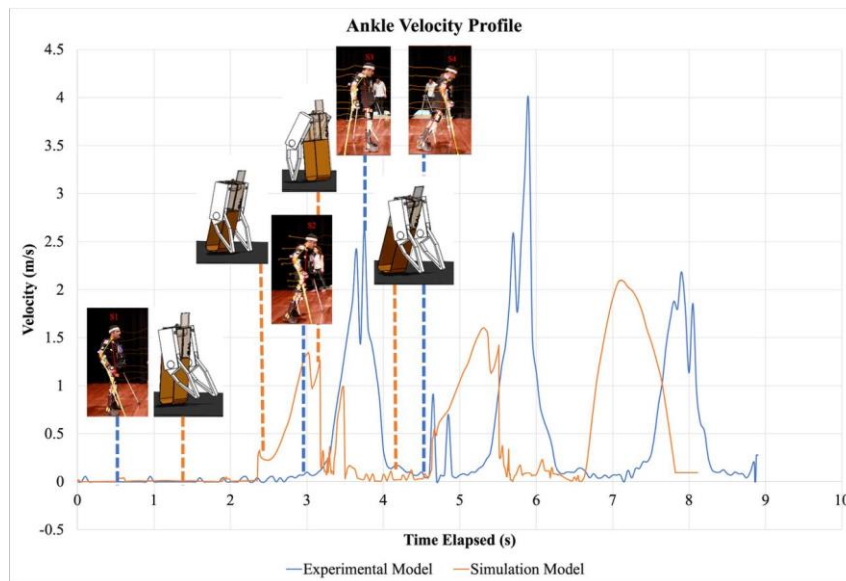


Figure 10 - Ankle joint velocity comparison of simulated and experimental models with each of the stages visualized on the graph.

Table VI

Event	Explanation
Stage 1	Start of motion, ankle is stationary.
Stage 2	Ankles remain stationary during the lean stage.
Stage 3	Ankles swing forward while shoulders remain stationary.
Stage 4	Ankles remain stationary as the cycle starts again.

A similar cyclical pattern is observed in the motion profile of the ankle joint. Moreover, there is a delay between the two motion profiles. Regardless, the

cyclical behavior of the ankle joint's motion profile on the simulated body matches that of the experimental model and further validates the performance of the controller.

Gait Correction Analysis

The safety mechanism ensures that future users of an upper-limb STG rehabilitation exoskeleton operating through the controller are protected against potential falls, to a certain extent. The extent of the emergency mode's efficacy can be explained analytically: A force disturbance is initially introduced to tip over the simulated body and trigger the emergency control mode when the body is stationary and during STG motion. The emergency mode's reaction is recorded as a proof of functionality. The paper's proposed controller detects this disturbance as a failure in the STG and triggers the transition to the emergency mode. This rotates the arms to catch the body and prevent the fall. The paper's controller is further tested for a range of starting positions that vary by torso angle relative to the perpendicular plane to the ground. The body starts in an upright position and is tilted in increments of 1° . The simulation is observed to determine if the emergency control mode remains functional at each torso angle. The emergency control mode begins to fail to prevent falls at $\theta_{1i} < -10^\circ$. Although failures are detected for larger torso angles, the emergency control sub-state machine is incapable of quickly rotating the arms to prevent a fall before it occurs.

Chapter Five

Conclusion & Evaluation

The objective of this paper was to develop and validate a control system that can be integrated into an upper body exoskeleton to help paraplegic subject rehabilitation using the Swing Through Gait. The exoskeleton controller proposed in this paper is shown to mimic the STG and prevent trajectory failures. The comparison of the experimental and simulated models shows that the performance of the controller is comparable to that of a paraplegic subject that relies on the STG. The motion profiles of the experimental and simulated models, shown in Figure 9 and Figure 10, display a similar cyclical behavior. The controller also provides a safety net that prevents falls to avoid injuries. This functionality is demonstrated through the controller's ability to recover from falls during the STG motion and when the body was stationary. The controller is shown to recover from falls/trajectory failures caused by force disturbances and through initial torso angle conditions; these results are summarized in Figure 7 and Figure 8. The results of this article could serve as a starting point for future research in the field of underactuated mechanical systems, particularly on the topic of upper-limb ambulation rehabilitation. Future work should focus on the implementation and testing of the STG controller into a physical upper-limb ambulation exoskeleton model.

Bibliography

- [1] B. S. Armour, E. A. Courtney-Long, M. H. Fox, H. Fredine, and A. Cahill, "Prevalence and Causes of Paralysis—United States, 2013," *American Journal of Public Health*, vol. 106, no. 10, p. 1855, Oct. 2016, doi: 10.2105/AJPH.2016.303270.
- [2] N. Sezer, S. Akkuş, and F. G. Uğurlu, "Chronic complications of spinal cord injury," *World Journal of Orthopedics*, vol. 6, no. 1, p. 24, 2015, doi: 10.5312/WJO.V6.I1.24.
- [3] R. Nasiri, A. Ahmadi, and M. N. Ahmadabadi, "Reducing the energy cost of human running using an unpowered exoskeleton," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 10, pp. 2026–2032, Oct. 2018, doi: 10.1109/TNSRE.2018.2872889.
- [4] F. Patané, S. Rossi, F. del Sette, J. Taborri, and P. Cappa, "WAKE-Up Exoskeleton to Assist Children With Cerebral Palsy: Design and Preliminary Evaluation in Level Walking," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 7, pp. 906–916, Jul. 2017, doi: 10.1109/TNSRE.2017.2651404.
- [5] B. Ostrach and R. Riemer, "Simulation of a Passive Knee Exoskeleton for Vertical Jump Using Optimal Control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2859–2868, Dec. 2020, doi: 10.1109/TNSRE.2020.3039923.
- [6] G. Orekhov, Y. Fang, J. Luque, and Z. F. Lerner, "Ankle Exoskeleton Assistance Can Improve Over-Ground Walking Economy in Individuals with Cerebral Palsy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 2, pp. 461–467, Feb. 2020, doi: 10.1109/TNSRE.2020.2965029.
- [7] S. Wang *et al.*, "Design and Control of the MINDWALKER Exoskeleton," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 2, pp. 277–286, 2015, doi: 10.1109/TNSRE.2014.2365697.
- [8] T. Vouga *et al.*, "EXiO - A Brain-Controlled Lower Limb Exoskeleton for Rhesus Macaques," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 1–10, Jan. 2020, doi: 10.1109/TNSRE.2020.2965029.

- Rehabilitation Engineering*, vol. 25, no. 2, pp. 131–141, Feb. 2017, doi: 10.1109/TNSRE.2017.2659654.
- [9] G. Scivoletto, F. Tamburella, L. Laurenza, M. Torre, and M. Molinari, “Who is going to walk? A review of the factors influencing walking recovery after spinal cord injury,” *Frontiers in Human Neuroscience*, vol. 8, no. MAR, Mar. 2014, doi: 10.3389/FNHUM.2014.00141.
- [10] “Ambulation Over Even Surfaces.”
<https://calder.med.miami.edu/providers/PHYSICAL/ameven.html>
 (accessed Jul. 13, 2021).
- [11] W. Ma, X. Zhang, and G. Yin, “Design on intelligent perception system for lower limb rehabilitation exoskeleton robot,” *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2016*, pp. 587–592, 2016, doi: 10.1109/URAI.2016.7625785.
- [12] C. Zhang, G. Liu, C. Li, J. Zhao, H. Yu, and Y. Zhu, “Development of a lower limb rehabilitation exoskeleton based on real-time gait detection and gait tracking,” *Advances in Mechanical Engineering*, vol. 8, no. 1, pp. 1–9, 2016, doi: 10.1177/1687814015627982.
- [13] S. M. Taslim Reza, N. Ahmad, I. A. Choudhury, and R. A. Raja Ghazilla, “A fuzzy controller for lower limb exoskeletons during sit-to-stand and stand-to-sit movement using wearable sensors,” *Sensors (Switzerland)*, vol. 14, no. 3, pp. 4342–4363, 2014, doi: 10.3390/s140304342.
- [14] G. Lv, H. Zhu, and R. Gregg, “On the design and control of highly backdrivable lower-limb exoskeletons,” *IEEE Control Systems Magazine*, vol. 38, no. 6, pp. 88–113, 2018, [Online]. Available: under review
- [15] W. Huo, S. Mohammed, Y. Amirat, and K. Kong, “Fast Gait Mode Detection and Assistive Torque Control of an Exoskeletal Robotic Orthosis for Walking Assistance,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1035–1052, 2018, doi: 10.1109/TRO.2018.2830367.
- [16] R. J. Farris, H. A. Quintero, S. A. Murray, K. H. Ha, C. Hartigan, and M. Goldfarb, “A preliminary assessment of legged mobility provided by a lower limb exoskeleton for persons with paraplegia,” *IEEE*

- Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 3, pp. 482–490, 2014, doi: 10.1109/TNSRE.2013.2268320.
- [17] S. Song and S. H. Collins, “Optimizing Exoskeleton Assistance for Faster Self-Selected Walking,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 786–795, 2021, doi: 10.1109/TNSRE.2021.3074154.
- [18] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, “Robot assisted gait training with active leg exoskeleton (ALEX),” *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2008*, vol. 17, no. 1, pp. 653–658, 2008, doi: 10.1109/BIOROB.2008.4762885.
- [19] D. S. Yoo, “Modeling and Control of an Acrobot Using MATLAB and Simulink,” *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 9, no. 6, pp. 1031–1034, 2015.
- [20] D. T. Pham, A. H. Darwish, E. E. Eldukhri, S. Otri, A. H. Darwish, and S. Otri, “Using the Bees Algorithm to tune a fuzzy logic controller for a robot gymnast Optimisation of control algorithms for complex dynamic systems View project Micro milling issues View project Using the Bees Algorithm to tune a fuzzy logic controller for a rob,” no. September 2015, 2007.
- [21] A. I. Al-Odienat and A. A. Al-Lawama, “The Advantages of PID Fuzzy Controllers Over The Conventional Types,” *American Journal of Applied Sciences*, vol. 5, no. 6, pp. 653–658, 2008.

Appendix

The appendix section aims to expand on the procedure used to recreate the repeatable test environment in the simulation and the control software that allowed for the analysis of the exoskeleton behavior. This section discloses the necessary information needed to replicate the study. This includes all the necessary simulation parameters in SolidWorks, LabVIEW, and SoftMotion.

Simulation Setup

As mentioned in a previous section, the exoskeleton model and motion study were setup on SolidWorks.

Assembly

The assembly consisted of four main components:

1. Crutches: The exoskeleton's support (attached to the elbows); creates contact with the ground. The joint between the arms and the crutches is actuated using a SolidWorks rotational motor.
2. Arms: The arms are connected to the body through the shoulder joint and are connected to the crutches at the elbow. The shoulder joint is actuated using a SolidWorks rotational motor.
3. Body: The exoskeleton body acts as the body of the wearer along with the exoskeleton. It is connected to the arms through the shoulder joint and contacts the ground throughout the motion.
4. Ground: The ground is the simulation space where the simulation takes place. The crutches and the body create contact with the ground.

Motion Study

There were several notable simulation parameters for the motion study. The parameters are shown in *Table VII*.

Table VII

Simulation Parameters	
Parameter	Value
Gravity	$g_z = 9.81 \text{ ms}^{-1}$
Contact Definitions	Crutches, Body, Arms, Ground
Actuators	Shoulder Joint, Elbow Joint Motors set to <i>distance</i> in “edit feature”
Sensors	Torso Angle relative to Ground

Moreover, the motion study was initialized using SolidWorks assembly *mates* so that the exoskeleton can start in a known position. Mates were defined to position the body, arms, and crutches at certain angles relative to each other and the ground. This allowed the exoskeleton to start in an upright balanced position and pursue the STG technique. It also allowed the exoskeleton to start in an imbalanced position (that would be detected as a fall by the FLC) to simplify the analysis and simulations. It is also important to note that the sensor angle was only possible using two separate angles. SolidWorks is only capable of measuring absolute angles from a certain reference; as such, two angle sensors were defined (that opposed each other) between the torso and the ground. These two angle measurements were subtracted in LabVIEW to determine the direction of inclination of the torso.

Control System

As previously discussed, SolidWorks has a simplistic control mechanism integrated within the software. The built-in control methods are not versatile enough to allow for fuzzy logic control. As such, the software must be interfaced with a control mechanism to allow for the implementation of a FLC.

Interfacing SolidWorks with a Control Software

National Instruments™ created the SoftMotion module that allows for the SolidWorks-LabVIEW interface. The novelty of LabVIEW and SolidWorks is clear: The exoskeleton and control system can be designed, calibrated, and analyzed prior to physical construction. This minimizes the cost of construction and the risks associated with the testing of the exoskeleton.

The documentation on SoftMotion is scarce and one of the few ways to learn to operate the module is through trial and error. This section will serve as an extensive guide to replicate the interface between SolidWorks and LabVIEW using the SoftMotion module. The following list is a step-by-step guide that progresses from the installation of the SoftMotion module up to the simulation of the body in SolidWorks.

1. **SoftMotion Installation**

- a. Install the relevant SoftMotion module version from the NI LabVIEW [website](#) that matches the installed version of LabVIEW.
- b. Activate the module using the activation key (or any other available methods).

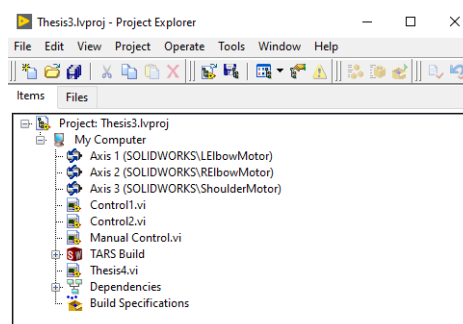
2. **SolidWorks Assembly**

- a. Create a new (or open a pre-existing) SolidWorks assembly.
- b. Activate the motion simulation add-on.
- c. Create the necessary motion study and define the physics parameters in the simulation.
- d. Add motors as necessary. The SoftMotion module only supports distance/angle motors; ensure that the motor parameters match that requirement (Right click on Motor > Edit Feature > Select distance from the drop down menu).
- e. Create sensors as necessary. The sensors are needed to measure the environment and the state of the system. An angle dimension was used as the main sensor for the exoskeleton system. Create a smart dimension through the sketch tab in SolidWorks; the dimension should measure the relevant properties of the system. The sensors can be found in the FeatureManager Design Tree (Right Click on Sensors > Add Sensor > Dimension > Select the relevant dimension).
- f. Ensure that all the sensors measure the correct dimensions. In the case of the exoskeleton, the only possible angle measurement was absolute. As such, I implemented a convenient method of determining the direction of inclination of the body by adding two more sensors that measured the

supplementary angles of the torso. This allowed me to determine the direction of inclination of the torso in LabVIEW by subtracting the two angles and using that value to determine the direction.

3. SoftMotion in LabVIEW

- a. Create a new (or open a pre-existing) LabVIEW project.
- b. Load the relevant SolidWorks assembly into the project (Right Click on My Computer > New > SolidWorks Assembly). This is a feature of Softmotion and will not appear if SoftMotion is installed incorrectly. The predefined motors in the motion study should appear as components of the assembly in the LabVIEW project.
- c. Ensure that the correct motion study is loaded into LabVIEW (Right Click on Assembly > Change Motion Study > Select the appropriate motion study). Synchronize the LabVIEW project to the motion study (Right Click on Assembly > Synchronize to Assembly).
- d. Add SoftMotion Axes. These components can be called within the code to control the motors according to the programmed behavior (Right Click on My Computer > New > NI SoftMotion Axis > Add New Axis).
- e. Ensure that the simulation step size range in LabVIEW matches that of the SolidWorks motion study settings (Right Click on the SolidWorks Assembly > Properties > Adjust the time steps accordingly).
- f. Enable drive on transition to active mode for each axis. This feature can be accessed through each axis' properties (Right Click on Axis > Properties > Select the relevant tab).
- g. Map the SolidWorks sensors to the appropriate digital/analog inputs (Right click on the assembly > Map Sensors > Select the



relevant sensors for each input and keep track of the label of each input). A sample SoftMotion project is shown in Figure 11.

4. LabVIEW Motor Control

- a. Create a new (or open a pre-existing) LabVIEW VI.
- b. Implement a synchronized time loop to the VI. The time loop can be found in the menu and should be adjusted to sync with the scan engine (Right click on Timed Loop in VI > Configure Input Node > Synchronized to Scan Engine).

Figure 11 - Sample LabVIEW project with the SoftMotion axes and the SolidWorks assembly.

- c. Implement a move from the SoftMotion Package (A straight-line move will be used for the purpose of this guide). Adjust the movement to be absolute (Right Click on the Move > Select Method > Absolute); this will allow the program to define the exact movement of the SolidWorks motors. The number of SoftMotion moves should be proportional to the number of SolidWorks motors in the motion study.
- d. Drag the Motion Resources (SoftMotion Axis) from the LabVIEW project into the VI. The motion resources reference the SolidWorks motors and will allow the moves to control the defined axis. Connect each straight-line move with the appropriate motion resource.
- e. Define the speed and distance of each straight-line movement. These could be pre-defined through constants or through user input using the available resources in LabVIEW.

5. LabVIEW Sensor Input

- a. Implement a SoftMotion read in the VI. Attach the relevant SoftMotion axis to the read and create a constant with the appropriate digital/analog input number (attached to the relevant input on the read). These two values will reference the SoftMotion axis and sensor.

- b. Differentiate the sensor signal in LabVIEW using the preexisting derivative block (This was done in the case of the exoskeleton to acquire the angular velocity and acceleration of the body).

6. Running the Simulation

- a. Run the simulation by deploying the axes and the SolidWorks assembly (Select My Computer, the axes, and the SolidWorks assembly > Right click on the selection > deploy). Wait until it initializes before proceeding.
- b. Run the VI using the run button in the VI (alternative: Right click on VI > Run).
- c. Begin the motion study in SolidWorks (Right click on assembly in project > Start Simulation).

7. Important Details on Process

- a. Ensure that the SolidWorks assembly is saved and synchronized to the LabVIEW project prior to running the simulation.
- b. There will be a delay between the VI and the motion study. The magnitude of the delay will depend on PC specifications and program complexity. Programming delays might be necessary to mitigate the delay between the two.
- c. Initialize the SolidWorks assembly using mates, then disable the mates before starting the simulation. This would help create identical starting parameters for every test/simulation.

Exoskeleton Control System

This section is dedicated to discussing the programming architecture in LabVIEW. The parameters of the FLC's were discussed in a previous section; however, a summary of the FLC implementation will be explained in the upcoming subsections.

Sensor Inputs

The first set of sensors and FLC were implemented in a separate timed loop. The output signal of the first FLC determined if the robot was operating within the regular STG motion or if it required an intervention through the emergency control mode. There are four components to any FLC in LabVIEW:

1. The FLC Block – Takes in the input data to implement the defined behavior of the membership functions and rules to create an output signal.
2. The FLC Loader – Specifies the parameters of the FLC by loading in a .fs file into the FLC block. The .fs format is a set of parameters that define the membership functions and rules. These can be created using the built-in Fuzzy System Designer in LabVIEW (Tools > Control and Simulation > Fuzzy System Designer).
3. Inputs – The inputs should be merged if a multiple input FLC is implemented. The inputs can be merged using a Build Array block.
4. Outputs – The outputs of the FLC should be separated if it is a multiple output FLC. The outputs can be used to control the actuators.

The sensor measurement part of the VI is shown in Figure 12.

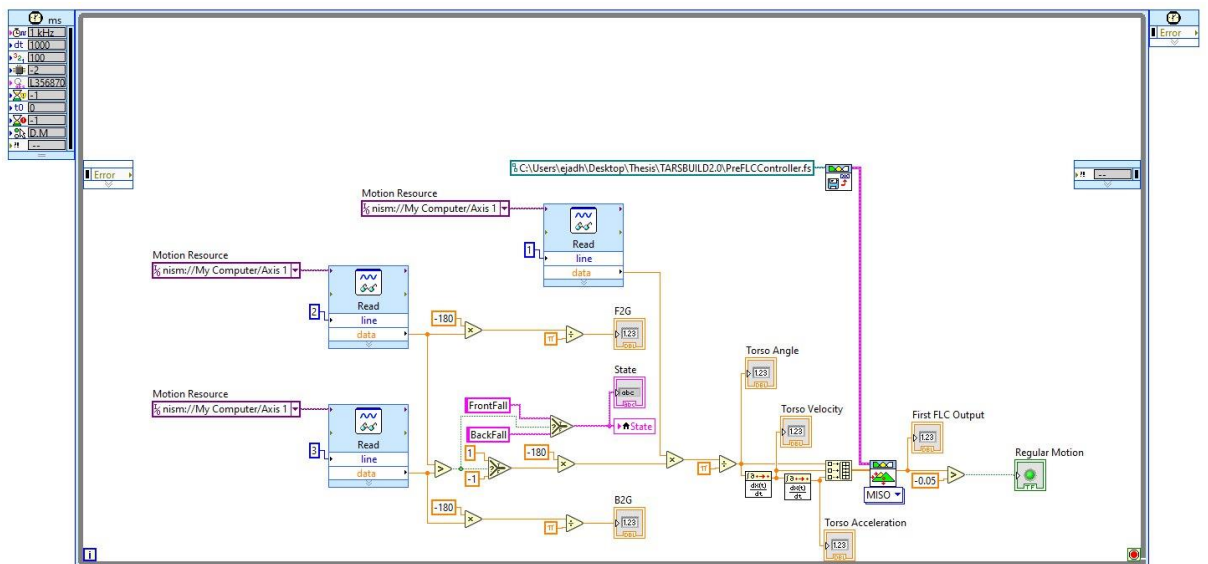


Figure 12 - Sensor measurements and first FLC in LabVIEW.

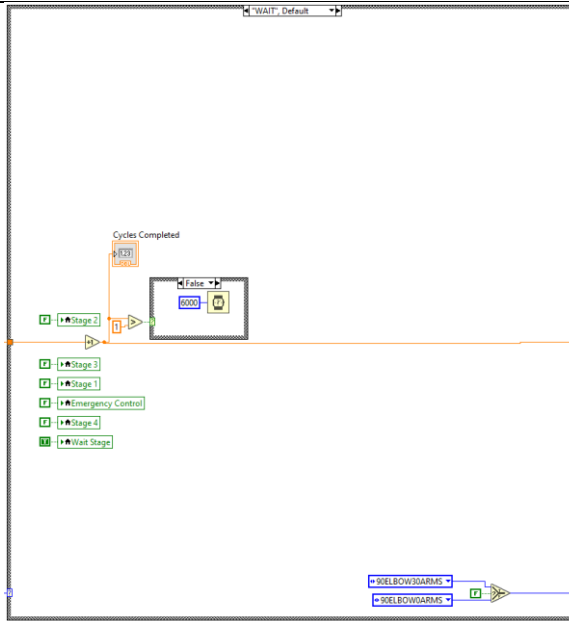
Exoskeleton Behavior

The behavior of the exoskeleton was dictated using a state machine on LabVIEW. State machines are convenient as they allow the user to define separate states that the exoskeleton could operate in. The states mimic the behavior of the STG ambulation technique. State machines can be created using enumerators, that define the individual states and transition between them, and a case structure nested within a timed loop. The individual states are shown in *Table VIII*.

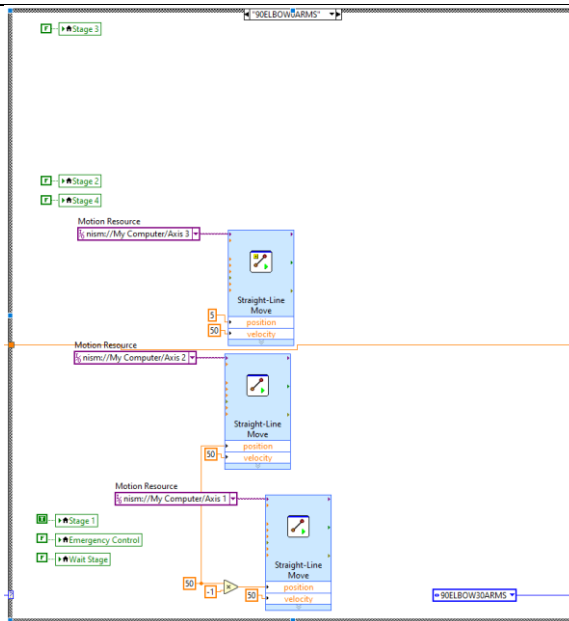
Table VIII

LabVIEW States

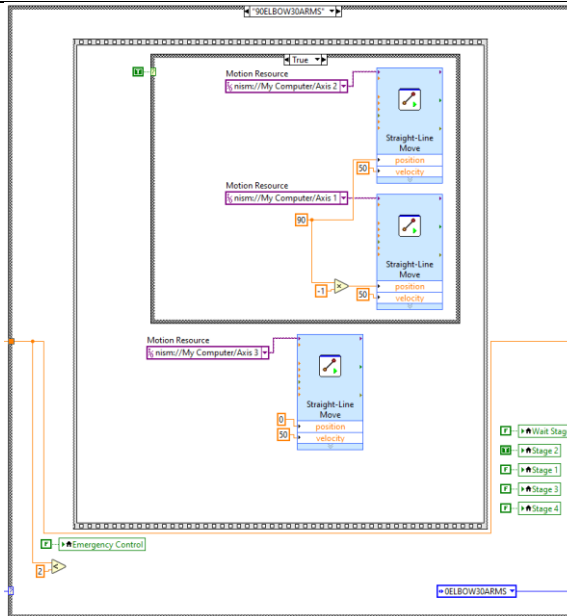
Initialization
State



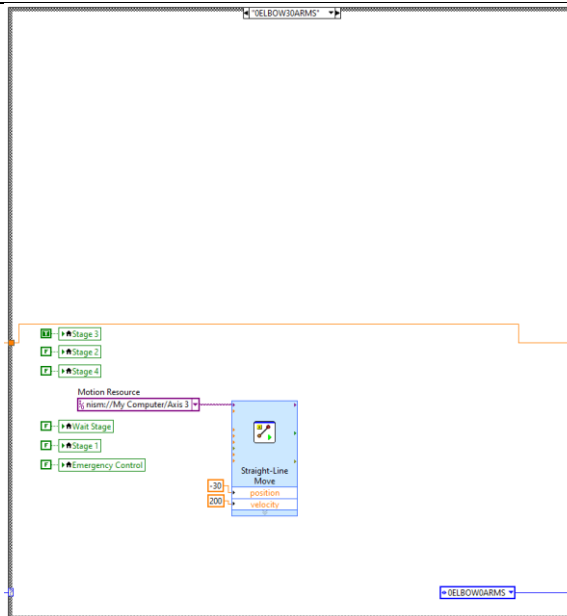
Stage 1



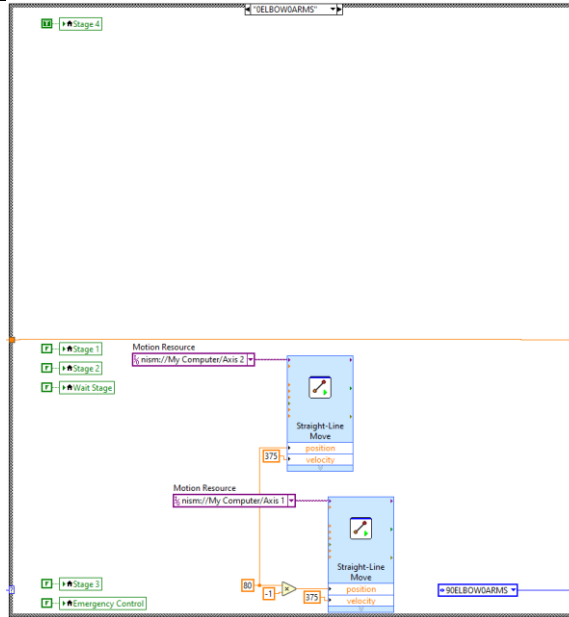
Stage 2



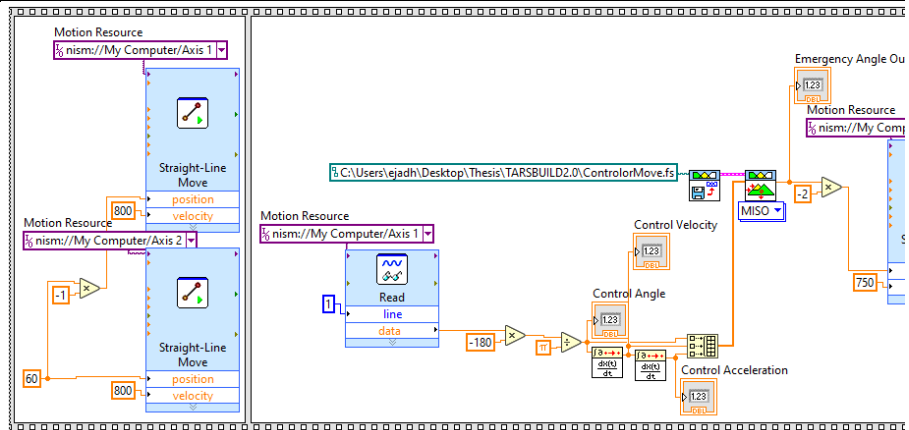
Stage 3



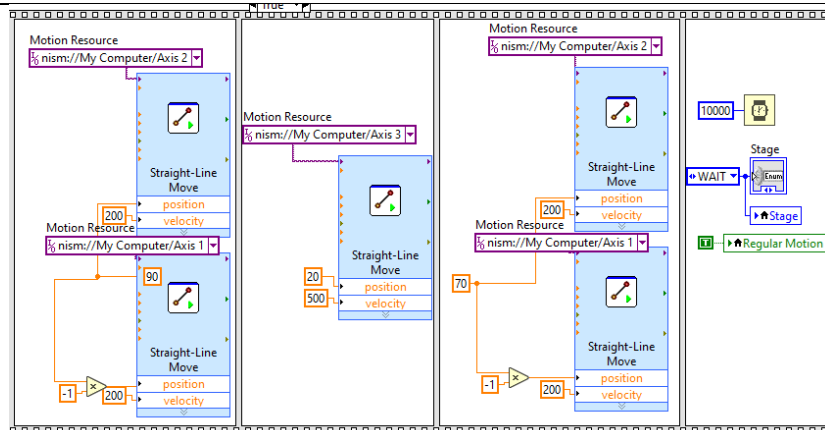
Stage 4



Emergency Control SM Part 1



Emergency Control SM Part 2



Note: The emergency control state was split to fit within the document.

The Booleans were used to create indicators for the UI; these indicators would help identify the current state of the state machine and the resulting motion. The system would switch to the emergency mode movement once the first timed loop would detect that the exoskeleton was falling over and needed intervention. The emergency mode consisted of a sequence of events that would identify the direction of the fall, tuck in the arms, retract the shoulders, extend the arms to an obtuse angle, and catch the body prior to the fall. This emergency mode would proceed by pushing the body forward by fully extending the arms and would switch to the regular motion once the fall is prevented. The user interface was critical in adjusting the membership functions and the behavior of the exoskeleton. It displayed the data from the sensors along with the current state of the state machine and allowed the user

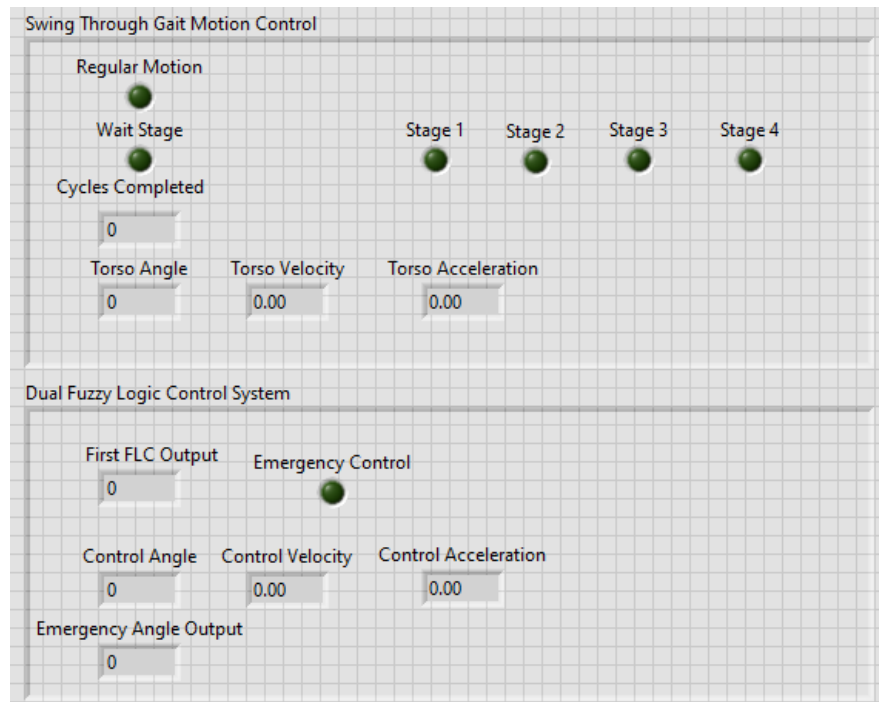


Figure 13 - The user interface of the control system. This helped guide the user by displaying the state of the system and its behavior.

to track the system's behavior and adjust accordingly. The user interface is shown in Figure 13.