User Intent Detection and Control of a Soft Poly-Limb

by

Nicholas M. Vale

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2018 by the
Graduate Supervisory Committee

Panagiotis Polygerinos, Chair
Wenlong Zhang
Panagiotis Artemiadis

ARIZONA STATE UNIVERSITY

May 2018

ABSTRACT

This work presents the integration of user intent detection and control in the development of the fluid-driven, wearable, and continuum, Soft Poly-Limb (SPL). The SPL utilizes the numerous traits of soft robotics to enable a novel approach to provide safe and compliant mobile manipulation assistance to healthy and impaired users. This wearable system equips the user with an additional limb made of soft materials that can be controlled to produce complex three-dimensional motion in space, like its biological counterparts with hydrostatic muscles. Similar to the elephant trunk, the SPL is able to manipulate objects using various end effectors, such as suction adhesion or a soft grasper, and can also wrap its entire length around objects for manipulation. User control of the limb is demonstrated using multiple user intent detection modalities. Further, the performance of the SPL studied by testing its capability to interact safely and closely around a user through a spatial mobility test. Finally, the limb's ability to assist the user is explored through multitasking scenarios and pick and place tests with varying mounting locations of the arm around the user's body. The results of these assessments demonstrate the SPL's ability to safely interact with the user while exhibiting promising performance in assisting the user with a wide variety of tasks, in both work and general living scenarios.

ACKNOWLEDGMENTS

I would like to thank my thesis director, Dr. Panos Polygerinos, for allowing me to be a member of the Bio-Inspired Mechatronics lab, as well as his guidance and support throughout the course of this research. I have enjoyed my time as a member of the lab and appreciate the mentorship he has offered me. In addition, I would like to thank my committee chairs, Dr. Wenlong Zhang and Dr. Panagiotis (Panos) Artemiadis, for their time and feedback in improving my research and thesis. Finally, I would like to thank Pham Huy Nguyen, Saivmal Sridar, and other fellow students of the Bio-Inspired Mechatronics lab for the numerous hours of assistance they provided to me while I completed this research.

To my family and friends, thank you for always supporting me.

TABLE OF CONTENTS

Page

CHAPTER Page

APPENDIX Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Table Page

Chapter 1

INTRODUCTION

Humans extensively use their upper appendages to accomplish a variety of tasks throughout day-to-day life, however impairments such as cervical spondolytic myelopathy (CSM) often limits one's ability to execute certain motor tasks due to decreased upper extremity function. Caused by general deterioration of the corticospinal tract, or by injury to the cervical region of the spinal cord, CSM is the most prevalent cause of spinal cord dysfunction in individuals across all age ranges (Lubelski et al. 2016). CSM can lead to upper limb impairment that causes numbness, weakness of muscle, and loss of fine motor control, ultimately limiting the individual's ability to perform motor tasks such as long reaches, grasping of objects, and precise positioning of the upper limbs. These motor tasks are employed when picking up a cup or performing other activities of daily living (ADLs). For individuals with neuromuscular impairments such as CSM, the inability to accomplish simple ADLs lowers quality of life and limits independence (Kirby et al. 2011). Therefore, a great focus of modern robotic technology has been on providing a means for these individuals to regain value of life through wearable robotic manipulators. Robotic manipulators can be worn on the body to offer the user additional, supernumerary appendages for support, assistance, or enhancement of existing capabilities. These robots do not need to match the user's skeletal structure nor replace biological limbs to enhance the user's capability, and therefore do not impose complications that come with prosthetic devices and exoskeletons (Bogue 2009; Gopura, Kiguchi, and Bandara 2011), such as restriction of motion or complex don and doff methods. While prosthetic devices improve value of life for amputees, they are unnecessary for impaired individuals that still have their upper limbs. Additionally, exoskeletons provide functional assistance for individuals

with upper limbs impairments but impose risks to the wearer's safety. For example, the application of external torques to the joints of arthritic individuals could cause pain and injury. Therefore, robotic manipulators benefit an impaired population that require task assistance.

Several wearable robotic manipulators have been developed to assist individuals with upper limb impairment by aiding their ability to execute ADLs. For example, the Robotic Sixth Finger is an assistive device, worn on the wrist, that compensates for diminishing hand function from chronic stroke by providing the wearer with an additional robotic finger to perform grasping tasks by constraining objects against their paretic limb (Salvietti et al. 2017). Another example device, the Supernumerary Robotic Fingers, assists individuals with upper limb impairment perform single-hand grasping and manipulation tasks using two additional robotic fingers on the forearm (Wu and Asada 2016). Use of wearable robotic manipulators, however, is not limited to clinical applications for impaired populations. These robots can be used among healthy individuals to complete day-to-day tasks, and have even shown use in industrial settings (Parietti and Asada 2014), as exemplified in Figure 1 below. Besides additional fingers, examples of wearable robotic manipulators have been demonstrated for functional benefit as robotic legs (Kurek and Asada 2017; Parietti et al. 2015) and arms (Sasaki et al. 2017; Parietti and Asada 2017; Vatsal and Hoffman 2017). While wearable robotic limbs have shown promising benefits across multiple realms, most have seen limitations as a result of their rigid designs. These challenges concern the safety of the user, of which include bulkiness, weight and the interaction between human tissue and rigid materials (del-Ama et al. 2012).

Recent work in the emerging field of soft robotics aims to overcome the obstacles associated with rigid designs. Unlike conventional robots, soft robots, also known as intrinsically soft devices (ISDs), are made of highly compliant materials that are

lightweight, low-cost, and offer a high power-to-weight ratio (Polygerinos et al. 2017). Additionally, their soft nature allows for high adaptability to changing environments and for conformity to the user's body with no potential for damage, making them safe for human-robot interaction (Polygerinos et al. 2017). Current work in the field of wearable soft robotic manipulators has primarily focused on pneumatically-actuated soft fingers. The Soft-SixthFinger (Hussain et al. 2017) and other modular, supernumerary robotic grasp-assist devices (Tiziani et al. 2017) have demonstrated the ability to perform grasping tasks for ADLs in a safer manner than rigid robotic manipulators. While beneficial, these specific soft devices are constrained to a single plane of motion (open-closed or grasp-release) and cannot assist the user in executing a wide variety of tasks along higher degrees of motion.

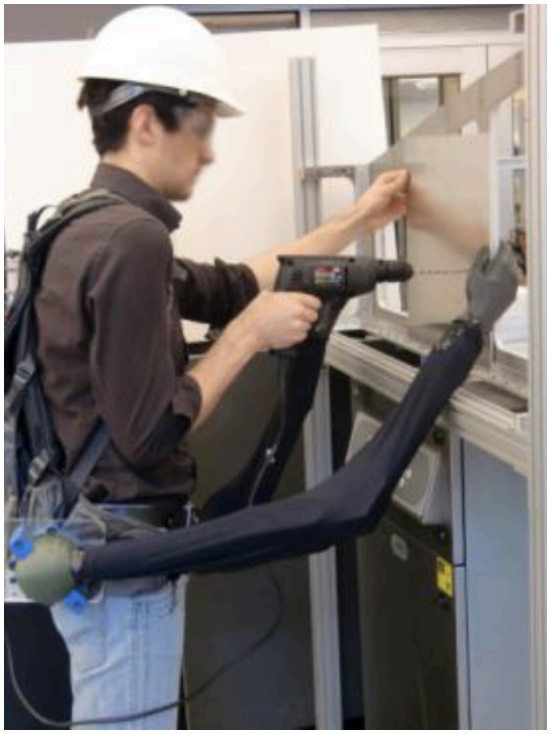

Figure 1: Example of wearable robotic manipulator with two additional arms being used to complete drilling tasks (Llorens-Bonilla, Parietti, and Asada 2012).

Since wearable robotic manipulators do not replace the wearer's biological limbs, but rather provide extra appendages, questions concerning adaptability and control may arise. Recent studies on the central nervous system (CNS) have discovered that humans are capable of adopting and learning to control additional limbs (Guterstam, Petkova, and Ehrsson 2011; Tsakiris et al. 2010). Therefore, humans should be able to complete tasks, such as ADLs, by using additional robotic limbs. While it would require some training of the CNS, utilizing additional robotic limbs should lead to more effective task completion due to an increase in the number of working limbs. The control of robotic manipulators plays an important role in exploiting its functional benefits. An important consideration for the method of detecting user intent is the use of a working biological limb to control a robotic manipulator. To maximize human-robot capability, it is ideal for the user to control the additional robotic appendage using a method that does not require use of a functional limb, as it enhances their potential for multi-tasking or executing tasks with all limbs, biological and robotic. Research has shown that humans are capable of controlling robotic manipulators through biological signals from the frontalis muscle on the forehead (Hussain et al. 2017; Salvietti et al. 2017), pectoral and abdominal muscles (Parietti and Asada 2014), forearm muscles (Ajoudani et al. 2014), the foot (Sasaki et al. 2017), and the elbow (Wu and Asada 2016). Examples of other user intent detection methods include robotic limb control via joystick (Maheu et al. 2011), neural interface (Hochberg et al. 2012), infrared hand tracking (Bassily et al. 2014) and vocal recognition (House et al. 2009). User intent detection and control of robotic manipulator will be discussed further in Chapter 2.

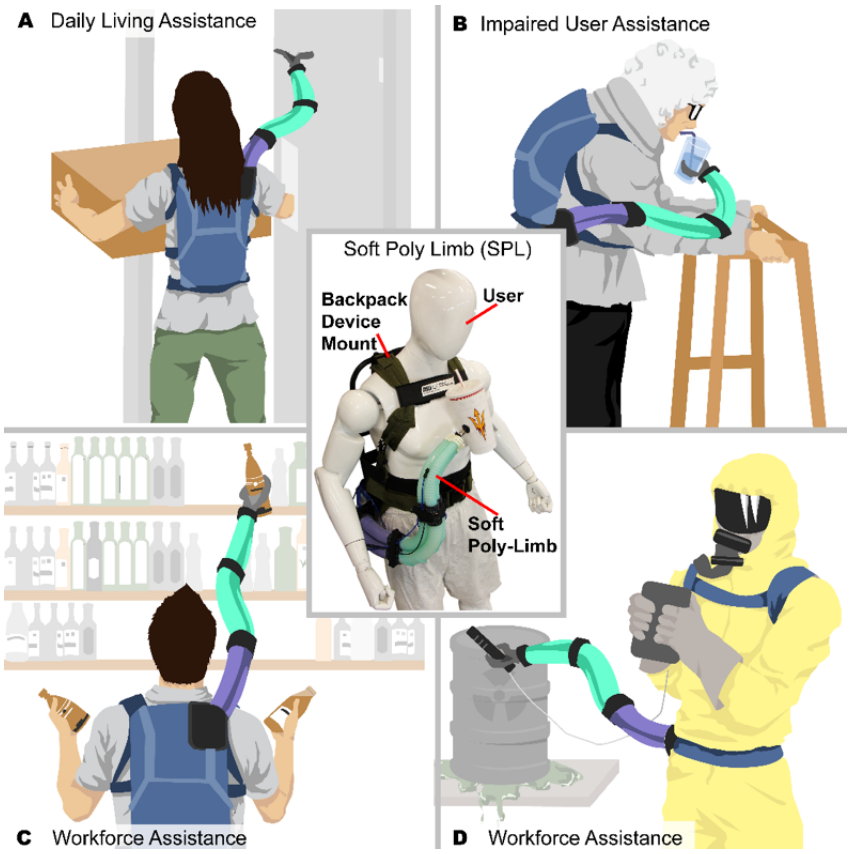

Figure 2: The prototype SPL (center) mounted on a back frame system is a soft, lightweight, wearable robot that is designed to assist users with functional tasks. Possible use cases include: **(A)** Daily living assistance: opening a door when the user's hands are full from carrying objects. **(B)** Impaired User Assistance: assisting an aging user with feeding. **(C)** Workforce Assistance: assisting a grocery store worker restack shelves with products quicker and access higher shelves that cannot normally be reached. **(D)** Hazardous Task Assistance: assisting a user measure, from safe distance, the radiation levels and collect radioactive samples on-site.

The combination of wearable manipulators and ISD principals has led to the development of a soft, wearable robotic manipulator, the Soft-Poly Limb (SPL), which is highlighted in Chapter 3. This continuum manipulator has the potential to provide assistance with ADLs and may prove highly beneficial to both populations with and without upper limb impairment (Ajiboye et al. 2017). Possible use cases of the SPL are shown in Figure 2. This thesis specifically focuses on the user intent detection and control of the Soft Poly Limb. The goal was to develop a method to pneumatically control the soft

limb while exploring various methods of detecting user intent. High-level control architecture was first realized using a joystick, then improved using an inertial measurement unit (IMU). While these intent-detection methods required hand use to control, they were vital to characterization and development of the control scheme. Surface electromyography (sEMG) was briefly investigated to offer control without sacrificing use of a functioning limb.

1.1 Organization of Paper

This thesis is organized to first provide critical background information about this project in Chapter 2, with a focus on user intent and robot control. This includes a literature review of existing wearable robotic manipulators, user intent modalities, and soft robots. It should be noted that the SPL is a novel soft manipulator, therefore no existing control schemes exist for a device such as this. An overview of the Soft Poly Limb and intent-detection modalities is provided in Chapter 3. This includes the final design of the limb and its modular system, and an explanation of the low-level and high-level control architectures. The low-level control includes the general process of robot actuation, mode selection end effector mapping, as well as a method to reduce the effects of dynamic instabilities. High-level control architecture is explored through three modalities of user intent detection. Chapter 4 encompasses an evaluation of the control system. This includes an assessment of user intent mapping to limb positioning, as well as workspace, payload, and various interaction tests. The results of these tests are discussed in Chapter 5. Finally, Chapter 6 examines the conclusions that were drawn from the project thus far, as well as future directions that can be pursed. Following the main chapters, all works cited in this thesis are exhibited in a reference section. Information supplemental to this project is contained within appendices at the end of this manuscript, such as control code.

Chapter 2

BACKGROUND RESEARCH

2.1 Introduction

A literature review had to be performed in a few areas before the control architecture could be developed. The first topic investigated was existing devices that offer additional robotic limbs. This was important for understanding current uses of wearable robotic limbs, and how to tailor the control of the SPL to accomplish similar tasks, or activities of daily living. Once examples of additional robotic limbs were identified, the methods of detecting user intent for robotic limbs was explored. This information was significant to the design of the control architecture to offer the user a simple, yet effective method for controlling the SPL. Finally, the field of soft robotics was briefly explored, and examples of soft robotic limbs are studied to understand current methods of control. This includes pneumatic actuation techniques to manipulate the limb as desired. All examples of robotic limbs and their control methods were assessed and considered in the development of the SPL control architecture.

2.2 Wearable Robotic Limbs

Various wearable robotic limbs were briefly investigated to understand current limb designs and the purpose they serve the user. This included exploring the tasks these limbs were used to accomplish and considering if they could be accomplished by the user with the SPL. It can be noted that additional robotic limbs have been designed for many users, however the focus of this review is on three types of appendages: legs, arms, and fingers.

The first appendage, robotic legs, have been used for load support and balance stabilization in both static and dynamic situations. One example of such devices is the MantisBot, a wearable robotic system that provides the wearer two additional legs for stabile task performance near the ground (Kurek and Asada 2017). This device employs controllable actuators to actively support the wearer by adapting to the change in forces and moment as the user moves around. The device assumes that the user is in a crouched or crawling position, as the additional legs hold up the user through a rigid plastic harness worn around the upper torso. Another example of a wearable robotic device that provides the user an additional set of legs is the Supernumerary Robotic Limbs (Parietti et al. 2015). These additional legs compensate for the user's weight in static conditions, allowing for autonomous balance support, as highlighted in Figure 3. Additionally, the device assists the user with bipedal locomotion by providing additional contact points around the user and thus, reducing the load on the joints.

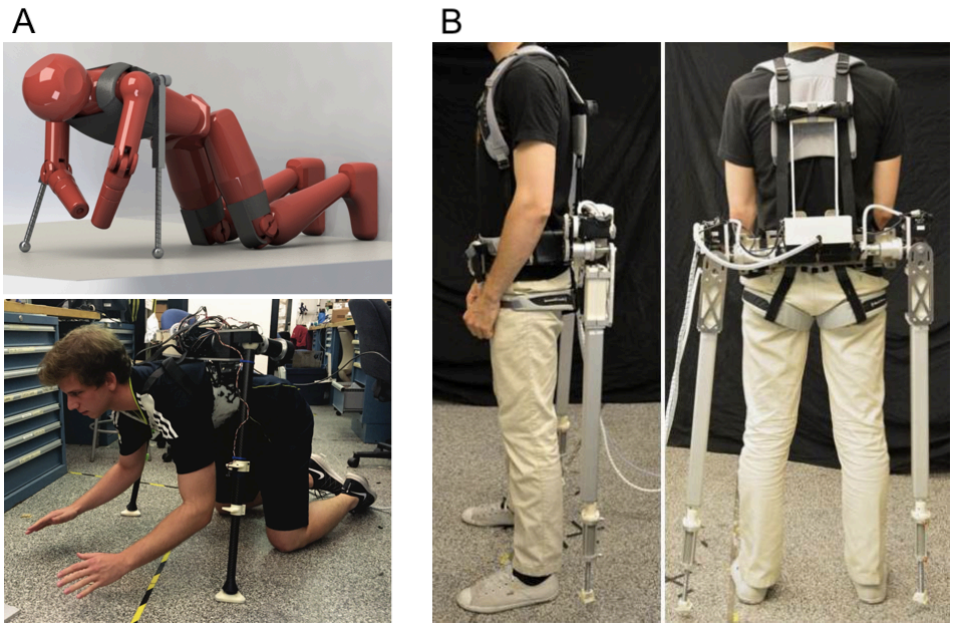

Figure 3: (A) MantisBot concept (top) and prototype (bottom) devices providing support to a user in a crouched position (Kurek and Asada 2017). **(B)** The Supernumerary Robotic Limbs device worn by a user for additional support while standing (Parietti et al. 2015).

Figure 4: Supernumerary Robotic Limbs assisting a user with the assembly of an aircraft fuselage. The left robotic limb provides contact support while the right limb performs a drilling task (Parietti and Asada 2014)**.**

Robotic arms, the second highlighted robotic appendage, are used for grasping, moving, and holding objects, as well as supporting the wearer. One example of an additional arm is a wearable robotic forearm (Vatsal and Hoffman 2017). This device is attached at the elbow and forearm, providing an additional arm that works along with a biological limb. The downside to such a design is that movement of the biological limb is required for positioning the robotic limb. This device assisted in stabilizing the user while on a ladder to allow them to perform bi-manual tasks, showing the capability for providing stability assistance. This device was also used to grasp objects beyond the wearer's natural reach, further augmenting the user's capability. Another example of a wearable additional arm robot is the Supernumerary Robotic Limbs (Parietti and Asada 2014). Similar to the robotic legs with the same name, these additional arms brace the wearer's body; however,

this device offers additional assistance by interacting with the environment, such as performing a drilling task while the user simultaneously performs other tasks, shown in Figure 4. This robot, worn around the torso, provides two additional arms to the user. This specific system focused on providing stabilization to the user with one limb and task assistance with the other limb.

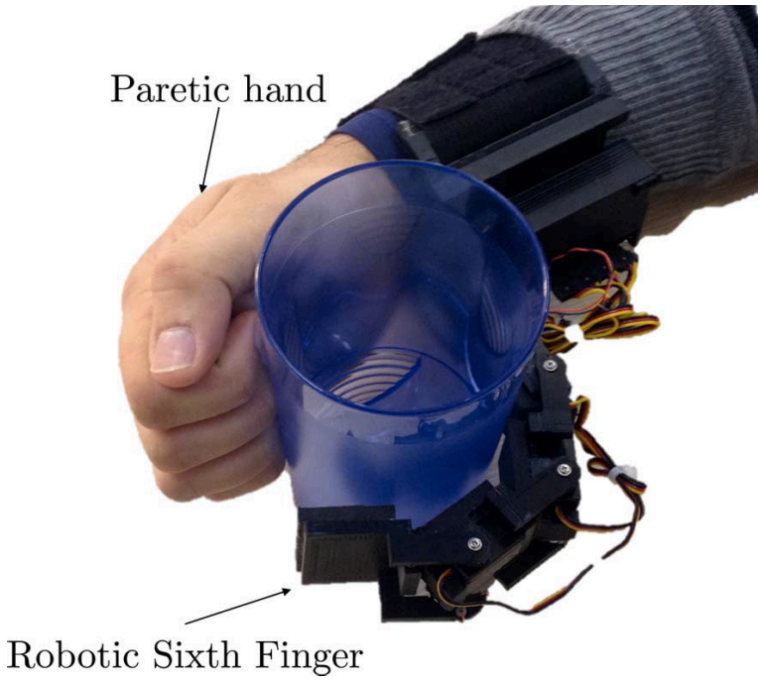

Figure 5: The Robotic Sixth Finger prototype assisting a user perform grasping tasks with the paretic hand (Salvietti et al. 2017)

Additional robotic fingers, the final wearable appendage, are primarily used for grasp assistance. The majority of wearable robotic finger devices aim to assist impaired populations that struggle performing grasping tasks and other activates that require use of the hands. One example of such a device is the Robotic Sixth Finger, an assistive device that provides an additional robotic finger to the wearer and is worn on the inside of the wrist (Salvietti et al. 2017). This device compensates for a reduction in upper limb function

in impaired users by constraining the motion of an object against a paretic limb, depicted in Figure 5. Another example of an additional finger robot is the Supernumerary Robotic Fingers (Wu and Asada 2016). This device is mounted at the wrist and offers two robotic fingers on the ventral side of the forearm to grasp and support objects to allow the user to manipulate the object with their hand. While helpful in performing their specific tasks, these limbs are limited in the diversity of tasks that they can assist the user complete.

2.3 Detection of User Intent and Control of Robotic Limbs

It was important to explore a number of methods for detecting user intent to understand the current approaches for controlling assistive robotic limbs. The method of controlling a robotic limb plays a significant role in the type of tasks it can accomplish as well as the population that the device can assist, such as healthy or impaired populations. In comparison to healthy users, impaired users are more restricted in terms of effective user intent detection methods. For example, it is difficult for users with upper limb impairments to perform upper body movements to control a robot, and is sometimes impossible, depending on the level of impairment.

The simplest and most straightforward method of detecting user intent is through a joystick, as it does not required acquisition of biological signals to control a robotic device. In one particular study, a JACO robotic arm was used to help wheelchair users with upper-extremity disabilities perform activities of daily living through the use of a three axis joystick, shown in Figure 6 (Maheu et al. 2011). The JACO arm is a three-finger, robotic manipulator with seven degrees of freedom (DOF), which can be mounted on a wheelchair, table or other workspaces. Positioning of this arm is accomplished by moving a joystick around its three axes and pressing buttons to switch between control of the arm, hand, and fingers. The users manipulated the robotic arm with the joystick to perform

activities such as grasping a bottle and pushing buttons on a calculator. While these users had some level of upper extremity impairment, they were capable of performing dexterous tasks with the robotic arm via joystick control. However, not all users are able to manipulate a joystick, therefore other methods of detecting user intent using biological signals are explored.

Figure 6: The JACO robotic manipulator and three axis joystick to assist impaired users perform activities of daily living. Movement of the joystick around its three axes positions the robotic arm and grasps objects using the fingers of the manipulator, while buttons on the joystick determine the mode of control (hand positioning, hand orientation, or finger use) (Maheu et al. 2011).

One method of detecting user intent to control robots is through surface electromyography (sEMG), which senses a change in electrical impulses of chosen skeletal muscles. For example, an sEMG sensor distinguishes a change in electrical activity through Ag/AgCl electrodes placed on a muscle as a user activates said muscle. Various robots have been controlled using sEMG signals measured from the frontalis muscle on the forehead (Hussain et al. 2017; Salvietti et al. 2017) and the extensor digitorum communis (EDC) and flexor digitorum superficialis (FDS) muscles on the forearm (Ajoudani et al. 2014). As the wearer voluntarily moves their eyebrows upwards or

opens/closes their hands, these systems detect a change in electrical signal through the sEMG electrodes placed on the respective muscles and actuate the robotic manipulators to perform flexion and extension motions for grasping assistance. One particular wearable robot employed a modular sensor suit that measured sEMG signals from the wearer's torso (Parietti and Asada 2017). The pectoral muscles were utilized for forward movement of the robotic limbs while abdominal muscle activity controlled movement in the backward direction, as portrayed in Figure 7. One benefit of sEMG is that it does not require a mechanical displacement to control robots, as required with a joystick or variable force mechanisms. In addition, sEMG can measure signals from almost any muscle on the body, offering diverse options for controlling a robotic device without sacrificing use of limbs.

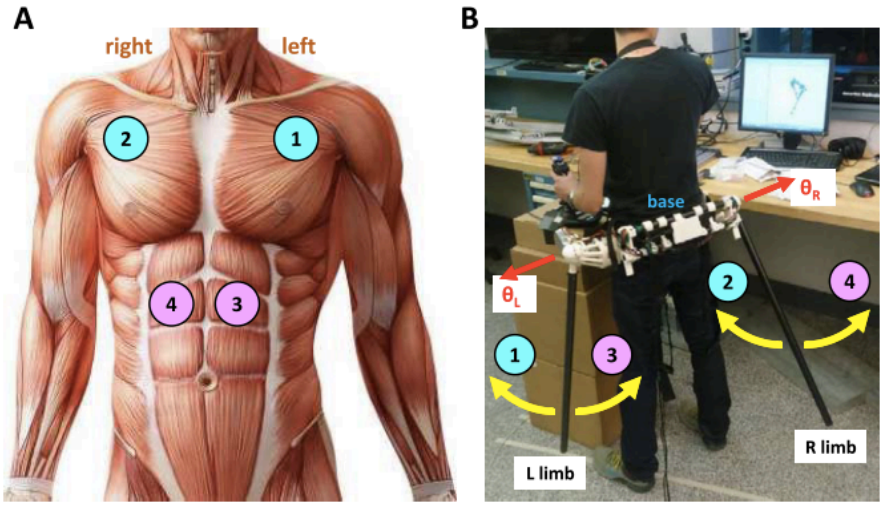

Figure 7: sEMG control of robotic limbs. **(A)** The placement of sEMG sensors on the pectoral (light blue) and abdominal (light purple) muscles used to control the robotic limbs. **(B)** Representation of the robot's potential motion. Forward motion (light blue) is based on muscle activation of pectoral muscles, while backwards motion (light purple) is associated with abdominal muscle activation (Parietti and Asada 2017).

Another method of detecting user intent is through a motion capture system, which tracks three-dimensional trajectories and correlates the trajectories to robot movement.

An example of a wearable robot that uses motion capture is the MetaLimbs (Sasaki et al. 2017). Reflective markers are attached to the user's foot, and movement of the foot within the motion caption camera workspace subsequently moves the additional arms to perform tasks around the user. Another example of a motion-capture controlled robot is highlighted in (Bassily et al. 2014). In this work, a JACO robotic arm is controlled via a Leap Motion Controller. This infrared hand tracking device records grasp and release motions and mimics the action with the robotic arm. The downside to motion capture modalities is that the environment in which the limb can be controlled is restricted by the motion capture camera setup.

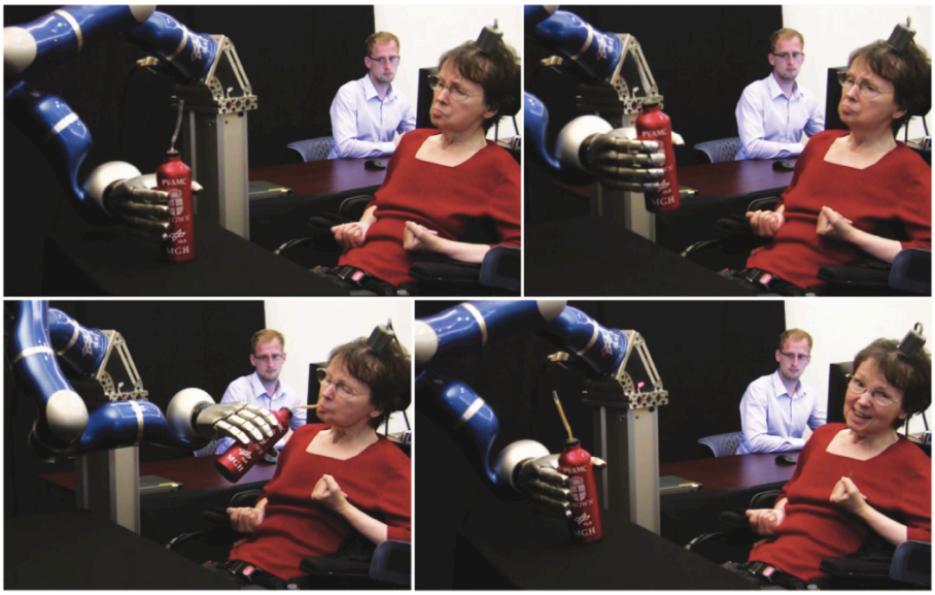

Figure 8: Sequence of images depicting a user controlling a robotic arm through a neural interface to grasp a bottle off of a table, move it towards her, take a drink, and set the bottle back on the table (Hochberg et al. 2012).

Other hands-free control methods have been documented as well, such as a control through a neural interface (Hochberg et al. 2012). The user was able to control a robotic arm using a 96-electrode array implanted in the brain, to perform tasks such as grasping and drinking from a bottle, shown in Figure 8. In a final example, control of a robotic arm,

called the VoiceBot, was demonstrated through the use of vocal recognition (House et al. 2009). In this system, the user was able to manipulate the robot around its workspace by creating vowel sounds. These methods of detecting user intent are highly beneficial for impaired populations, especially individuals suffering from paralysis, as they do not require biomechanical manipulation of sensors. While beneficial, the collection and processing of these neural and vocal signals requires highly complex procedures. For a more comprehensive review of robotic controls, see (George Thuruthel et al. 2018).

2.4 Soft Robotics

While the robotic manipulators previously highlighted in this thesis are capable of assisting in task execution, they are costly, heavy, and bulky due to their rigid design; therefore, to overcome these limitations, the benefits of soft robotics technology are explored. Soft robotics is an emerging field in which highly compliant materials are utilized to accomplish tasks that traditional, rigid robots are incapable of performing (Iida and Laschi 2011; Ilievski et al. 2011; Majidi 2014; Polygerinos et al. 2017). These inherently flexible devices are engineered to change morphology and produce novel types of movement through fluidic actuation, making soft robots capable of moving through and adapting to various environments (Shepherd et al. 2011). Additionally, the soft material composition allows these robots to perform under actuated maneuvers, such as using the entirety of their soft body to grasp and delicately manipulate fragile or deformable objects (Kim, Laschi, and Trimmer 2013). Furthermore, these intrinsically soft devices also offer safer physical interaction between the human and robot due to their highly compliant nature. Finally, soft robotic technology employs inexpensive materials, such as silicone, fabrics and thermoplastic polyurethane, to create devices with a wide range of uses. (Polygerinos et al. 2017).

There have been several wearable soft robotic limbs developed to assist in completing tasks. Unlike traditional robots, soft robots are often fluidic actuated and offer safer integration with the user, due to their compliant nature. Additionally, some soft robots are capable of conforming to their environment to perform complex tasks. An example of a wearable, soft robotic limb is the Soft-SixthFinger (Hussain et al. 2017). Similar to other robotic fingers previously highlighted, this is an active assist device to allow post-stroke users to grasp objects between the additional finger and paretic limb. This device is controlled via a pulley system rather than pneumatic actuation but employs soft robotic design principles. Another example of a wearable soft robotic device is the Supernumerary Robotic (SR) grasp-assist device (Tiziani et al. 2017). This device achieves a similar goal as other robotic appendages, however does so through the use of multiple soft digits, shown in Figure 9.

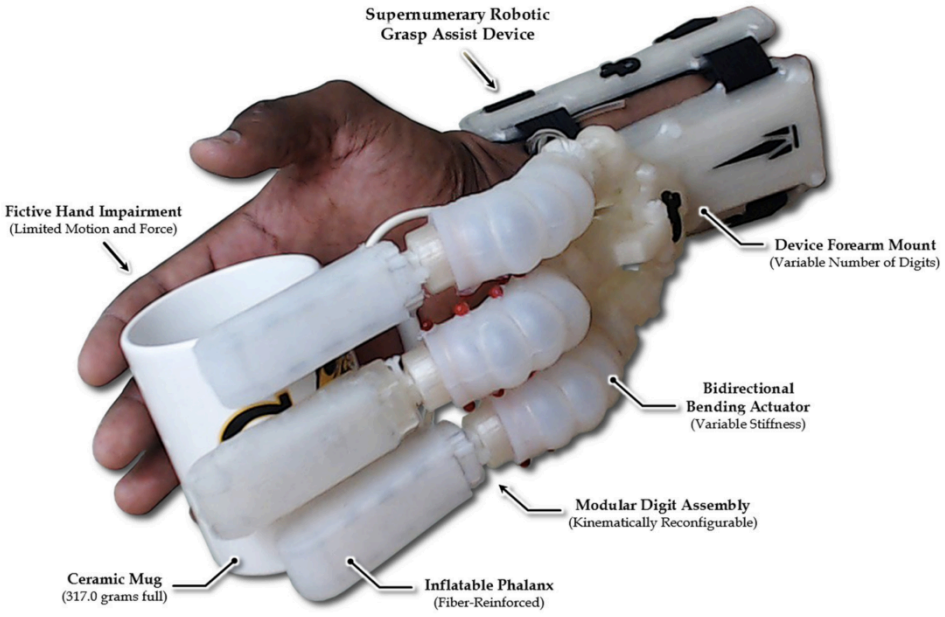

Figure 9: The prototyped Supernumerary Robotic grasp-assist device being used to grasp a ceramic mug by constraining motion against the biological hand.

Chapter 3

OVERVIEW OF SOFT CONTINUUM ARM AND CONTROL ARCHITECTURE

3.1 Soft Poly Limb (SPL)

Figure 10: The prototype SPL mounted on a back frame system is a soft, lightweight, wearable robot designed to assist users perform functional tasks.

The Soft Poly Limb (SPL) is a soft, modular upper limb developed to assist various user populations complete tasks, such as activities of daily living. Presented in Figure 10, this limb is a wearable robotic manipulator comprised of three modular segments called three-chambered actuators (3CAs). Each 3CA is comprised of three ring-reinforced actuators (RRAs), which are tubular elastomeric actuators made of soft-silicone material (Dragon Skin 30, Smooth-On Inc., PA) surrounded by robust ABS-printed rings around the outer wall. These rings serve to restrict radial expansion of the actuator upon pressurization, thus promoting linear extension. Interaction between bundled RRAs, or a single 3CA, produces a bending motion with three degrees of actuation (DoA) in space

(Nguyen et al. 2017). ABS-printed (RGD531, Stratysys, MN) modular connector pieces between each 3CA join the three segments in series (as seen in Figure 11) to create a soft robotic continuum limb, with a total of nine DoAs from its nine RRAs. These highly maneuverable segments allow the soft limb to perform complex movements that other wearable robotic appendages are incapable of executing.

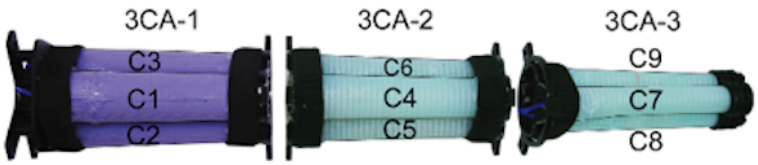

Figure 11: The SPL body is made of three tapered, modular segments, called three-chambered actuators (3CAs), when connected in series. Each 3CA is consists of three hyper-elastic actuators, called ring-reinforced actuators (RRAs) arranged in an equilateral triangle configuration. Each RRA extends axially upon pressurization, therefore interactions between RRAs in each segment creates a bending motion in three directions.

The SPL is approximately $1.6kg$ in total weight and $0.57m$ in total length upon complete assembly. The length of the SPL was designed to mimic the length of the average human arm from the center of the shoulder to the center of the wrist, which measures roughly $0.59m$ (Plagenhoef, Evans, and Abdelnour 1983). Therefore, the user is capable of interacting with the SPL to perform tasks, for example, feeding themselves or holding and manipulating objects with both biological and robotic limbs, within a workspace to which they are accustomed. Following the lightweight principals of soft robotics, the weight of the SPL is minimized through a tapered shape (as seen in Figure 11) similar to the highly flexible and articulated tapered appendages utilized by animals such as the octopus and elephant (Kier and Smith 1983). The decreased limb diameter along subsequent segments reduces the weight of the SPL at its distal end, bringing the center of mass closer to the body of the user. Thus, reducing the potential for torsional affects caused by gravitational

torque and increasing the acceleration of locomotion, due to an increased ratio of force over mass (Charles et al. 2016).

3.1.2 Wearable System

The SPL is mounted to an external back frame that is worn by the user, similar to a backpack, offering a simple and recognizable don and doff procedure. The frame includes a waist strap, which effectively shifts the load of the back frame and SPL closer to the body center of mass (i.e. mid-to-lower back), therefore maximizing postural comfort and minimizing energy cost for the user (Knapik, Reynolds, and Harman 2004; Devroey et al. 2007). Throughout characterization of control methods and functional testing, the primary mounting position of the SPL was the right hip, however the frame allows for a variety of mounting positions, as displayed in Figure 12, for increased operational space to execute tasks in different workspaces around the user's body.

Figure 12: The back frame provides a simple, comfortable means for wearing the SPL. It offers the ability to mount the limb at different positions to work in various task spaces. The opaque limbs are the positions used for testing the SPL in this work, while the translucent limbs provide an example for other possible mounting positions.

3.1.2 End Effectors

In light of the SPL's modular design, various end effectors can be equipped to the limb for interaction among different environments. The user can substitute a variety of functional end effector modules as required to achieve a particular task. Shown in Figure 13, numerous end effectors have been developed and utilized by the SPL, such as a suction cup, a soft robotic grasper, and a multi-tool holder. These, however, are only a few of many potential end effector modules that can be integrated with the limb, with the only limitation being the end effector weight.

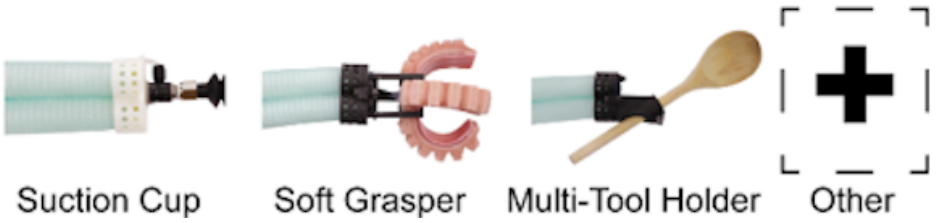

Figure 13: The SPL end effector is reconfigurable depending on functional end effector module required to complete a certain task. In addition, the entire body of the SPL can be utilized as an end effector to grasp large objects.

A robotic manipulator such as the SPL, is not limited to task accomplishment through use of modular end effectors and can be utilized to execute tasks without graspers or other variable end effector tools. The body of the entire soft arm can be used to perform grasping and carrying tasks by effectively wrapping its continuum structure around objects.

3.1.3 Mannequin

A human mannequin, shown in Figure 10 and Figure 12, wears the back frame, with the SPL attached, when the system is not in use. The mannequin played a key role in the development of the control architecture, offering a safe and realistic environment to

characterize movement of the limb. Furthermore, the mannequin was utilized for evaluating control of the limb throughout preliminary interaction tests, such as the evaluation of the SPL's workspace and payload capability. It helped ensure that the control method was stable in scenarios representative of human wear and use, without requiring a human to wear the manipulator.

3.2 Low-Level Architecture

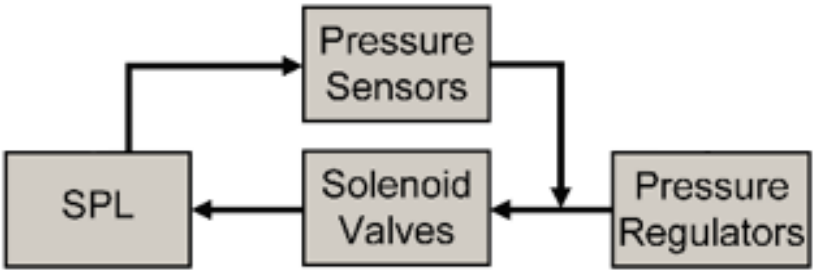

Figure 14: Low-level control flow diagram. The pressure necessary to create a desired motion by each 3CA segment is delivered to each RRA in the SPL and monitored by a pressure sensor.

Low-level pneumatic control of the SPL is accomplished through a closed-loop system, shown in Figure 14, by varying the pressure within each RRA to create various bending motions. Each RRA is pneumatically actuated using two fast-switching, high-flow solenoid valves (MHE3-MS1H-3/2G-1/8-K, Festo, Esslingen, DE) and one pressure regulator (ITV1050-21N2CL4 Pressure Regulators SMC). The solenoid valves allow for both independent and simultaneous actuation of the RRAs, offering the user the ability to control motion of the entire limb or each individual 3CA segment. The pressure in each soft actuator and its respective pneumatic line is monitored and adjusted by a pressure sensor (PSAN-1C(P)V, Autonics, Mundelein, IL). A total of 18 solenoid valves, 9 pressure regulators, and 9 pressure sensors are utilized to govern movement of the SPL.

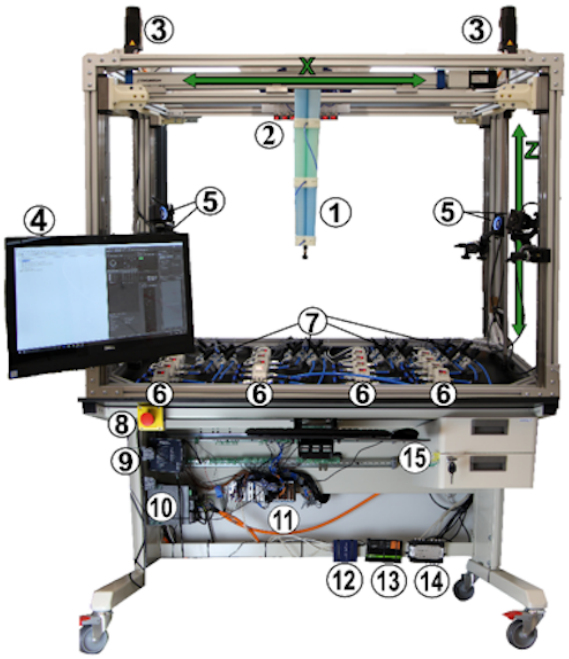

Figure 15: Soft robotics evaluation platform with all necessary equipment for monitoring and controlling the SPL. (1) the SPL with a suction cup end-effector, (2) digital pressure sensors, (3) DC servo-motors that govern motion of the vertical and horizontal linear stages, (4) graphical user interface on a computer system, (5) motion capture system with six wide-angle cameras, (6) sixteen high-speed pressure regulators, (7) thirty-two fast-switching, high-flow solenoid valves, (8) emergency stop button, (9) power supplies, (10) DC servo-motor drivers, (11) real-time National Instruments Compact-Rio controller, (12) ethernet adapter, (13) 8-port Gigabit PoE/PoE+ switch for motion capture cameras, (14) eSync 2 box to integrate motion synchronize data with the NI system, and (15) keyboard and mouse for use with computer.

The low-level architecture is executed through a soft robotics evaluation platform, presented in Figure 15, that is equipped with 32 solenoid valves, 16 pressure regulators and 16 pressure sensors, providing enough equipment for full control of the limb, as well as any pneumatically-actuated end-effectors. A pneumatic supply with a constant pressure of $0.413MPa$ (approximately $60psi$) is connected to the pressure regulators on the evaluation platform. Design of the RRAs only allows each actuator to be safely pressurized up to $60psi$ before failure can occur, therefore this pressure was chosen as the upper limit for safe manipulation of the limb. The pneumatic devices on the evaluation platform are

regulated by a real-time controller (CompactRIO, National Instruments Inc., Austin, TX) executing motion logic developed in LabVIEW program software.

3.2.1 Mode Selection

The low-level architecture allows for gross motion of the SPL, as well as fine motion through the control of individual 3CA segments. The default mode of control simultaneously manipulates all segments of the limb, however, the user is capable of changing modes, via the user intent-detecting modality (see section 3.3, High-Level Architecture), to individually control each segment for precise positioning of the end effector. The user can cycle between the modes in both forward and backward directions. For example, the forward direction cycles from the entire limb, to the first (proximal) segment, through the third segment, and back to control of the entire limb. The backward direction goes in the opposite order: the entire SPL, to the third (distal) segment, through the first segment and back to the entire SPL. This mode switching, however, is not limited in a single direction, as the user can move forward and backward between modes of control. For example, the mode can be changed from entire limb, to the first segment, to the second segment, back to the first segment, back to the entire limb, then to the third segment.

3.2.2 End Effector Mapping

To control the SPL, a pneumatic control scheme was developed in LabVIEW. The potential motion was discretized into a 5x5 motion controller, shown in Figure 16, which represents 25 reachable end effector positions within the workspace of the SPL when it is projected forward from the hip. The highest vertical position, distinguished as “0x +2Y” in dark gray in Figure 16, was determined by the maximum pressure that the actuators

could safely operate (approximately 55*psi*). Since the limb's center of gravity is highest at this point, and therefore affected most by gravity, this was chosen as the baseline position to map the other fifteen outer-most end effector positions, also highlighted in dark gray, to create a similar bending profile around the base of the SPL. The limb is depressurized at the center-most position, "0X 0Y", shown in white. The reachable inner locations, displayed in light gray in Figure 16, were mapped so that the end-effector was positioned at approximately half the distance as the outer-most position in each direction. For example, when the SPL is positioned at "+1X +1Y", it is approximately half the distance between the depressurized state and the "+2X +2Y" position. The chamber pressures needed to move the arm to each location are listed in Table 1. Similar pressures are delivered to the respective chambers in control modes of individual segment, or precise limb control.

-2X +2Y	-1X +2Y	0X +2Y	+1X +2Y	+2X +2Y
-2X +1Y	-1X +1Y	0X +1Y	+1X +1Y	+2X +1Y
-2X 0Y	-1X 0Y	0X 0Y	+1X 0Y	+2X 0Y
-2X -1Y	-1X -1Y	0X -1Y	+1X -1Y	+2X -1Y
-2X -2Y	-1X -2Y	0X -2Y	+1X -2Y	+2X -2Y

Figure 16: Reachable positions of the 5x5 discretized motion controller. The dark gray positions represent the outer-most points of the workspace, based on the maximum pressure that the actuators can withstand in each direction. The light dray positions represent the halfway point between the outer-most positions and deflation. The SPL is deflated within the white square (center).

End Effector Position	Pressure in Each Chamber (*psi*)								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
0X 0Y	0	0	0	0	0	0	0	0	0
0X +2Y	0	55	45	0	55	47.5	0	40	40
+1X +2Y	0	55	30	0	55	40	0	40	40
+2X +2Y	0	50	30	0	55	35	0	40	30
+2X +1Y	5	55	30	5	55	25	5	45	45
+2X 0Y	15	50	30	15	30	15	15	45	15
+2X -1Y	30	35	30	35	45	0	30	35	0
+2X -2Y	40	35	0	50	40	0	40	35	0
+1X -2Y	40	20	0	50	30	0	40	20	0
0X -2Y	40	0	0	50	0	0	40	0	0
-1X -2Y	40	0	35	50	0	40	40	0	35
-2X -2Y	30	0	35	35	0	50	30	0	40
-2X -1Y	0	20	35	15	20	50	0	15	35
-2X 0Y	0	30	50	0	25	55	0	20	40
-2X +1Y	7.5	30	50	7.5	45	55	7.5	50	45
-2X +2Y	0	30	50	0	50	55	0	40	40
-1X +2Y	0	40	50	0	50	50	0	40	40
0X +1Y	0	35	25	0	40	40	0	45	35
+1X +1Y	0	50	30	0	37.5	30	0	30	30
+1X 0Y	7.5	50	20	7.5	25	5	7.5	25	5
+1X -1Y	20	17.5	0	25	20	0	20	17.5	0
0X -1Y	20	0	0	25	0	0	20	0	0
-1X -1Y	20	0	25	25	0	20	25	0	25
-1X 0Y	0	30	50	0	17.5	30	0	15	27.5
-1X +1Y	0	40	50	0	35	35	0	20	25

Table 1: Pressure values in each RRA chamber to create bending motion to each position within the 5x5 motion controller. End effector positions are colored coded to match Figure 16.

3.2.3 Reduction of Dynamic Instabilities

The movement of the SPL between output locations appeared dynamically instable due to the nature of the fast-switching solenoids. Once each chamber was fully pressurized to its desired pressure, the limb would oscillate around the target end effector position. This was especially apparent as the limb moved across the furthest opposing positions of the workspace. This oscillation was combated by a "stepping" algorithm within the control

logic, shown in Figure 17. This algorithm divides the total change in pressure for each RRA by an iteration variable, *i*, representing the number of user-defined steps, to move the limb between desired output positions. The result is a uniform pressure change in each RRA for every step (i.e. *i)*, to move the SPL to a position in the workspace as defined by the user. Each step has an experimentally calibrated delay of 200*ms*, which presents a tradeoff between actuation speed and oscillation around the end position. For example, if the number of iteration steps is set to ten, the SPL moves between positions across a two second window. However, due to additional delays for system stability, a single movement sequence takes closer to three seconds to fully execute. Movement of the limb to subsequent positions commences once all steps for the prior movement have executed. This stepping algorithm is shown in Figures D1 and D2 in Appendix D.

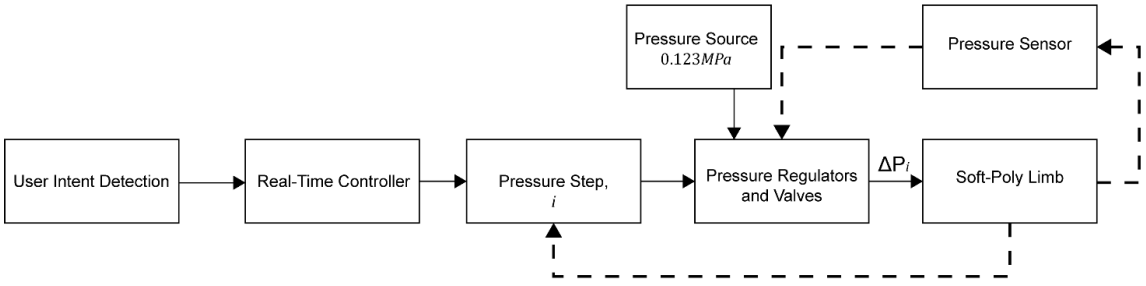

Figure 17: Process to reduce SPL oscillation during periods of actuation. Based on detected user intent, desired position of the SPL is processed by the real-time controller's 5x5 discretized control logic to position the limb end effector by a change in pressure within each RRA. Change in desired end effector position, and subsequent RRA pressures, is accomplished through *i* number of steps, as defined by the user. For each step, the RRAs are pressurized by an amount ΔP_i every 200*ms*.

3.3 High-Level Architecture

User intent detection methods are explored to enable operation of the SPL at a higher level of control. User control is demonstrated through three user intent detection modalities, namely an analog joystick (OM300B-M2, Yueqing Omter Electronic &

Technology Co., Ltd., Wenzou, China), an inertial measurement unit (IMU) sensor (BNO055, Bosch, Broadview, IL), and a surface electromyography (sEMG) sensor (MyoWare AT-04-001, Advancer Technologies, Raleigh, North Carolina), as shown in Figure 18. User-desired motion of the SPL is captured via one of the three modalities and processed through the 5x5 discretized motion controller and low-level pneumatic architecture, shown in Figure 19. Each output position within the motion controller implies a set of bounds for the desired intent-sensed positions, therefore any user-desired position that falls within these bounds will move the limb to one of the 25 pre-defined positions. These three modalities were pursued because they offered easy manipulation for highly variable outputs. Once the high-level architecture was properly characterized with the joystick, more innovative methods of user intent detection were explored.

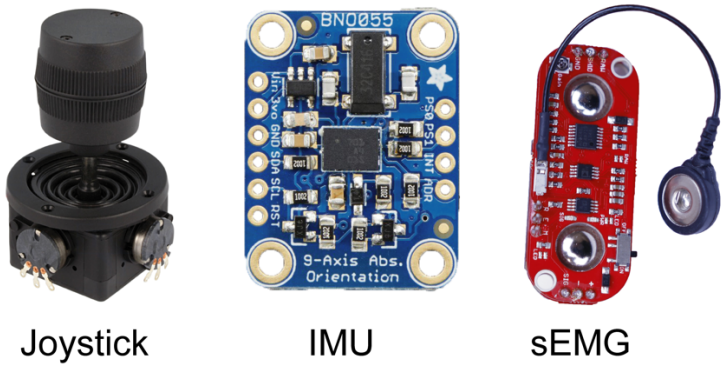

Figure 18: Sensors used to detect user intent to control the SPL. The high-level architecture was first explored using the joystick, then improved using IMU and sEMG methods.

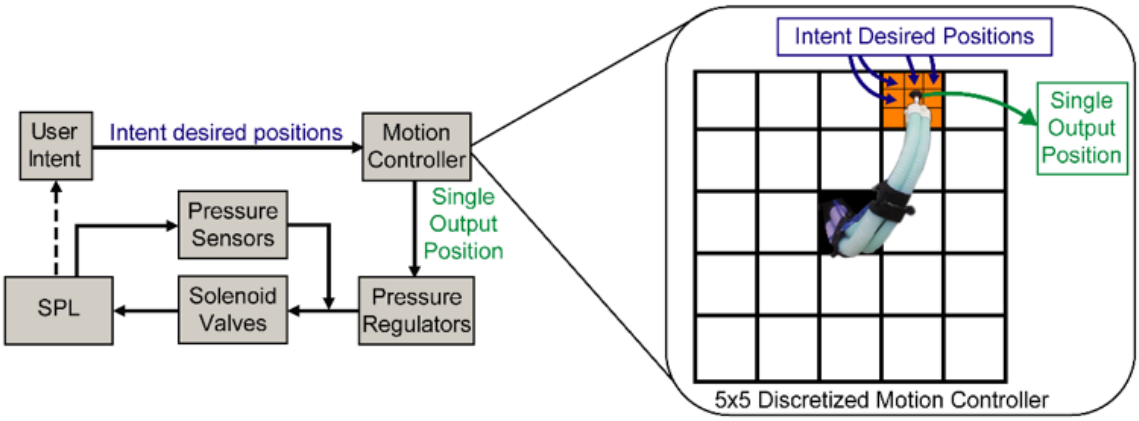

Figure 19: The system control flow diagram. User intent is captured via one of the following systems: joystick, IMU and sEMG. User-desired position of the end effector is interpreted by the 5x5 discretized motion controller, correlating possible desired positions within a set of bound to a single output position. Any intent-sensed positions within the orange box moves the end effector to one of the 25 pre-defined positions.

3.3.1 Joystick

High-level control of the SPL was first investigated using a three-axis analog joystick. This was chosen as the initial method for mapping user intent to the low-level control as it offered a simple approach for detecting user intent. User intent detection via the joystick sensing setup is measured by a change in joystick position along its three degrees of freedom: X, Y, and Z (twist). Desired position of the SPL end effector is achieved by moving the joystick along its X and Y axes. In addition to gross motion, twisting the joystick around the Z-axis switches to the individual segment control mode, allowing for finer control of each segment.

An electronics box was designed and 3D-printed to house the joystick and necessary electrical components, as seen in Figure 20, for easier handling of this system. This device was connected to the soft robotics evaluation platform via analog input and used to control the arm through the real-time controller. The LabVIEW control code for the joystick is displayed in Appendix D. While the joystick provided a rudimentary method

of detecting user intent, it restricts the use of a hand to perform tasks simultaneously with the SPL.

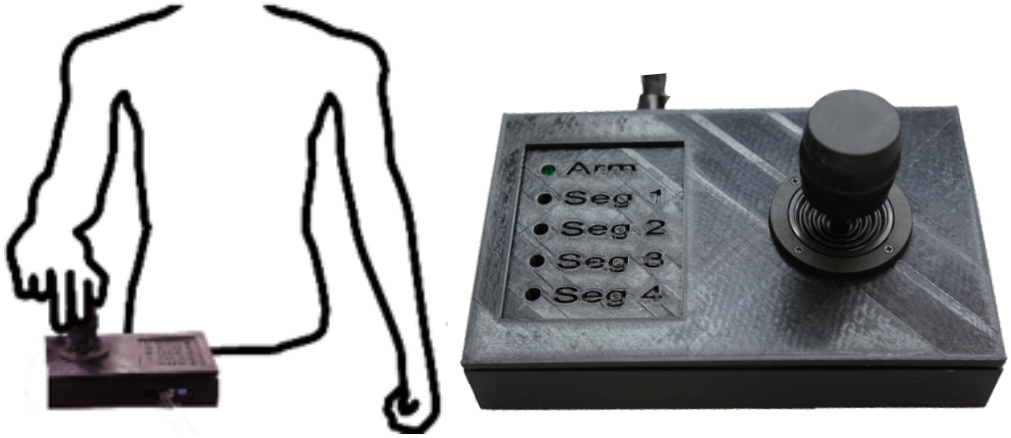

Figure 20: User intent detection via analog joystick is sensed by a change in position along the X, Y, and Z axes. The final box design is on the right.

3.3.2 Inertial Measurement Unit (IMU)

To expand upon the capability of sensing user intent, a system using an inertial measurement unit (IMU) was developed by incorporating the sensor on the dorsal side of a glove, as shown in Figure 21. The IMU sensor incorporates an accelerometer, gyroscope, and magnetometer to measure the acceleration along three axes, the rotation around three axes, and direction in space in respect to the Earth's magnetic field. Rather than a change position along an X, Y, and Z axes, user intent detection via IMU is measured by a change in pitch and yaw rotational angles in addition to tactile buttons. The intent-sensed position of the SPL is derived from the deviation in pitch and yaw angles from a calibration position. Individual segment control mode through this modality is achieved by pressing tactile buttons on the side of the glove near the first metacarpophalangeal (MCP) joint.

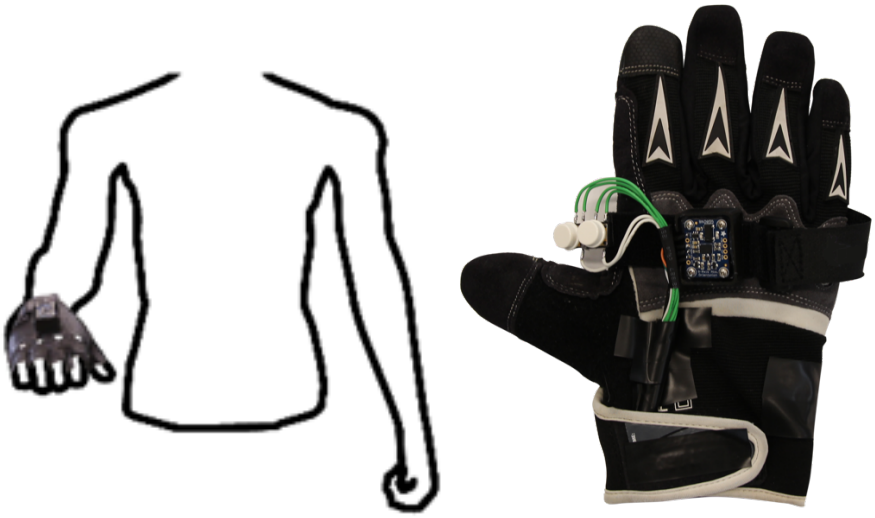

Figure 21: User intent detection via IMU is sensed by a change in pitch and yaw rotations as well as tactile buttons near the first metacarpophalangeal (MCP) joint. The prototyped glove with IMU sensor and tactile buttons is on the right.

The user-intended control of the SPL via IMU is processed by a microcontroller (*Arduino Mega 2560*, Arduino LLC., Italy), which outputs the user-desired position to the real-time controller via serial communication. The Arduino and LabVIEW control codes for the IMU system are shown in Appendixes B and E, respectively. Similar to the joystick, this modality also limits the use of a hand during control of the SPL. However, the glove design allows the user to grasp objects while concurrently manipulating the SPL. Although this device was developed for control using the hand, this same principal can be employed by other parts of the body that can produce two-dimensional rotation, such as the foot or the head.

3.3.3 Surface Electromyography (sEMG)

A hands-free approach for controlling the SPL was explored through the use of two surface electromyography (sEMG) sensors. An sEMG sensor non-invasively measures the electrical activity produced by the body's skeletal muscles. User intent detection via the sEMG sensing setup is measured by a change in myoelectric impulses from two sEMG

sensors, one on each bicep brachii muscle. The first sEMG sensor (EMG 1) controls pressurization of the SPL, while the second sEMG sensor (EMG 2) dictates the direction of the SPL motion, as seen in Figure 22.

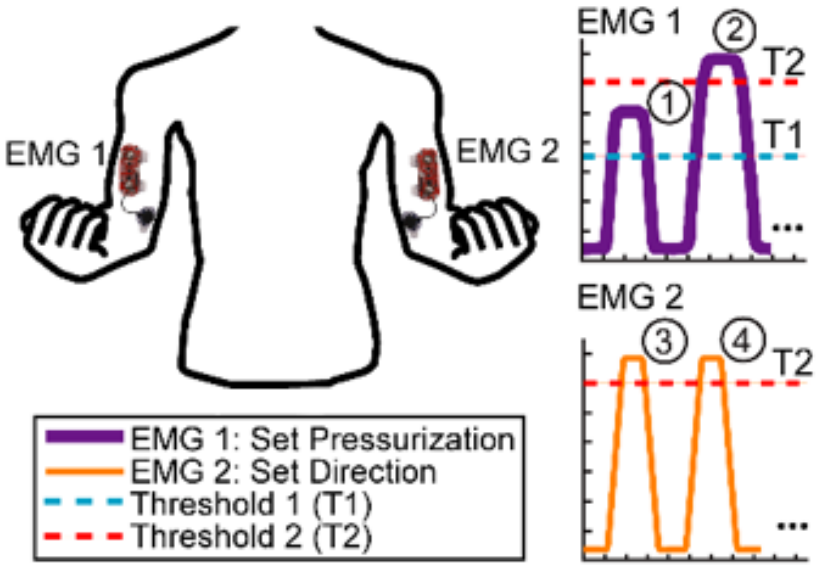

Figure 22: User intent detection and control via sEMG. Pressurization of the SPL is accomplished by EMG 1 and direction of gross motion is achieved by EMG 2. **(1)** EMG 1 output between T1 and T2 pressurizes the SPL to half of its maximum pressure in the set direction of motion. **(2)** EMG 1 output above T1 pressurizes the SPL to its maximum pressure in the set direction of motion. EMG 1 output below T1 depressurizes the SPL. **(3)** EMG 2 output above T2 sets the SPL direction of motion 45° counter-clockwise in the coronal plane (facing the user). **(4)** A second EMG 2 output above T1 adds another 45° counter-clockwise (i.e. to a direction of 90°). Direction of motion can be set in 45° increments around the entire 360° of movement.

From the trend graph of EMG 1 (purple line), if the EMG 1 signal is between threshold 1 (T1) and threshold 2 (T2), as seen in Figure 22(1), then the SPL is pressurized to half of its maximum capacity in the set direction. These half capacity pressures position the joystick to the locations in light gray in Figure 16. If the signal is above T2, as seen in Figure 22(2), then the SPL is lifted to its maximum capacity in the set direction. These maximum capacities are represented by the dark gray positions around the outer-most

area of the discretized motion controller in Figure 16. Finally, if the signal is below T1, the SPL is depressurized entirely. From the trend graph of EMG 2 (orange line), if the signal is above T2 (Figure 22(3)), the SPL is triggered to rotate counter-clockwise in the coronal plane (facing the user) by 45°. Successive activations of EMG 2 will cause the SPL to step through its workspace at 45° intervals. Figure 22(4) shows a second successive activation, which causes the arm to move to the 90° direction from its prior position in the 45° direction.

Similar to the IMU system, the user-intended control via sEMG sensors is processed by a microcontroller and sent to the real-time controller via serial communication. The Arduino and LabVIEW control codes for the sEMG system are shown in Appendixes C and F, respectively. Unlike the previous user intent detection modalities, the sEMG approach permits use of hands while controlling the SPL. This modality is not limited to control by the biceps and can be applied to other muscles of the body. For example, the trapezius (neck) or medial gastrocnemius (calf) muscles could be utilized by tetraplegic individuals.

Chapter 4

SYSTEM EVALUATION

4.1 Introduction

There are three primary focuses in terms of testing the user intent detection and control of the SPL. The first is to determine the accuracy of mapping between the desired intent-sensed position, measured from the user intent devices, and the end-effector position. The second is to run experiments using the control structure to determine the workspace capabilities and payload capacity of the SPL. The third is to perform various interaction tests to validate that the user is capable of controlling the SPL to perform a variety of tasks.

4.2 Mapping Accuracy Test

A mapping accuracy test was performed to determine the precision of the mapping scheme between the user-desired positions and end effector positions, specifically from the analog joystick and IMU glove modalities, and output position of the SPL end effector. The sEMG modality of user intent detection was not evaluated through this approach as it was not fully developed during the timeframe of this thesis. However, its preliminary control is explored in the interaction tests (see section 4.5.1).

To perform this experiment for the two modalities (independent of one another), a motion capture system (Optitrack Prime 13W, NaturalPoint Inc., Corvallis, OR) is used to track the three-dimensional position of a passive reflective marker placed on the distal end of the SPL. The limb is positioned at the 16 outer-most reachable locations (shown in dark gray in Figure 16), while the intent-sensed signal is concurrently recorded. Position of the joystick is tracked in the first iteration of the experiment, while rotation of the IMU

is recorded in the second iteration of the experiment. The X and Z positions of the end effector are recorded to compare the horizontal and vertical motion in the coronal plane (facing the user) with the two-dimensional motion of the joystick and IMU systems. Three trials are performed for each modality, and the desired intent-sensed position and end effector output position are averaged and normalized for each of the 16 locations. The root-mean-square error (RMSE) between the normalized average intent-sensed signal and end effector position is calculated.

4.3 Workspace Test

A workspace assessment experiment is performed to analyze the capability of the SPL to maneuver about its mounted base. In this experiment, the SPL is mounted on a back frame and worn by a human mannequin, so that the limb projects forward from the mannequin's hip. Using the analog joystick as the user intent control system, the soft limb is manipulated to each of the 25 fixed output locations in the 5x5 motion controller, which covers the reachable 3D workspace of the limb. A motion capture system is used to record the position of the SPL at the interconnecting points of each segment. These three-dimensional positions are plotted against one another, and the workspace volume, maximum reach, horizontal range, and vertical range of the SPL are calculated.

4.4 Payload Test

To evaluate the SPL's payload, or load bearing capabilities, three experiments are performed. The first two experiments are performed with the limb fully extended and parallel to the ground, so that gravitational forces act orthogonal to its body. This configuration requires the maximum torque to be exerted upon the SPL. The distal end of the SPL is placed under the upper load cell of a universal testing machine (Instron 5944,

Instron Corp., High Wycombe, United Kingdom) in the first experiment, highlighted in Figure 23A. Using the low-level control architecture, the bottom two of the three RRAs in all segments are pressurized until the limb revealed torsional effects. Five trials were performed, and the exerted force was averaged. In the second experiment, a set of loads between 0.05 to 0.35*kg*, at intervals of 0.05kg, were placed at the end-effector location of the SPL and lifted vertically from the depressurized position to a position parallel to the ground, as shown in Figure 23B. The pressures in each chamber, to lift the set of loads, are recorded for the entire SPL. A third experiment is performed to further verify the effective load bearing capacity of the SPL, in which the user employs a joystick controller to pick, move, and place the same set of loads (0.05 to 0.35*kg*) into a rectangular box (0.11x0.13x0.55m) positioned 0.95 m away.

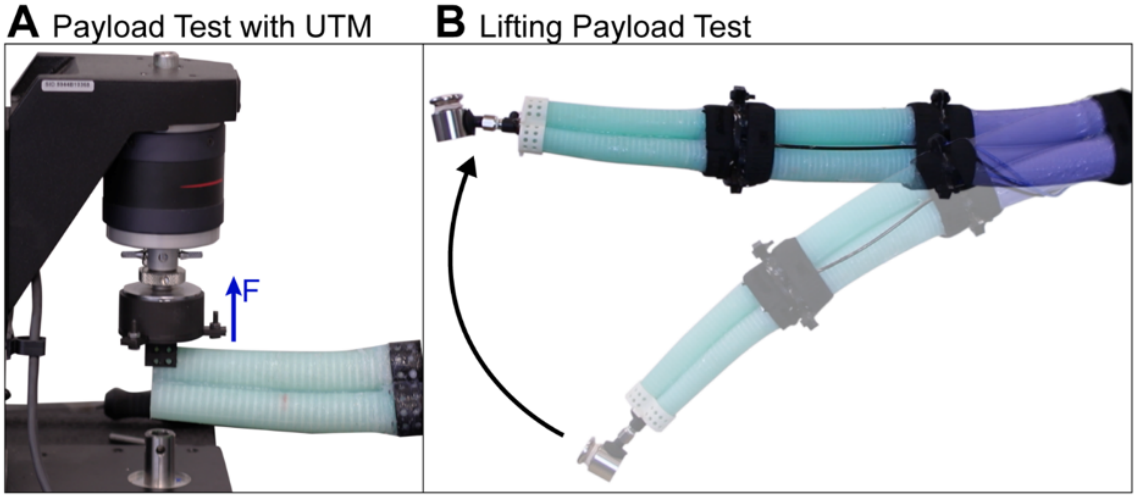

Figure 23: Payload tests. **(A)** The distal end of the SPL is placed underneath the load cell of a universal testing machine (UTM). A unique mounting adaptor was created to distribute the lifting force at the tip of the SPL across the body of the anvil connected to the load cell. The bottom two chambers were inflated for all segments until the arm showed any sign of torsion. **(B)** The SPL's effective payload was tested by lifting a set of weights between 0.05 to 0.3kg, from a deflated position to the position of maximum torque (straight and parallel to the ground).

4.5 Interaction Tests

To further investigate the user controllability and versatility of the modular SPL, various interaction tests are performed. These tests include investigation of the SPL control to perform hands-free operation with sEMG sensors, pick and place testing with common household objects, whole arm grasping techniques, multi-tasking scenarios, and tasks in overhead mounting positions. For these tests, a vacuum suction cup is used as the limb's end-effector. The suction cup manipulator unit's theoretical holding load is $0.86kg$; it is connected to a vacuum pump capable of facilitating depressurization rates of $1.42x10^{-3}$ m^3/s. Performance of each task is recorded for qualitative analysis to therefore investigate the user capability to control the SPL to perform a variety of tasks.

4.5.1 Surface Electromyography (sEMG)

For control of the SPL via sEMG sensors, a similar protocol to the mapping accuracy experiment is executed but the end-effector position is not recorded using a motion capture system. Additionally, a sample motion sequence is performed to display potential motion of the SPL using sEMG sensors. The goal of these tests is to assess the user's ability to control the SPL through a hands-free method.

4.5.2 Pick and Place Test

A pick and place experiment is designed to assess the capability of the user to operate the SPL to grasp objects placed inside of a cup with a radius of $0.032m$ and a height of $0.11m$. Using the analog joystick, the objects are moved across a table into a rectangular box ($0.11x0.13x0.55m$), similar to the third payload experiment. A variety of daily living objects are used to assess the limb's capability to grasp and place objects in a precise location.

4.5.3 Whole Arm Grasping Test

In this experiment, the limb's inherent compliance is tested by performing underactuated grasping maneuvers. Due to its soft continuum design, the SPL is capable of performing whole arm grasping, similar to the trunk of an elephant. Whole arm grasping is a technique where the entire body of the SPL is used to wrap and grasp a variety of objects of different shapes and sizes. The joystick controller is used to manipulate the SPL to hold different weighted objects with dissimilar material texture, sizes, and shapes. The weight of each subsequent object is increased to discover the payload capacity for this method of grasping.

4.5.4 Multi-Tasking Tests

Another set of experiments are performed to evaluate the ability of the SPL to enable users to perform numerous tasks in parallel using both the robotic limb and their biological limbs. The goal of these experiments is for the user to hold a box, scan an access card, and enter a room in various ways using all available limbs. In addition, the potential of the SPL for supporting a user with visual impairments is explored.

4.5.5 Variable Mounting Test

The SPL's variable mounting positions on the back frame and the subsequent modification of the limb's operating range is explored in a final evaluation assessment. Through manual control of the low-level architecture, the performance of the SPL is explored when mounted near the user's shoulder joint for overhead task assistance. The SPL is mounted $0.45m$ above the waist for this experiment, with the end effector pointed upwards in the depressurized resting position. Mounting the limb in this position

translated the limb's workspace from around the right side of the user's chest to around and above the user's head to perform tasks in the respective area.

Chapter 5

RESULTS AND DISCUSSION

5.1 Mapping Accuracy Test

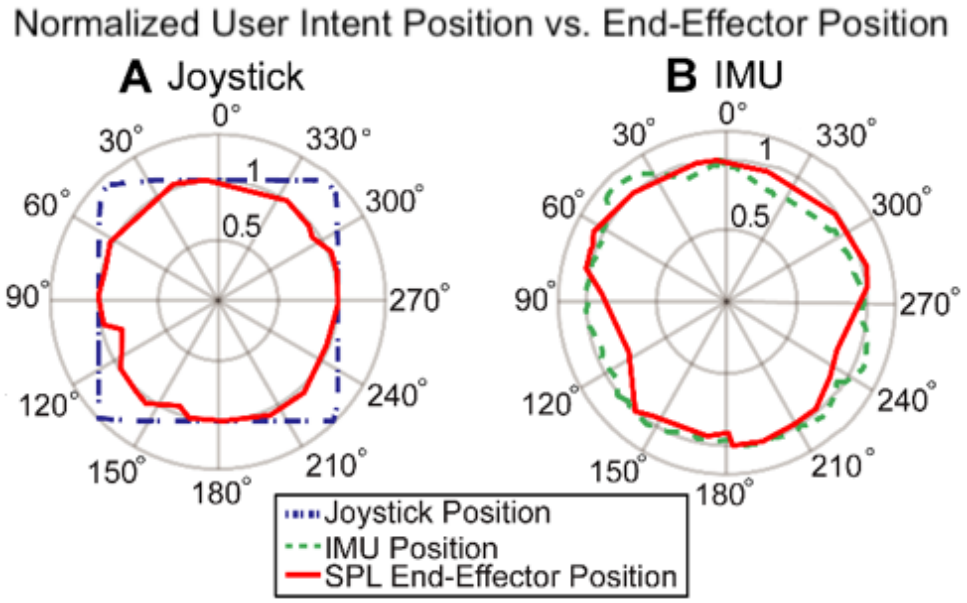

Figure 24: (A) The normalized X-Y position of the joystick (dashed blue line) and resulting position of the SPL end effector (red line) around the outer locations of the workspace. **(B)** The normalized observed IMU position is tracked (dotted green line) and compared to the SPL motion (red line).

The normalized user intent positions versus SPL end effector positions is presented in Figure 24. We expected the end effector to produce a circular motion as each user intent device was manipulated around the outer-most reachable locations. This expectation was consistent with experimental results from the mapping test of the joystick and IMU systems as shown in Figure 24A and Figure 24B, respectively.

The root-mean-square error (RMSE) between the normalized horizontal (X) positions of the joystick and end effector is 0.169, while the error between the normalized vertical (Y) positions is 0.159. The observed square-shaped joystick position pattern in

Figure 24A is a result of its elementary, two axis design. To put in perspective the effect of the varying figure shapes, the RMSE value between the four diagonal positions is 0.253 along the normalized X-direction and 0.277 along the normalized Y-direction. In comparison, the RMSE value between the four relative positions (up, down, left, and right) is 0.057 and 0.024 in the normalized X- and Y-directions, respectively. Despite the different shaped signal paths, the joystick successfully articulates the end effector to the user-desired positions. The RMS error between the IMU and end effector is 0.099 along the horizontal axis and 0.088 along the vertical axis. The circular motion produced by the user's hand, thus the IMU, resulted in the end effector of the SPL following a similar circular motion, as seen in Figure 24B.

5.2 Workspace Test

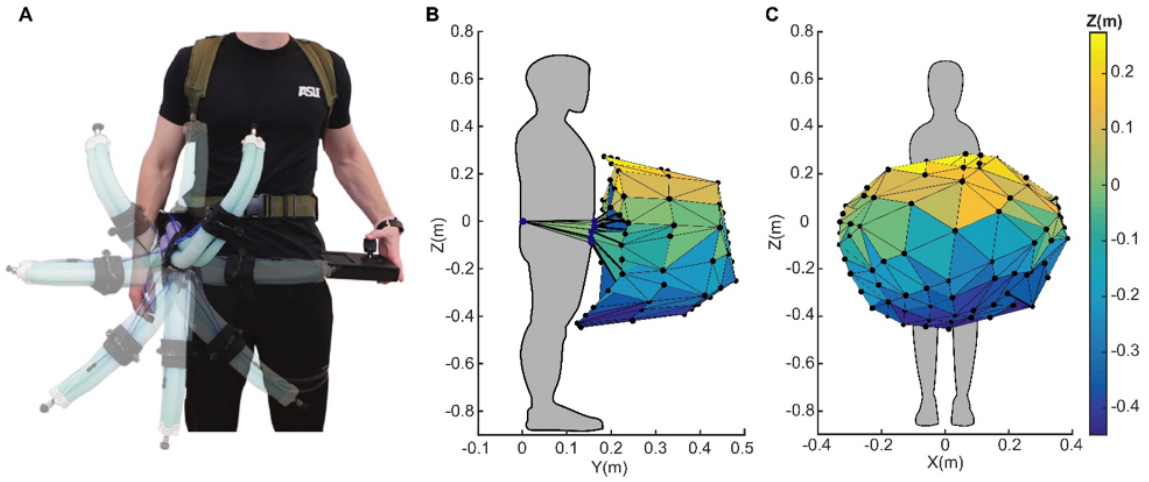

Figure 25: Workspace of the SPL. **(A)** A user manipulating the SPL through its maximum workspace when mounted at the waist**. (B)** View of the SPL' workspace in the sagittal plane. The maximum reach and vertical range of the SPL are 0.55m and 0.72m, respectively. **(C)** View of the SPL's workspace in the coronal plane. The horizontal range of the SPL is 0.69m.

The workspace of the SPL is illustrated from the side and front views in Figure 25B and Figure 25C, respectively. The total workspace was achieved through linear

extrapolation of the 25-reachable end effector positions in three dimensions. The volume of the SPL's measured workspace is calculated to be approximately $0.08m^3$. In addition, the maximum reach, horizontal range, and vertical range of the SPL is 0.55m, 0.69m and 0.72m, respectively. Shown in Figure 25A, the SPL moves closely around the user's body, however does not make contact with the user as it traverses around its workspace.

5.3 Payload Test

Payload Weight	Pressures in Each Chamber (*psi*)					
	C2	**C3**	**C5**	**C6**	**C8**	**C9**
0kg	40	40	15	15	10	10
0.05kg	42	42	17	17	12	12
0.1kg	45	45	19	19	14	14
0.15kg	47	47	21	21	16	16
0.2kg	49	49	23	23	18	18
0.25kg	51	51	30	26	22	22
0.3kg	53	53	30	26	22	23

Table 2: Pressures in each chamber of the SPL to lift the limb parallel to the ground with various payloads in the second payload experiment.

At the maximum reach of the SPL, i.e. $0.55m$, the limb exhibits an average payload capacity of $9.43N$. The amount of force is comparable to carrying a weight of nearly 1kg at the distal end of the SPL. With a suction cup manipulator as an end effector, the maximum effective payload is found to be 0.3kg. The pressure values required to lift the set of loads in the second experiment are documented in Table 2 for the bottom two chambers of each segment. During the first payload experiment, it is determined that movement of the SPL's end effector position is constrained to motion along a single plane, and subsequently exerts force only in the upward direction. On the other hand, during the second payload experiment, the limb lifts the set of loads in an unconstrained motion. The weight of the SPL effects this spatial motion by introducing torsional forces along its body that cause

instabilities as the limb to twists around its center axis. From the third experiment, it is demonstrated that the SPL is able to successfully transfer up to 0.35kg of load across its horizontal workspace. During this dynamic test, the SPL was subject to torsional forces along its body when carrying loads above 0.3kg, similar to the instabilities presented in the second payload experiment. While some motion oscillations are present, the SPL is still capable of accomplishing these tasks.

5.4 Interaction Tests

5.4.1 Surface Electromyography (sEMG)

Prior to testing the controllability of the SPL with the sEMG detection modality, the second modular segment failed, and a different segment was used, therefore the limb could not produce end effector positions similar to that of the mapped system. While these positions could not be achieved, the user is capable of switching between pressurization levels and directions of movement to control the SPL as the sEMG modality was designed.

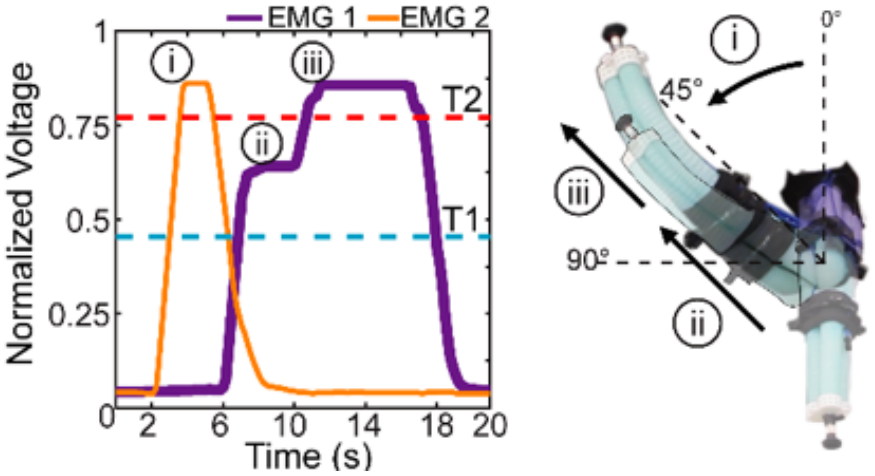

Figure 26: Example control sequence using sEMG modality user intent detection.

An example SPL motion is shown in Figure 26. Displayed in Figure 26(i), an EMG 2 activation above T2 sets the uninflated SPL's direction of motion along the 45° plane. An

EMG 1 activation between T1 and T2 pressurizes the SPL to half of its maximum pressure in the 45° direction, as portrayed in Figure 26(ii). Finally, an increased EMG 1 output above T2 pressurizes the SPL to its maximum pressure in the 45° direction and held in position while EMG 1 is greater than T2, shown in Figure 26(iii). The release of EMG 1 below T1 depressurizes the SPL to its initial, deflated position.

5.4.2 Pick and Place Test

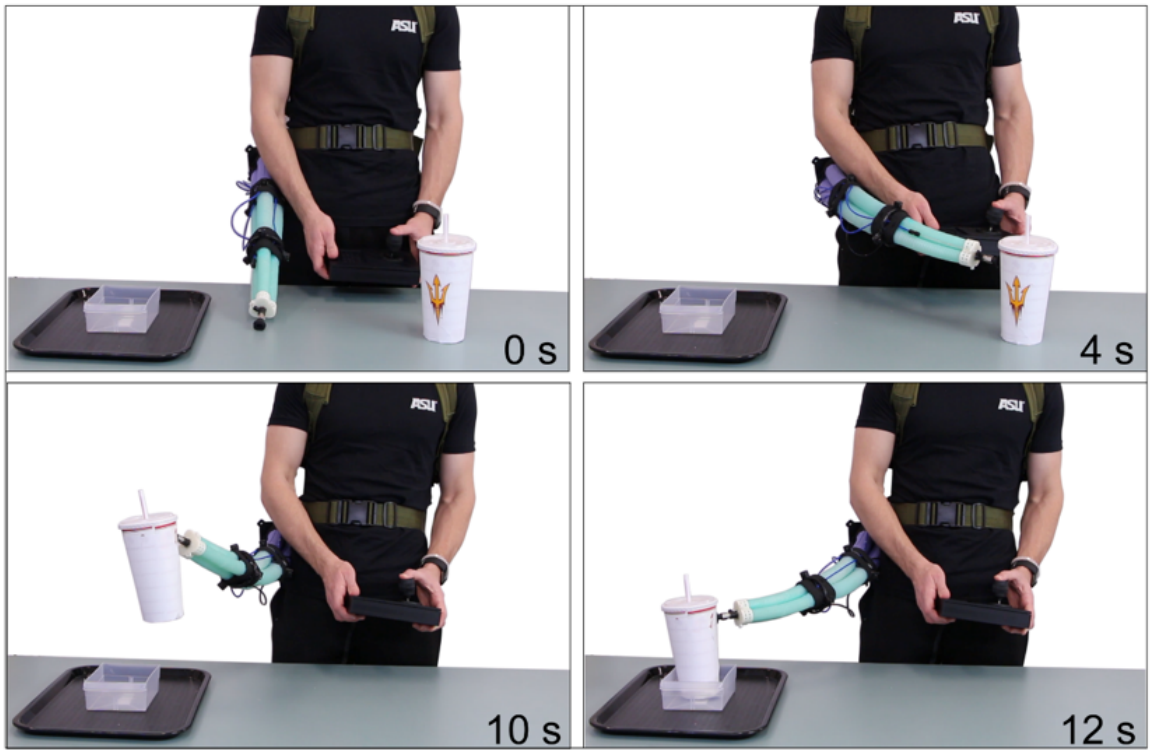

Figure 27: A series of frames depicting the pick and place experiment. The SPL is shown picking up a 0.13kg cup, transferring it across the workspace, and placing it into a small box.

The pick and place task, achieved by actuating the SPL using the joystick user-intent detection modality and suction cup end effector, successfully transferred a variety of daily living objects. These objects included a fork (0.015kg), a cup (0.13kg), an apple (0.15kg), and a banana (0.18kg). Movement of the arm was smooth during transport of

each object, and no motion oscillations were noted due to the light weight of the objects. Figure 27 depicts successful accomplishment of this task with a cup.

5.4.3 Whole Arm Grasping Test

Figure 28: The SPL performing whole arm grasping techniques to carry a ball (0.43kg), a box (0.83kg), and a bag (1.04kg).

The SPL's segments were pressurized to a maximum curvature angle of 192.3° to successfully grasp three objects of increasing weight using the entire body of the limb. These objects had different material texture, size, and shape, as seen in Figure 28, which included a soccer ball, a box, and a backpack, weighing 0.43kg, 0.83kg, and 1.04kg, respectively. By applying forces around the perimeter of the object and holding it against the user's body, the SPL is capable of holding objects up to approximately 3.8kg in weight. Using the whole body grasping strategy, the SPL demonstrates the capability of carrying objects nearly 2.35 times its own weight. It is noted that the SPL would not be able to carry such objects with the suction cup manipulator due to their size and weight. In comparison

to the suction cup end effector, utilizing the whole arm grasping technique increases the effective payload capacity of the limb by 1,158%.

5.4.4 Multi-Tasking Tests

Figure 29: Multi-tasking assistance using the SPL in two ways to accomplish the same task. **(A)** The user holds a box with both hands while using the limb to scan an access card. The SPL returns the card to the user then pushes the door open. **(B)** The user holds the box with SPL, using a whole body grasping technique. The hands are free to scan the access card and push the door open.

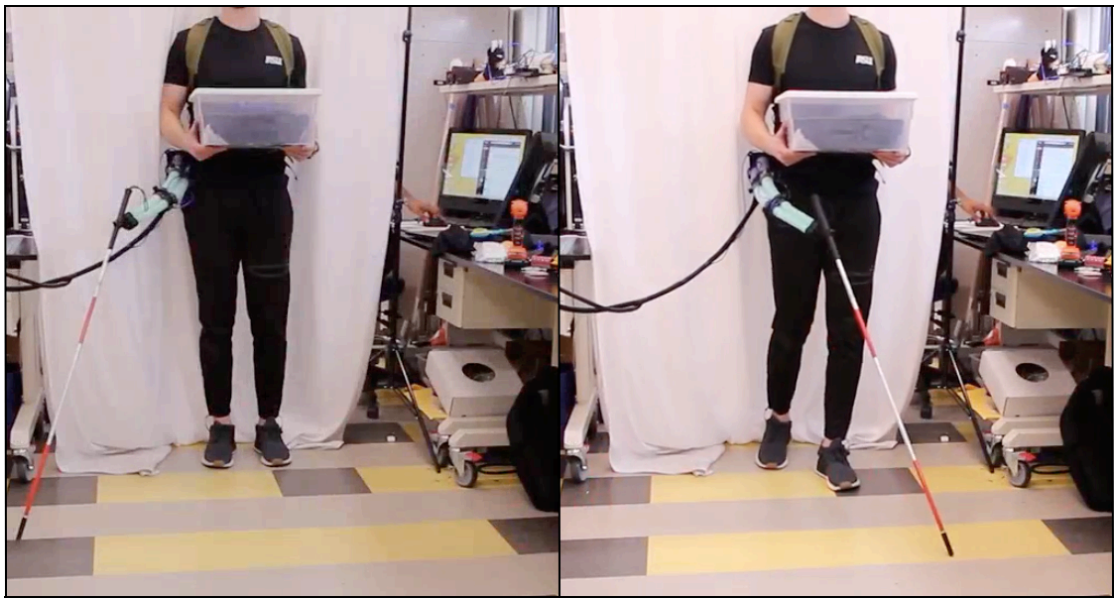

Figure 30: Multitask assistance using the SPL to support a visually impaired user with a walking cane. The cane is swung and tapped to provide object location and echolocation feedback, respectively.

The user is able to successfully perform multiple tasks using both biological and robotic arms. In the first multitasking scenario, seen in Figure 29, the user is able to employ their biological and robotic limbs in various approaches to carry a box, swipe an access card, and open a door to enter a room. In the first execution of this test, the user holds the box with their biological limbs, while utilized the SPL to gain access to the room, as presented in Figure 29A. In this first scenario, the user assigns the SPL to grasp, swipe and return the card, then extends the SPL to open the door. In the second execution of this task, as shown in Figure 29B, the whole body grasping technique is employed by the SPL to carry the box, while allowing the user to swipe the access card and open the door with their hands. This shows that the user is not only capable of performing multiple tasks at once but are also capable of executing the same set of tasks in numerous ways. In another multitasking scenario, the SPL demonstrates the ability to assist a user with visual impairments. In this scenario, the SPL is tasked with swinging a walking cane while the

user carries a box. At the end of each swing, the cane taps the floor in front of the user to provide echolocation feedback, as seen in Figure 30.

5.4.5 Variable Mounting Test

Figure 31: The SPL holding an umbrella (0.25*kg*) on its distal end to demonstrate potential for overhead manipulation.

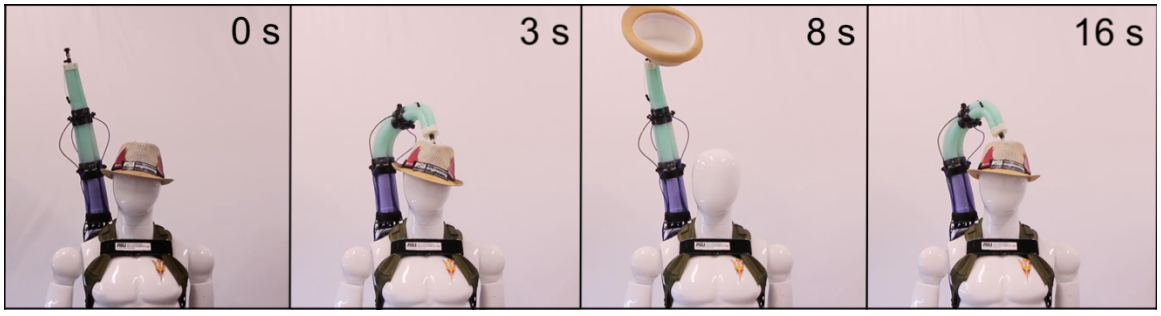

Figure 32: A series of frames captured showing the SPL grab a hat off the mannequin's head and delicately place it back on the head.

The SPL was capable of assisting users with overhead tasks through the use of low-level control architecture. Figure 31 highlights a demonstrated use case for overhead

manipulation, in which the SPL holds and umbrella to offer the user free use of their hands. In addition, the SPL successfully removed a hat from the mannequin's head and delicately returned it back in place, as seen in Figure 32. Overhead multitasking assistance, shown in Figure 33, is achieved by using the SPL to grasp an object from a shelf above the user's head and transport the object to the user, while using their biological limbs to perform other tasks. With the SPL mounted at different positions of the body, these experiments provide insight on the prospective development of systems that incorporate numerous SPLs along the body of a single user.

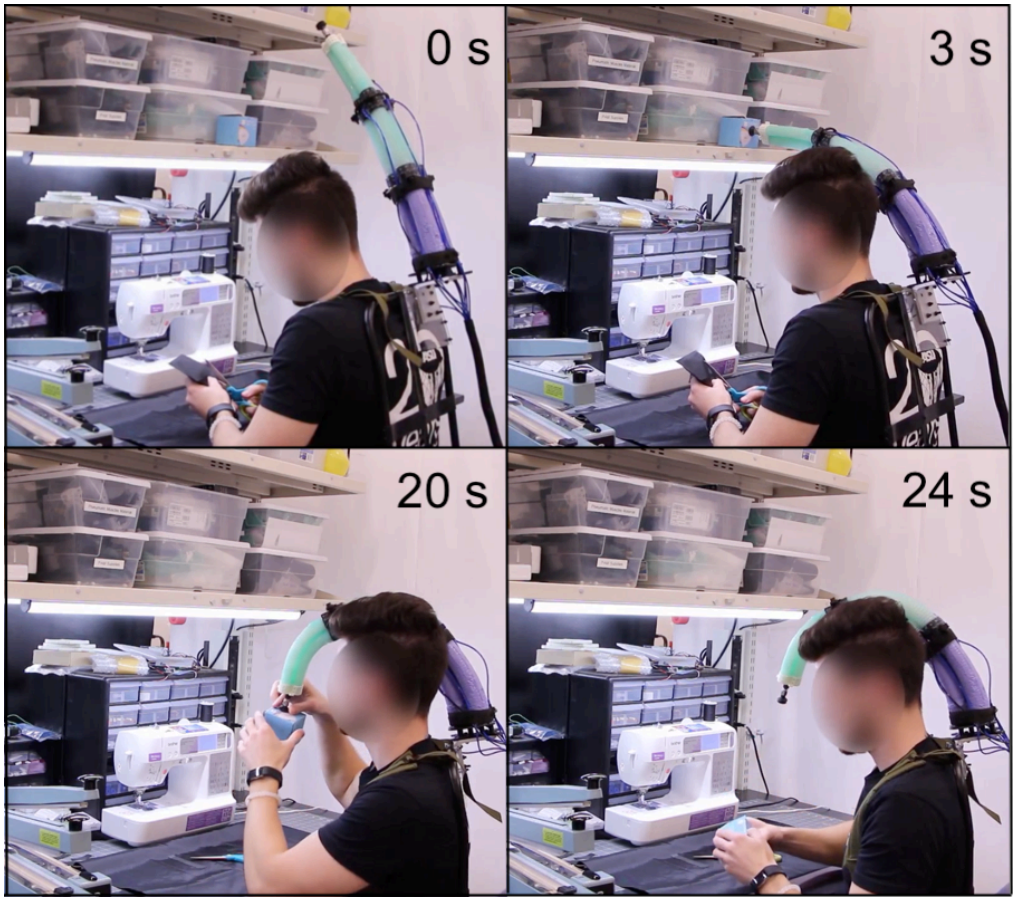

Figure 33: A series of frames captured showing the SPL performing multitasking assistance in an overhead position to grab a box of chalk off the shelf and hand it to the user.

Chapter 6

CONCLUSIONS AND FUTURE WORK

The goal of this research was to develop simple and effective means of controlling a Soft Poly-Limb to assist healthy and impaired users perform a variety of tasks, such as activities of daily living. Additionally, it was important to explore and integrate various forms of user intent detection. These aims were met by mapping the SPL's entire workspace across a 5x5 discretized motion controller using different user intent sensors, namely analog joystick, IMU, and sEMG.

The potential for performing multitasking assistance with the SPL was explored through a number of scenarios, such as unlocking and opening a door while the user held objects with their other hands, grabbing an object from an overhead shelf while the user worked on another task, and possibly assisting a visually impaired user with their folding walking cane in order to leave their hands free for other tasks. Through these demonstrations, the SPL was shown to be capable of assisting users execute daily tasks such as ADLS. While successful, an external user was utilized to control the SPL to perform these fairly complicated test scenarios. Using the joystick and IMU detection-sensing approaches, the user was able to achieve finer control of the SPL through individual control of each 3CA segment. These two sensing approaches, however, currently limit the user's ability to manipulate the SPL without sacrificing the use of a working limb, as their controls require the use of a hand. In contrast, the sEMG sensing approach offered a hands-free approach to control the limb, however, did not allow the user to position the end effector in as many discretized positions as the joystick and IMU systems. This control system serves as an initial proof of concept, and therefore holds room for optimization and improvement to achieve more robust user control.

Future direction for the control of the SPL includes improving and integrating one or more hands-free user intent detection approaches. This would enable the user to achieve genuine multitasking assistance using both biological and robotic limbs at their own volition, rather than requiring an external user to control the SPL. In addition, this would improve the control capabilities of users with upper limb impairment; while some users in this population are capable of manipulating a joystick, not all are able to perform fine motor movements and therefore require a hand-free method of controlling the SPL. Improving the sEMG approach is one potential method of achieving hands-free control. While EEG or neural interfaces require no physical movement of the body, detecting user intent with these methods requires complex data collection and processing methods, and also restricts the user's potential workspace due to the design of their tethered systems. On the other hand, sEMG offers both a simple technique for collecting and processing data. sEMG sensors do not require the user to perform complex motor tasks, and the output signal amplitude is easily adjusted based on the sensor gain. To maximize the human-robot interaction, the sEMG should measure from muscles that are not heavily used when performing tasks with the upper extremities (e.g. abdominal muscles), therefore allowing the user to perform tasks with the SPL and biological limbs in conjunction. The current sEMG control system requires the user to perform large muscle contractions, however the gain can be reduced to manipulate the arm through isometric muscle contractions. Detecting isometric contractions would allow the user to manipulate the SPL while holding objects. To further improve the sEMG system in its current state, additional methods or sensors are needed for fine control as well as control in more directions. Furthermore, various sensors could be incorporated into the SPL to create a closed-loop control system that adjustes the desired position of the SPL to the correct

position, regardless of the load being carried. Finally, continuous mapping between the user-desired signals and end effecter would allow for more precise positioning of the limb.

REFERENCES

Ajiboye, A. Bolu, Francis R. Willett, Daniel R. Young, William D. Memberg, Brian A. Murphy, Jonathan P. Miller, Benjamin L. Walter, et al. 2017. "Restoration of Reaching and Grasping Movements through Brain-Controlled Muscle Stimulation in a Person with Tetraplegia: A Proof-of-Concept Demonstration." *The Lancet* 389 (10081). Elsevier Ltd: 1821–30. doi:10.1016/S0140-6736(17)30601-3.

Ajoudani, Arash, Sasha-b Godfrey, Matteo Bianchi, and Manuel Catalano. 2014. "Exploring Teleimpedance and Tactile Feedback for IntuitiveControl of the Pisa/IIT SoftHand." *Ieeexplore.Ieee.Org* 7 (2): 1–12. doi:10.1109/TOH.2014.2309142.

Bassily, D., C. Georgoulas, J. Güttler, T. Linner, T. Bock, and T. U. München. 2014. "Intuitive and Adaptive Robotic Arm Manipulation Using the Leap Motion Controller." *Isr Robotik*, 78–84.

Bogue, Robert. 2009. "Exoskeletons and Robotic Prosthetics: A Review of Recent Developments." *Industrial Robot: An International Journal* 36 (5): 421–27. doi:10.1108/01439910910980141.

Charles, James P., Ornella Cappellari, Andrew J. Spence, John R. Hutchinson, and Dominic J. Wells. 2016. "Musculoskeletal Geometry, Muscle Architecture and Functional Specialisations of the Mouse Hindlimb." *PLoS ONE* 11 (4): 1–21. doi:10.1371/journal.pone.0147669.

del-Ama, Antonio J., Aikaterini D. Koutsou, Juan C. Moreno, Ana de-los-Reyes, ngel Gil-Agudo, and Jos L. Pons. 2012. "Review of Hybrid Exoskeletons to Restore Gait Following Spinal Cord Injury." *The Journal of Rehabilitation Research and Development* 49 (4): 497. doi:10.1682/JRRD.2011.03.0043.

Devroey, Christa, Ilse Jonkers, An de Becker, Gerlinde Lenaerts, and Arthur Spaepen. 2007. "Evaluation of the Effect of Backpack Load and Position during Standing and Walking Using Biomechanical, Physiological and Subjective Measures." *Ergonomics* 50 (5): 728–42. doi:10.1080/00140130701194850.

George Thuruthel, Thomas, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. 2018. "Control Strategies for Soft Robotic Manipulators: A Survey." *Soft Robotics* 0 (0): soro.2017.0007. doi:10.1089/soro.2017.0007.

Gopura, R A R C, Kazuo Kiguchi, and D S V Bandara. 2011. "A Brief Review on Upper Extremity Robotic Exoskeleton Systems." In *2011 6th International Conference on Industrial and Information Systems*, 8502:346–51. IEEE. doi:10.1109/ICIINFS.2011.6038092.

Guterstam, Arvid, Valeria I. Petkova, and H. Henrik Ehrsson. 2011. "The Illusion of Owning a Third Arm." *PLoS ONE* 6 (2). doi:10.1371/journal.pone.0017208.

Hochberg, Leigh R., Daniel Bacher, Beata Jarosiewicz, Nicolas Y. Masse, John D. Simeral, Joern Vogel, Sami Haddadin, et al. 2012. "Reach and Grasp by People with Tetraplegia Using a Neurally Controlled Robotic Arm." *Nature* 485 (7398). Nature Publishing Group: 372–75. doi:10.1038/nature11076.

House, B., J. Malkin, J. a. Bilmes, and J. Bilmes. 2009. "The VoiceBot: A Voice Controlled Robot Arm." *Proc. of the 27th Int. Conf. on Human Factors in Computing Systems*, 183–92. doi:10.1145/1518701.1518731.

Hussain, Irfan, Gionata Salvietti, Giovanni Spagnoletti, Monica Malvezzi, David Cioncoloni, Simone Rossi, and Domenico Prattichizzo. 2017. "A Soft Supernumerary Robotic Finger and Mobile Arm Support for Grasping Compensation and Hemiparetic Upper Limb Rehabilitation." *Robotics and Autonomous Systems* 93. Elsevier B.V.: 1–12. doi:10.1016/j.robot.2017.03.015.

Iida, Fumiya, and Cecilia Laschi. 2011. "Soft Robotics: Challenges and Perspectives." *Procedia Computer Science* 7: 99–102. doi:10.1016/j.procs.2011.12.030.

Ilievski, Filip, Aaron D. Mazzeo, Robert F. Shepherd, Xin Chen, and George M. Whitesides. 2011. "Soft Robotics for Chemists." *Angewandte Chemie - International Edition* 50 (8): 1890–95. doi:10.1002/anie.201006464.

Kier, W.M., and K.K. Smith. 1983. "The Biomechanics of Movement in Tongues and Tentacles." *Journal of Biomechanics* 16 (4): 292–93. doi:10.1016/0021-9290(83)90176-8.

Kim, Sangbae, Cecilia Laschi, and Barry Trimmer. 2013. "Soft Robotics: A Bioinspired Evolution in Robotics." *Trends in Biotechnology*. Elsevier Ltd. doi:10.1016/j.tibtech.2013.03.002.

Kirby, Russell S., Martha S. Wingate, Kim Van Naarden Braun, Nancy S. Doernberg, Carrie L. Arneson, Ruth E. Benedict, Beverly Mulvihill, et al. 2011. "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." *Research in Developmental Disabilities* 32 (2). Pergamon: 462–69. doi:10.1016/J.RIDD.2010.12.042.

Knapik, JJ, KL Reynolds, and E Harman. 2004. "Soldier Load Carriage: Historical, Physiological, Biomechanical, and Medical Aspects." *Military Medicine* 169 (1): 45–56. doi:10.7205/MILMED.169.1.45.

Kurek, Daniel A., and H. Harry Asada. 2017. "The MantisBot: Design and Impedance Control of Supernumerary Robotic Limbs for near-Ground Work." *Proceedings - IEEE International Conference on Robotics and Automation*, 5942–47. doi:10.1109/ICRA.2017.7989700.

Llorens-Bonilla, Baldin, Federico Parietti, and H. Harry Asada. 2012. "Demonstration-Based Control of Supernumerary Robotic Limbs." *IEEE International Conference on Intelligent Robots and Systems*, no. Figure 1: 3936–42. doi:10.1109/IROS.2012.6386055.

Lubelski, Daniel, Matthew D. Alvin, Sergiy Nesterenko, Swetha J. Sundar, Nicolas R. Thompson, Edward C. Benzel, and Thomas E. Mroz. 2016. "Correlation of Quality of Life and Functional Outcome Measures for Cervical Spondylotic Myelopathy." *Journal of Neurosurgery: Spine* 24 (3): 483–89. doi:10.3171/2015.6.SPINE159.

Maheu, Veronique, Philippe S. Archambault, Julie Frappier, and François Routhier. 2011. "Evaluation of the JACO Robotic Arm: Clinico-Economic Study for Powered Wheelchair Users with Upper-Extremity Disabilities." *IEEE International Conference on Rehabilitation Robotics*, 4–6. doi:10.1109/ICORR.2011.5975397.

Majidi, Carmel. 2014. "Soft Robotics: A Perspective—Current Trends and Prospects for the Future." *Soft Robotics* 1 (1): 5–11. doi:10.1089/soro.2013.0001.

Nguyen, Pham Huy, Saivimal Sridar, Wenlong Zhang, and Panagiotis Polygerinos. 2017. "Design and Control of a 3-Chambered Fiber Reinforced Soft Actuator with off-the-Shelf Stretch Sensors." *International Journal of Intelligent Robotics and Applications* 1 (3). Springer Singapore: 342–51. doi:10.1007/s41315-017-0020-z.

Parietti, Federico, and H. Harry Asada. 2014. "Supernumerary Robotic Limbs for Aircraft Fuselage Assembly: Body Stabilization and Guidance by Bracing." *Proceedings - IEEE International Conference on Robotics and Automation*, 1176–83. doi:10.1109/ICRA.2014.6907002.

Parietti, Frederico, and H. Harry Asada. 2017. "Independent, Voluntary Control of Extra Robotic Limbs." *Proceedings - IEEE International Conference on Robotics and Automation*, 5954–61. doi:10.1109/ICRA.2017.7989702.

Parietti, Federico, Kameron C. Chan, Banks Hunter, and H. Harry Asada. 2015. "Design and Control of Supernumerary Robotic Limbs for Balance Augmentation." *Proceedings - IEEE International Conference on Robotics and Automation* 2015–June (June): 5010–17. doi:10.1109/ICRA.2015.7139896.

Plagenhoef, Stanley, F. Gaynor Evans, and Thomas Abdelnour. 1983. "Anatomical Data for Analyzing Human Motion." *Research Quarterly for Exercise and Sport* 54 (2): 169–78. doi:10.1080/02701367.1983.10605290.

Polygerinos, Panagiotis, Nikolaus Correll, Stephen A. Morin, Bobak Mosadegh, Cagdas D. Onal, Kirstin Petersen, Matteo Cianchetti, Michael T. Tolley, and Robert F. Shepherd. 2017. "Soft Robotics: Review of Fluid-Driven Intrinsically Soft Devices; Manufacturing, Sensing, Control, and Applications in Human-Robot Interaction." *Advanced Engineering Materials*, 1–22. doi:10.1002/adem.201700016.

Salvietti, Gionata, Irfan Hussain, David Cioncoloni, Sabrina Taddei, Simone Rossi, and Domenico Prattichizzo. 2017. "Compensating Hand Function in Chronic Stroke Patients Through a Robotic Extra-Finger" 25 (2): 142–50.

Sasaki, Tomoya, MHD Yamen Saraiji, Charith Lasantha Fernando, Kouta Minamizawa, and Masahiko Inami. 2017. "MetaLimbs." *ACM SIGGRAPH 2017 Emerging Technologies on - SIGGRAPH '17*, no. July: 1–2. doi:10.1145/3084822.3084837.

Shepherd, R. F., F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, and G. M. Whitesides. 2011. "Multigait Soft Robot." *Proceedings of the National Academy of Sciences* 108 (51): 20400–403. doi:10.1073/pnas.1116564108.

Tiziani, Lucas, Alexander Hart, Thomas Cahoon, Faye Wu, H. Harry Asada, and Frank L. Hammond. 2017. "Empirical Characterization of Modular Variable Stiffness Inflatable Structures for Supernumerary Grasp-Assist Devices." *International Journal of Robotics Research* 36 (13–14): 1391–1413. doi:10.1177/0278364917714062.

Tsakiris, Manos, Lewis Carpenter, Dafydd James, and Aikaterini Fotopoulou. 2010. "Hands Only Illusion: Multisensory Integration Elicits Sense of Ownership for Body Parts but Not for Non-Corporeal Objects." *Experimental Brain Research* 204 (3): 343–52. doi:10.1007/s00221-009-2039-3.

Vatsal, Vighnesh, and Guy Hoffman. 2017. "Wearing Your Arm on Your Sleeve : Studying Usage Contexts for a Wearable Robotic Forearm." doi:10.1109/ROMAN.2017.8172421.

Wu, Faye Y., and H. Harry Asada. 2016. "Implicit and Intuitive Grasp Posture Control for Wearable Robotic Fingers: A Data-Driven Method Using Partial Least Squares." *IEEE Transactions on Robotics* 32 (1): 176–86. doi:10.1109/TRO.2015.2506731.

APPENDIX A

CO-AUTHOR PERMISSION

The work highlighted in this document was included in a paper recently submitted for publication, in collaboration with Panagiotis Polygerinos, Pham Huy Nguyen, Curtis Sparks, and Gai G. Nuthi. Permission was given from all co-authors to include materials from the paper in this thesis.

APPENDIX B

IMU ARDUINO CONTROL CODE

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imumaths.h>

// IMU Variables
#define BNO055_SAMPLERATE_DELAY_MS (100)
Adafruit_BNO055 bno = Adafruit_BNO055();
int initQuality = 3;
int numCalRead = 40;
float i = 0;
float pi = 3.14159265359;
float totalX = 0;
float totalY = 0;
float totalZ = 0;
float avgX = 0;
float avgY = 0;
float avgZ = 0;
float offsetX = 0;
float offsetY = 0;
float offsetZ = 0;
int calibrated = 0;

// Button Variables
const int upButton = 2;
const int downButton = 4;
int upButtonState = 0;
int downButtonState = 0;
int mode = 0;

/* ------- Code ------- */
void setup(void) {
  Serial.begin(9600);
  pinMode(upButton, INPUT); // initialize push button as input
  pinMode(downButton, INPUT);
  /* Initialise the BNO055 sensor */
  if (!bno.begin()) {
    while (1);
  }

  int8_t temp = bno.getTemp();
  bno.setExtCrystalUse(true);
}

void loop(void) {
  const unsigned long milliWait = 700; //milliseconds
  static unsigned long lastSampleTime = 0 - milliWait;
  // initialize such that a reading is due the first time through loop()

  /* Get Euler readings and sensor calibration status of BNO055 */
  imu::Vector<3> euler = bno.getVector(Adafruit_BNO055::VECTOR_EULER);
  uint8_t system, gyro, accel, mag = 0;
  bno.getCalibration(&system, &gyro, &accel, &mag);

  /* Read state of the push button */
  upButtonState = digitalRead(upButton);
  downButtonState = digitalRead(downButton);

```

Figure B1: IMU Arduino code (C++) lines 1-58.

```
59  /* Display calibration status at beginning of output */
60  if (i > numCalRead) {
61    calibrated = 2;
62    Serial.print("YES");
63    Serial.print(" ");
64  } else if (i > 0 && i <= numCalRead) {
65    calibrated = 1;
66    Serial.print("CALIBRATING");
67    Serial.print(" ");
68  } else {
69    calibrated = 0;
70    Serial.print("NO");
71    Serial.print(" ");
72  }
73
74  /* Calibration code */
75  if (gyro && mag == initQuality) {
76    // can change the required system quiality above
77    if (i <= numCalRead) {
78      // can change the number of readings above void setup()
79      if (i == 0) {
80        for (int startTimer = 0; startTimer < 50; startTimer++) {
81          Serial.print("START_CAL ");
82          Serial.println(50 - startTimer);
83          delay(BNO055_SAMPLERATE_DELAY_MS);
84        }
85        totalX = 0;
86        totalY = 0;
87        totalZ = 0;
88        avgX = 0;
89        avgY = 0;
90        avgZ = 0;
91        i = i + 1;
92      } else {
93        totalX = totalX + euler.x();
94        totalY = totalY + euler.y();
95        totalZ = totalZ + euler.z();
96        avgX = totalX / i;
97        avgY = totalY / i;
98        avgZ = totalZ / i;
99        Serial.print(" ");
100       Serial.print(avgX, 3);
101       Serial.print(" ");
102       Serial.print(avgY, 3);
103       Serial.print(" ");
104       Serial.print(avgZ, 3);
105       Serial.print(" ");
106       Serial.println(numCalRead + 1 - i); //count down samples
107       i = i + 1;
108     }
109   } else if (i == numCalRead + 1) {
110     for (int endTimer = 0; endTimer < 30; endTimer++) {
111       Serial.print("END_CAL ");
112       Serial.println(30 - endTimer);
113       delay(BNO055_SAMPLERATE_DELAY_MS);
114     }
115     i = i + 1;
116   }
117 }
118
```

Figure B2: IMU Arduino code (C++) lines 59-118.

```
119  /* Display absolute sensor calibration status if not calibrated */
120  if (calibrated == 0) {
121    /* Display the Euler X, Y, and Z (floating point data) */
122    Serial.print(euler.x(), 3);
123    Serial.print(" ");
124    Serial.print(euler.y(), 3);
125    Serial.print(" ");
126    Serial.print(euler.z(), 3);
127    Serial.print(" ");
128    Serial.print(system, DEC);
129    Serial.print(" ");
130    Serial.print(gyro, DEC);
131    Serial.print(" ");
132    Serial.print(accel, DEC);
133    Serial.print(" ");
134    Serial.println(mag, DEC);
135  }
136  else if (calibrated == 2) {
137    offsetX = (euler.x() - avgX);
138    offsetY = (euler.y() - avgY);
139    offsetZ = (euler.z() - avgZ);
140
141    float posNinety = 90;
142    float negNinety = -90;
143
144    float plusY = posNinety - avgY;
145    float minY = negNinety - avgY;
146
147    if (offsetX > 180) {
148      offsetX = offsetX - 360;
149    } else if (offsetX < - 180) {
150      offsetX = offsetX + 360;
151    }
152
153    if (offsetZ > 180) {
154      offsetZ = offsetZ - 360;
155    } else if (offsetZ < - 180) {
156      offsetZ = offsetZ + 360;
157    }
158
159    float cutoffX = offsetX;
160    float cutoffY = offsetY;
161    float cutoffZ = offsetZ;
162
163    if (cutoffX > 90) {
164      cutoffX = 90;
165    } else if (cutoffX < -90) {
166      cutoffX = -90;
167    }
168    if (cutoffZ > 90) {
169      cutoffZ = 90;
170    } else if (cutoffZ < -90) {
171      cutoffZ = -90;
172    }
173
174    double mapX = mapf(cutoffX, -90, 90, -1, 1);
175    double mapY = mapf(cutoffY, minY, plusY, -1, 1);
176    double mapZ = mapf(cutoffZ, -90, 90, -1, 1);
177
```

Figure B3: IMU Arduino code (C++) lines 119-177.

```
178     /* Send XY position to ouput based on mapped values */
179     if (mapX < -0.40 && mapY > 0.40) {
180       Serial.print("-2X+2Y");
181     } else if (mapX > -0.40 && mapX < -0.20 && mapY > 0.40) {
182       Serial.print("-1X+2Y");
183     } else if (mapX > -0.20 && mapX < 0.20 && mapY > 0.40) {
184       Serial.print("0X+2Y");
185     } else if (mapX > 0.20 && mapX < 0.40 && mapY > 0.40) {
186       Serial.print("+1X+2Y");
187     } else if (mapX > 0.40 && mapY > 0.40) {
188       Serial.print("+2X+2Y");
189     } else if (mapX < -0.40 && mapY > 0.20 && mapY < 0.40) {
190       Serial.print("-2X+1Y");
191     } else if (mapX > -0.40 && mapX < -0.20 && mapY > 0.20 && mapY < 0.40) {
192       Serial.print("-1X+1Y");
193     } else if (mapX > -0.20 && mapX < 0.20 && mapY > 0.20 && mapY < 0.40) {
194       Serial.print("0X+1Y");
195     } else if (mapX > 0.20 && mapX < 0.40 && mapY > 0.20 && mapY < 0.40) {
196       Serial.print("+1X+1Y");
197     } else if (mapX > 0.40 && mapY > 0.20 && mapY < 0.40) {
198       Serial.print("+2X+1Y");
199     } else if (mapX < -0.40 && mapY > -0.20 && mapY < 0.20) {
200       Serial.print("-2X0Y");
201     } else if (mapX > -0.40 && mapX < -0.20 && mapY > -0.20 && mapY < 0.20) {
202       Serial.print("-1X0Y");
203     } else if (mapX > -0.20 && mapX < 0.20 && mapY > -0.20 && mapY < 0.20) {
204       Serial.print("0X0Y");
205     } else if (mapX > 0.20 && mapX < 0.40 && mapY > -0.20 && mapY < 0.20) {
206       Serial.print("+1X0Y");
207     } else if (mapX > 0.40 && mapY > -0.20 && mapY < 0.20) {
208       Serial.print("+2X0Y");
209     } else if (mapX < -0.40 && mapY > -0.40 && mapY < -0.20) {
210       Serial.print("-2X-1Y");
211     } else if (mapX > -0.40 && mapX < -0.20 && mapY > -0.40 && mapY < -0.20) {
212       Serial.print("-1X-1Y");
213     } else if (mapX > -0.20 && mapX < 0.20 && mapY > -0.40 && mapY < -0.20) {
214       Serial.print("0X-1Y");
215     } else if (mapX > 0.20 && mapX < 0.40 && mapY > -0.40 && mapY < -0.20) {
216       Serial.print("+1X-1Y");
217     } else if (mapX > 0.40 && mapY > -0.40 && mapY < -0.20) {
218       Serial.print("+2X-1Y");
219     } else if (mapX < -0.40 && mapY < -0.40) {
220       Serial.print("-2X-2Y");
221     } else if (mapX > -0.40 && mapX < -0.20 && mapY < -0.40) {
222       Serial.print("-1X-2Y");
223     } else if (mapX > -0.20 && mapX < 0.20 && mapY < -0.40) {
224       Serial.print("0X-2Y");
225     } else if (mapX > 0.20 && mapX < 0.40 && mapY < -0.40) {
226       Serial.print("+1X-2Y");
227     } else if (mapX > 0.40 && mapY < -0.40) {
228       Serial.print("+2X-2Y");
229     }
230
231     Serial.print(" ");
232     Serial.print(mapX, 3);
233     Serial.print(" ");
234     Serial.print(mapY, 3);
235     Serial.print(" ");
236
```

Figure B4: IMU Arduino code (C++) lines 178-236.

```
237     unsigned long now = millis();
238     if (now - lastSampleTime >= milliWait) {
239       lastSampleTime += milliWait;
240       if (upButtonState == HIGH) {
241         mode = mode + 1;
242       }
243       if (downButtonState == HIGH) {
244         mode = mode - 1;
245       }
246       if (mode == 8) {
247         //0 = whole arm, 1 = hold, 2 = seg 1, 3 = hold,
248         //4 = seg 2, 5 = hold, 6 = seg 3, 7 = hold
249         mode = 0;
250       } else if (mode == -1) {
251         mode = 7;
252       }
253       Serial.println(mode);
254     }
255     else {
256       Serial.println(mode);
257     }
258   }
259
260   delay(BNO055_SAMPLERATE_DELAY_MS - 3.9); //with button loop
261 }
262
263 double mapf(double x, double in_min, double in_max, double out_min, double out_max) {
264   return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
265 }
```

Figure B5: IMU Arduino code (C++) lines 237-265.

APPENDIX C

EMG ARDUINO CONTROL CODE

```
/* Smoothing - Reads repeatedly from an analog input, calculating a running average
 *  and printing it to the computer. Keeps ten readings in an array and continually
 *  averages them.Define the number of samples to keep track of. The higher the
 *  number, the more the readings will be smoothed, but the slower the output will
 *  respond to the input. Using a constant rather than a normal variable lets us use
 *  this value to determine the size of the readings array. */
const int numReadings = 10;

int readings1[numReadings];    // the readings from the analog input 1
int readings2[numReadings];    // the readings from the analog input 2
int readIndex = 0;             // the index of the current reading
int total1 = 0;                // the running total for EMG 1 - inflate/deflate
int total2 = 0;                // the running total for EMG 2 - mode
int average1 = 0;              // the average for EMG 1
int average2 = 0;              // the average for EMG 2


int emgPin1 = A0;
int emgPin2 = A1;
const int thresh1_high = 610; // max inflation
const int thresh1_low = 370; // half inflation
const int thresh2 = 610;
int pressurize = 0;
int mode = 0; //start in upward modeion

void setup() {
  Serial.begin(9600);
  /* Initialize array of EMG readings to 0 */
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings1[thisReading] = 0;
    readings2[thisReading] = 0;
  }
}

void loop() {
  const unsigned long milliWait = 1200; //milliseconds
  static unsigned long lastSampleTime = 0 - milliWait;
  // initialize such that a reading is due the first time through loop()

  total1 = total1 - readings1[readIndex]; // subtract the last reading:
  total2 = total2 - readings2[readIndex];
  readings1[readIndex] = analogRead(emgPin1); // read from the sensor:
  readings2[readIndex] = analogRead(emgPin2);
  total1 = total1 + readings1[readIndex]; // add the reading to the total:
  total2 = total2 + readings2[readIndex];
  readIndex = readIndex + 1; // advance to the next position in the array:

  if (readIndex >= numReadings) { // if we're at the end of the array...
    readIndex = 0;                // ...wrap around to the beginning:
  }

  average1 = total1 / numReadings; // calculate the average...
  average2 = total2 / numReadings; // send it to the computer as ASCII digits

  if (abs(average1) > thresh1_high) {
    pressurize = 2;
  } else if (abs(average1) > thresh1_low && abs(average1) < thresh1_high) {
    pressurize = 1;
  } else {
    pressurize = 0;
  }

```

Figure C1: sEMG Arduino code (C++) lines 1-62.

```
63  unsigned long now = millis();
64  if (now - lastSampleTime >= milliWait) {
65    lastSampleTime += milliWait;
66    if (abs(average2) > thresh2) {
67      mode = mode + 1;
68    }
69    if (mode == 8) {
70      //0 = whole arm, 1 = hold, 2 = seg 1, 3 = hold,
71      //4 = seg 2, 5 = hold, 6 = seg 3, 7 = hold
72      mode = 0;
73    }
74  }
75
76  if (pressurize == 0) {
77    Serial.print("0X0Y");
78  } else if (mode == 0 && pressurize == 1) {
79    Serial.print("0X+1Y");
80  } else if (mode == 0 && pressurize == 2) {
81    Serial.print("0X+2Y");
82  } else if (mode == 1 && pressurize == 1) {
83    Serial.print("+1X+1Y");
84  } else if (mode == 1 && pressurize == 2) {
85    Serial.print("+2X+2Y");
86  } else if (mode == 2 && pressurize == 1) {
87    Serial.print("+1X0Y");
88  } else if (mode == 2 && pressurize == 2) {
89    Serial.print("+2X0Y");
90  } else if (mode == 3 && pressurize == 1) {
91    Serial.print("+1X-1Y");
92  } else if (mode == 3 && pressurize == 2) {
93    Serial.print("+2X-2Y");
94  } else if (mode == 4 && pressurize == 1) {
95    Serial.print("0X-1Y");
96  } else if (mode == 4 && pressurize == 2) {
97    Serial.print("0X-2Y");
98  } else if (mode == 5 && pressurize == 1) {
99    Serial.print("-1X-1Y");
100 } else if (mode == 5 && pressurize == 2) {
101   Serial.print("-2X-2Y");
102 } else if (mode == 6 && pressurize == 1) {
103   Serial.print("-1X0Y");
104 } else if (mode == 6 && pressurize == 2) {
105   Serial.print("-2X0Y");
106 } else if (mode == 7 && pressurize == 1) {
107   Serial.print("-1X+1Y");
108 } else if (mode == 7 && pressurize == 2) {
109   Serial.print("-2X+2Y");
110 }
111
112 Serial.print(" ");
113 Serial.print(abs(average1));
114 Serial.print(" ");
115 Serial.print(abs(average2));
116 Serial.print(" ");
117 Serial.print(mode);
118 Serial.print(" ");
119 Serial.println(pressurize);
120
121
122 delay(99); // delay for stability
123 }
```

Figure C1: sEMG Arduino code (C++) lines 63-123.

APPENDIX D

JOYSTICK LABVIEW CONTROL: FRONT PANEL AND BLOCK DIAGRAM

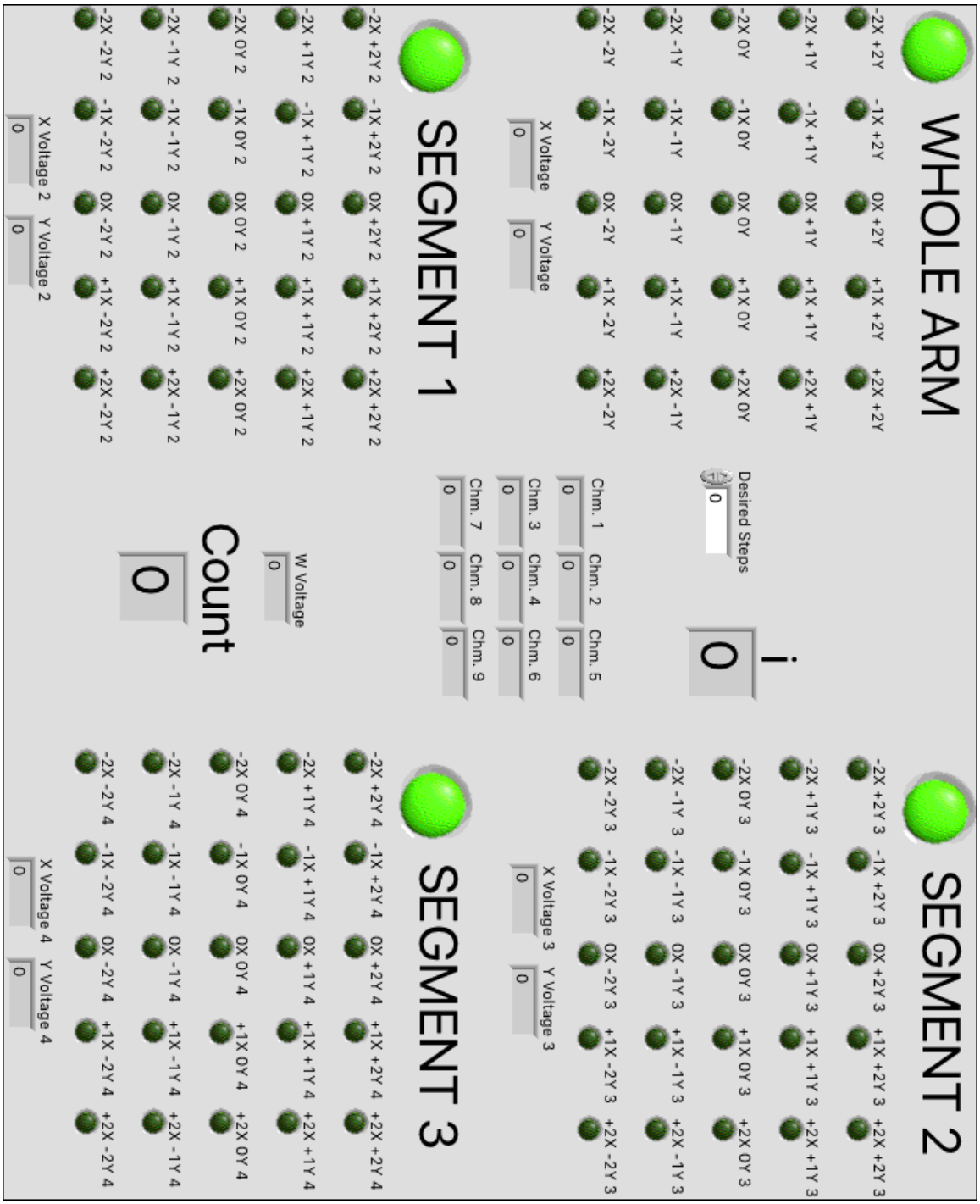

Figure D1: Joystick front panel. Includes stepping algorithm ("Desired Steps" as chosen by user) to reduce dynamic instabilities

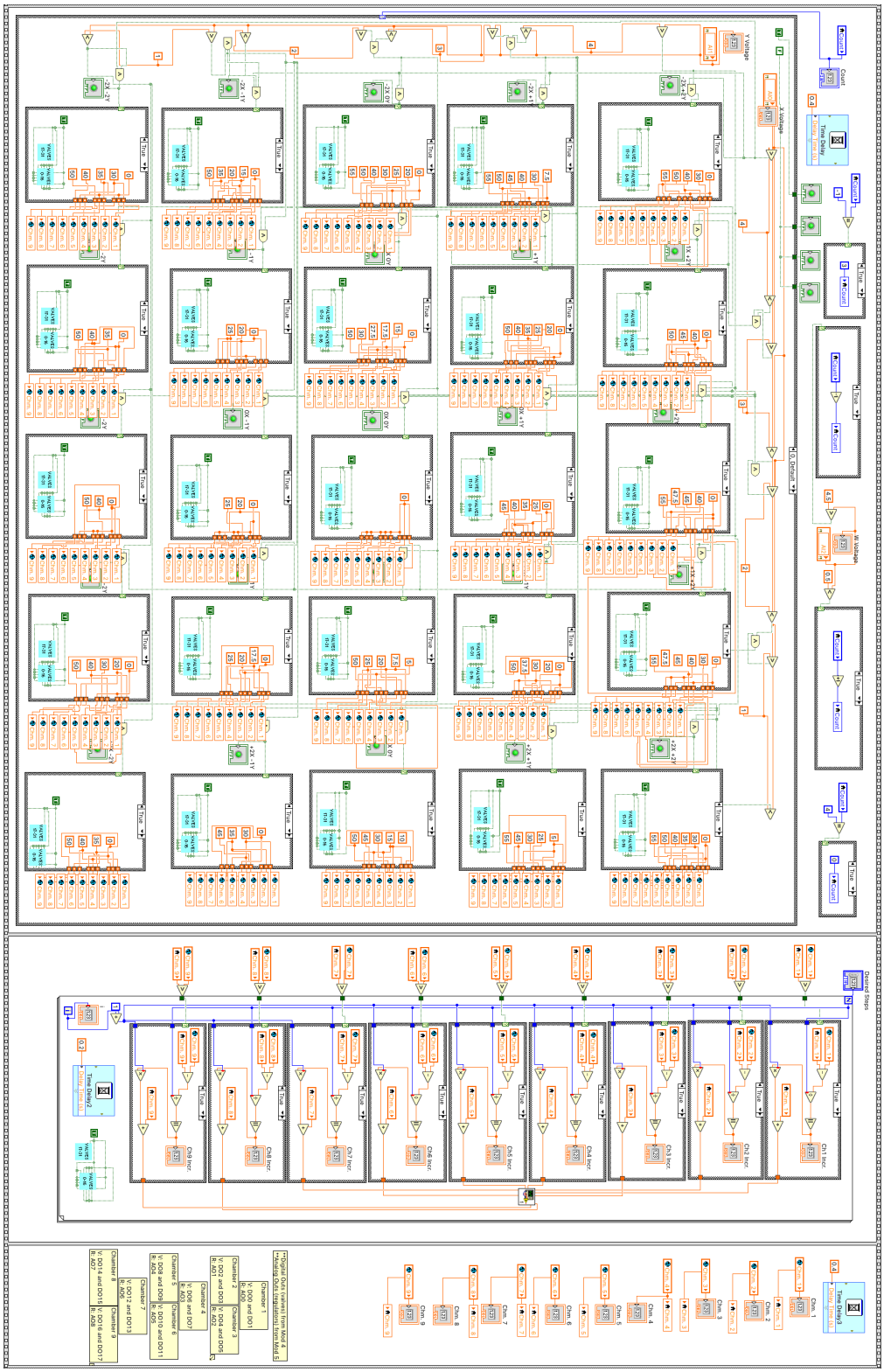

Figure D2: Joystick block diagram – full SPL control (gross motion). The dynamic instability reduction sequence is shown.

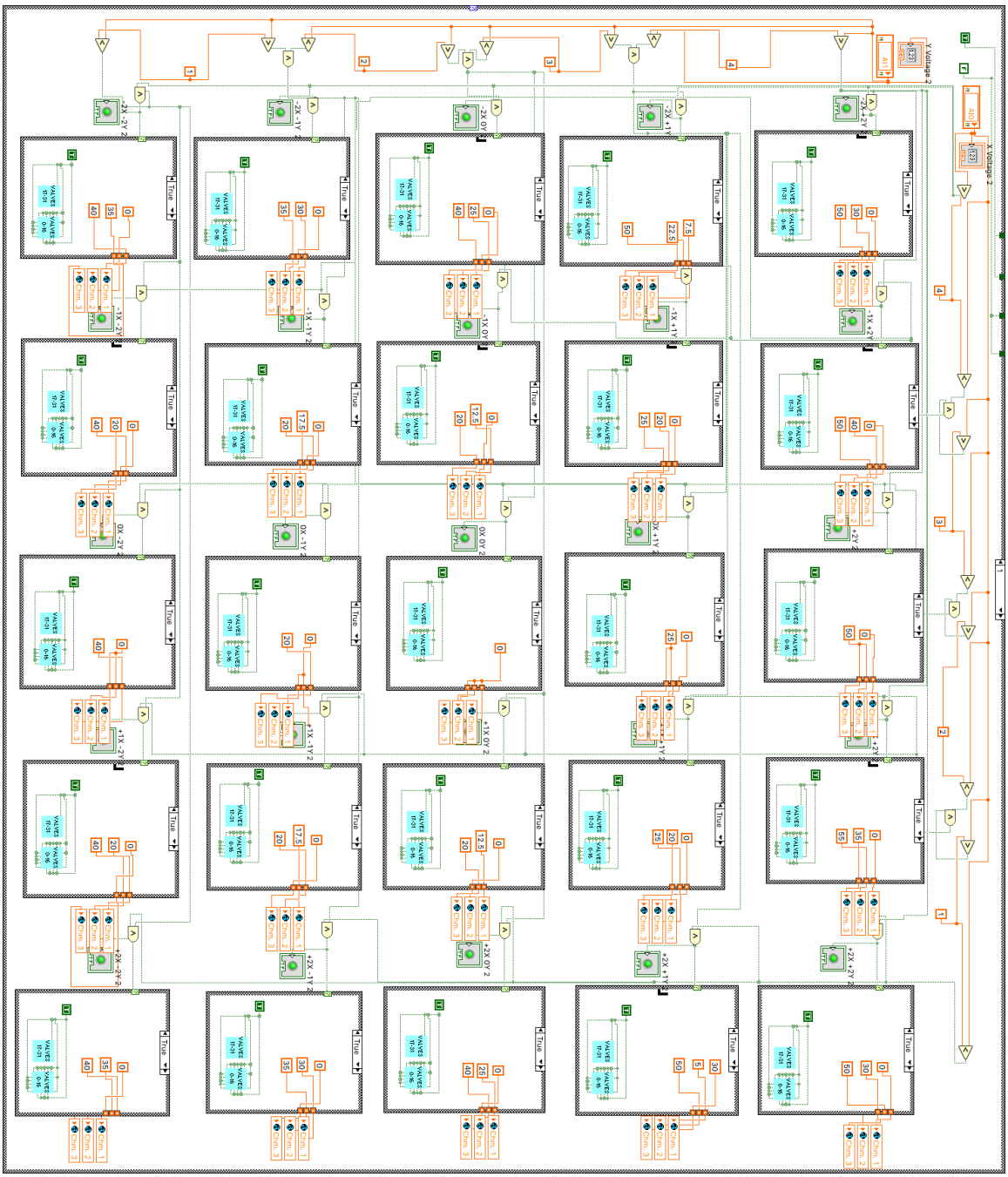

Figure D3: Joystick block diagram – segment 1 (proximal) control (fine motion).

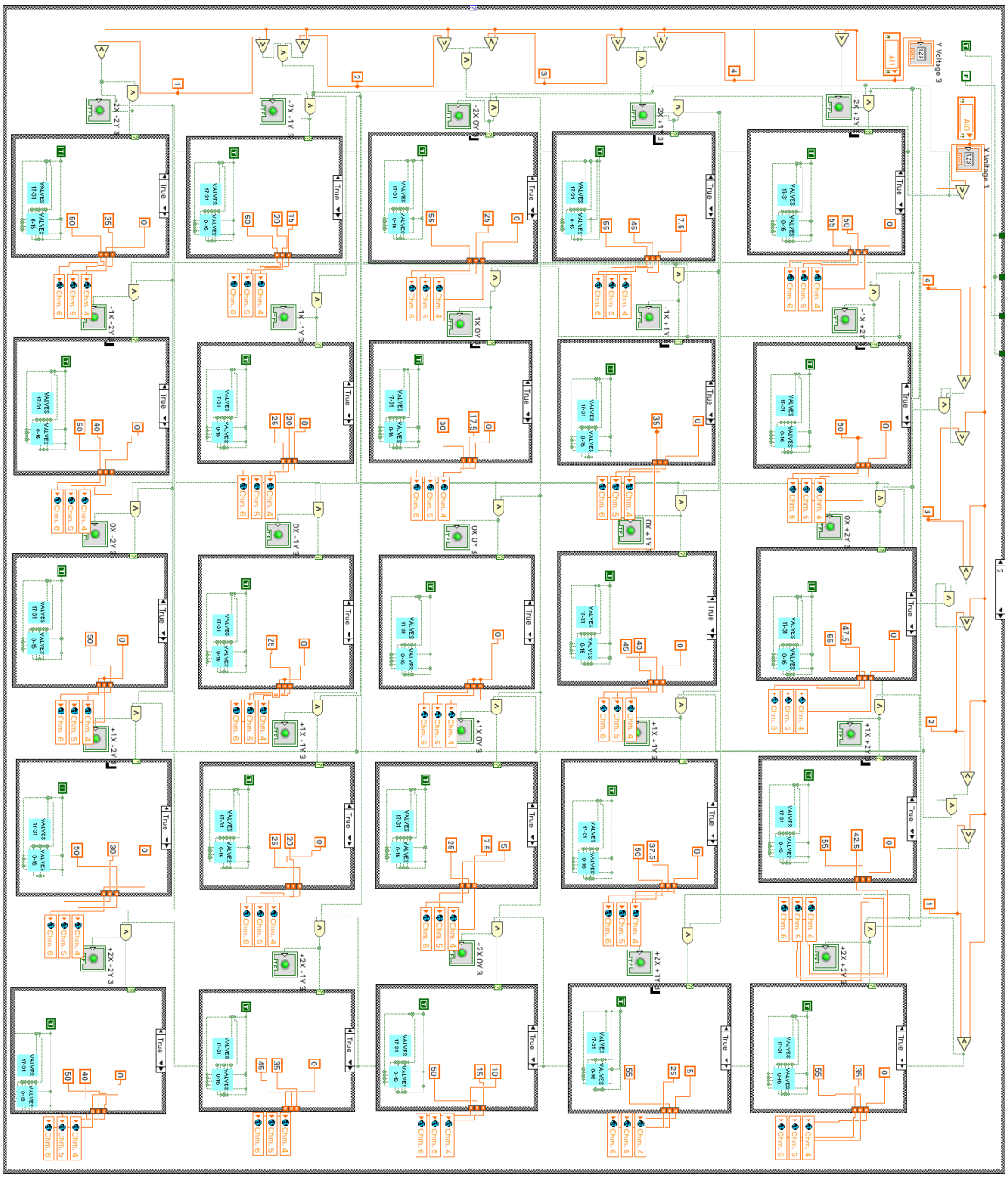

Figure D4: Joystick block diagram – segment 2 control (fine motion).

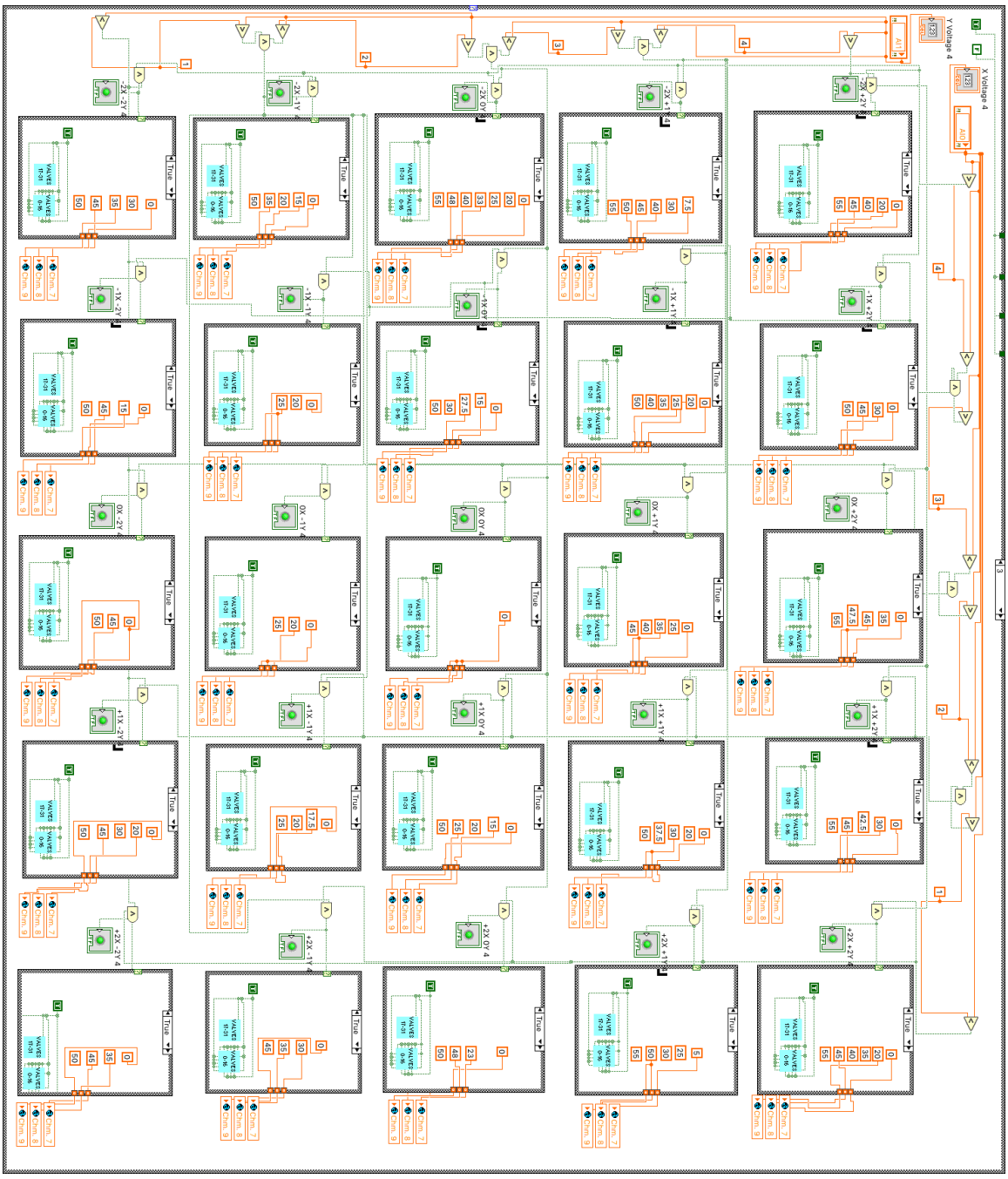

Figure D4: Joystick block diagram – segment 3 control (fine motion).

APPENDIX E

IMU LABVIEW CONTROL: FRONT PANEL AND BLOCK DIAGRAM

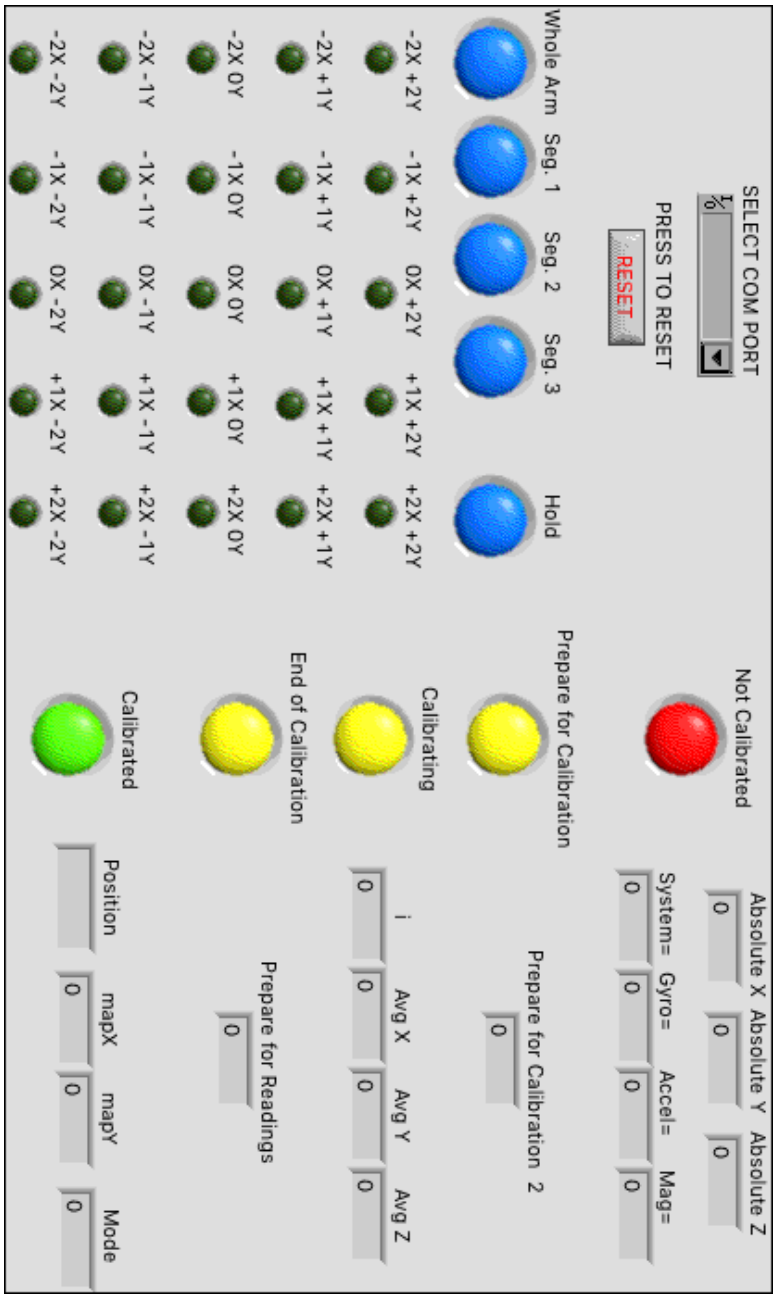

Figure E1: IMU front panel.

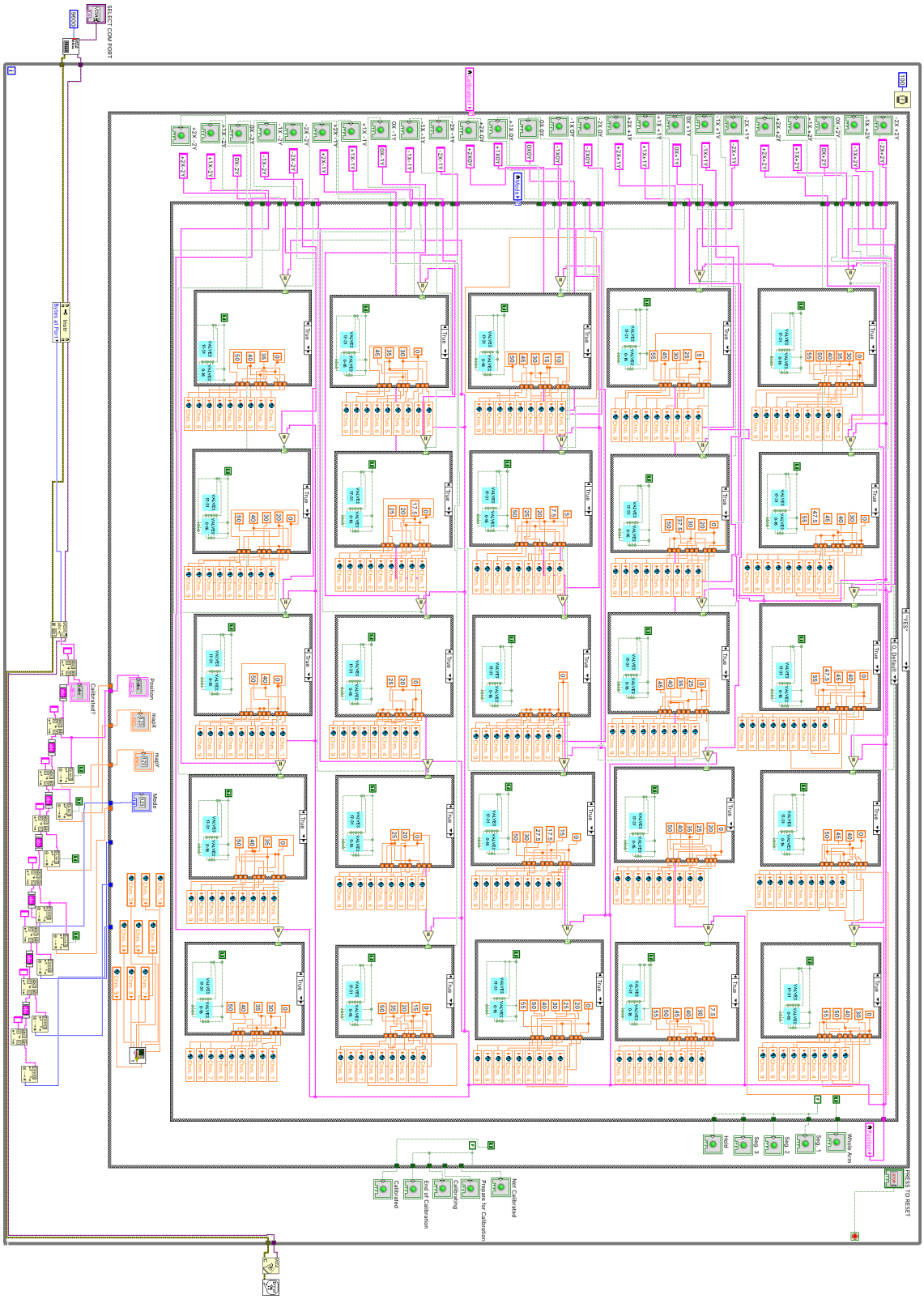

Figure E2: IMU block diagram – “Calibrated” full SPL control (gross motion).

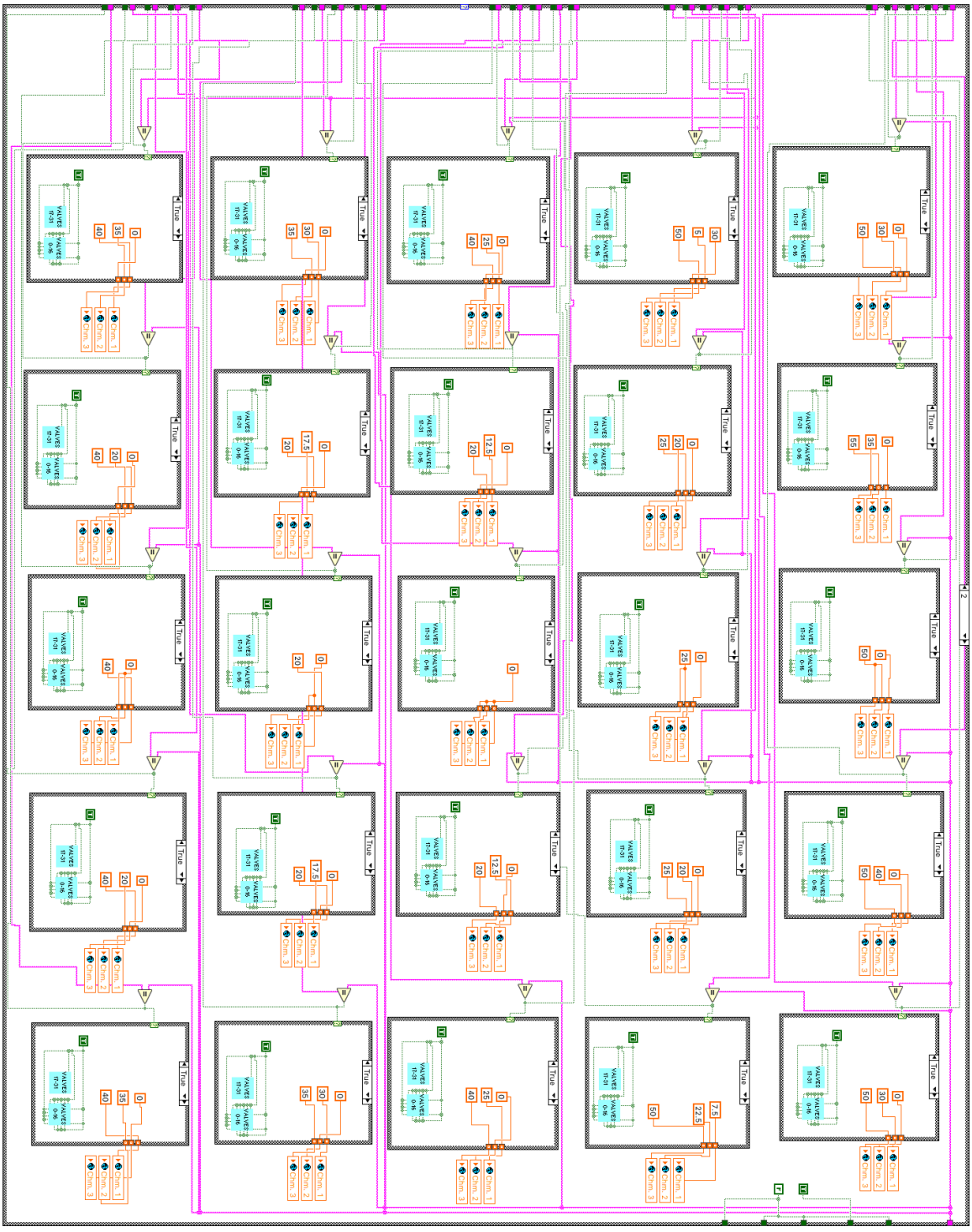

Figure E3: IMU block diagram – “Calibrated” segment 1 (proximal) control (fine motion).

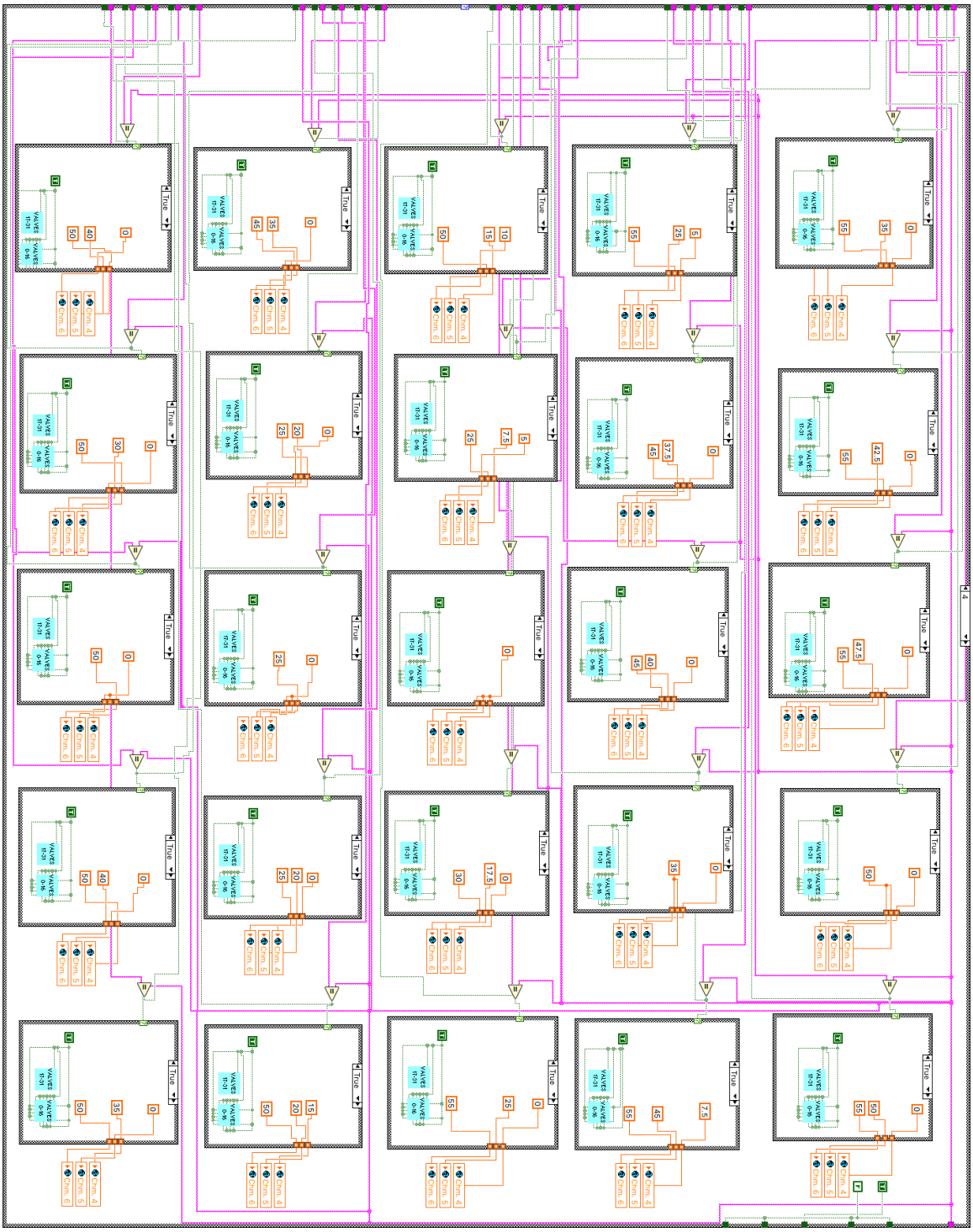

Figure E4: IMU block diagram – “Calibrated” segment 2 control (fine motion).

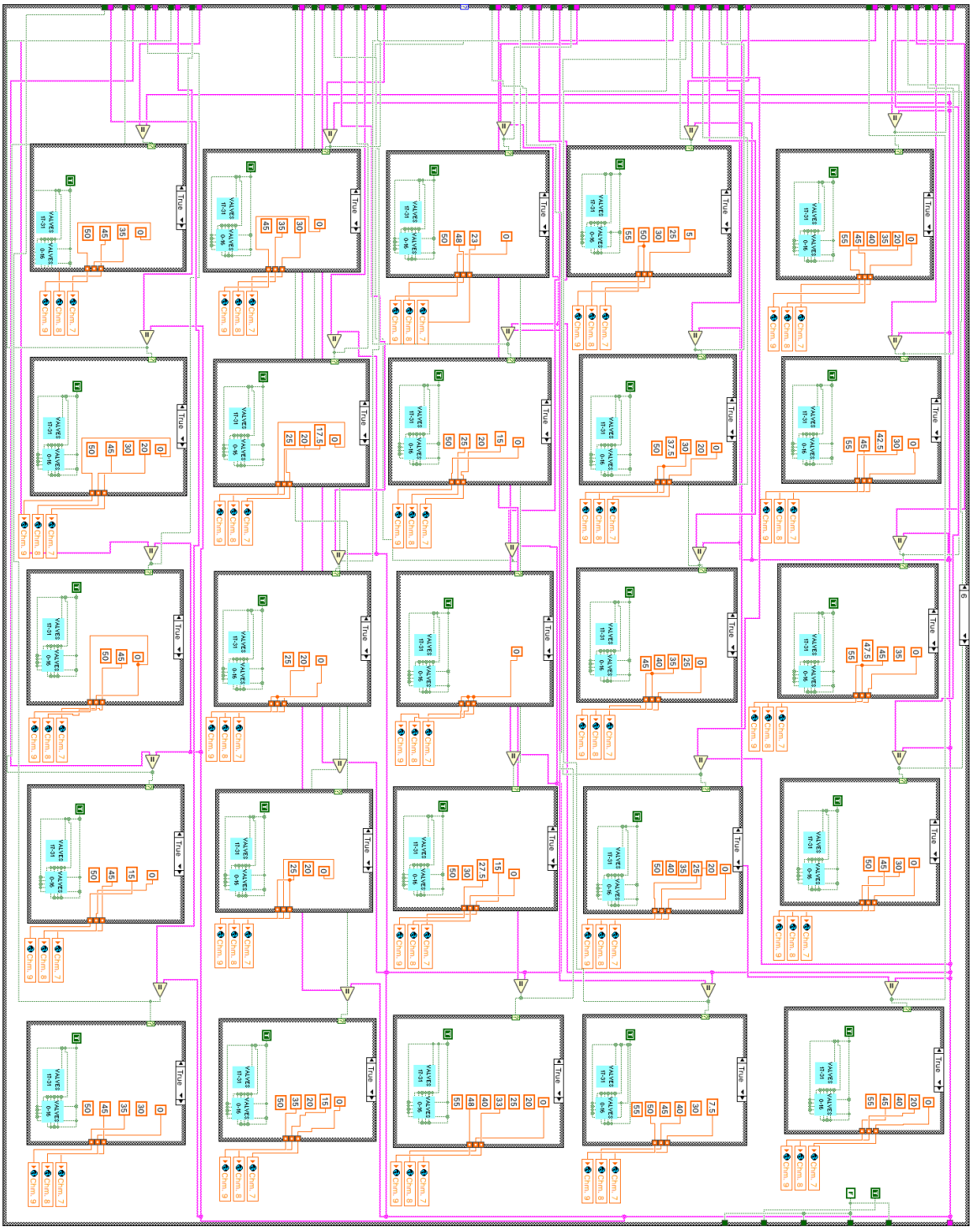

Figure E5: IMU block diagram – “Calibrated” segment 3 control (fine motion).

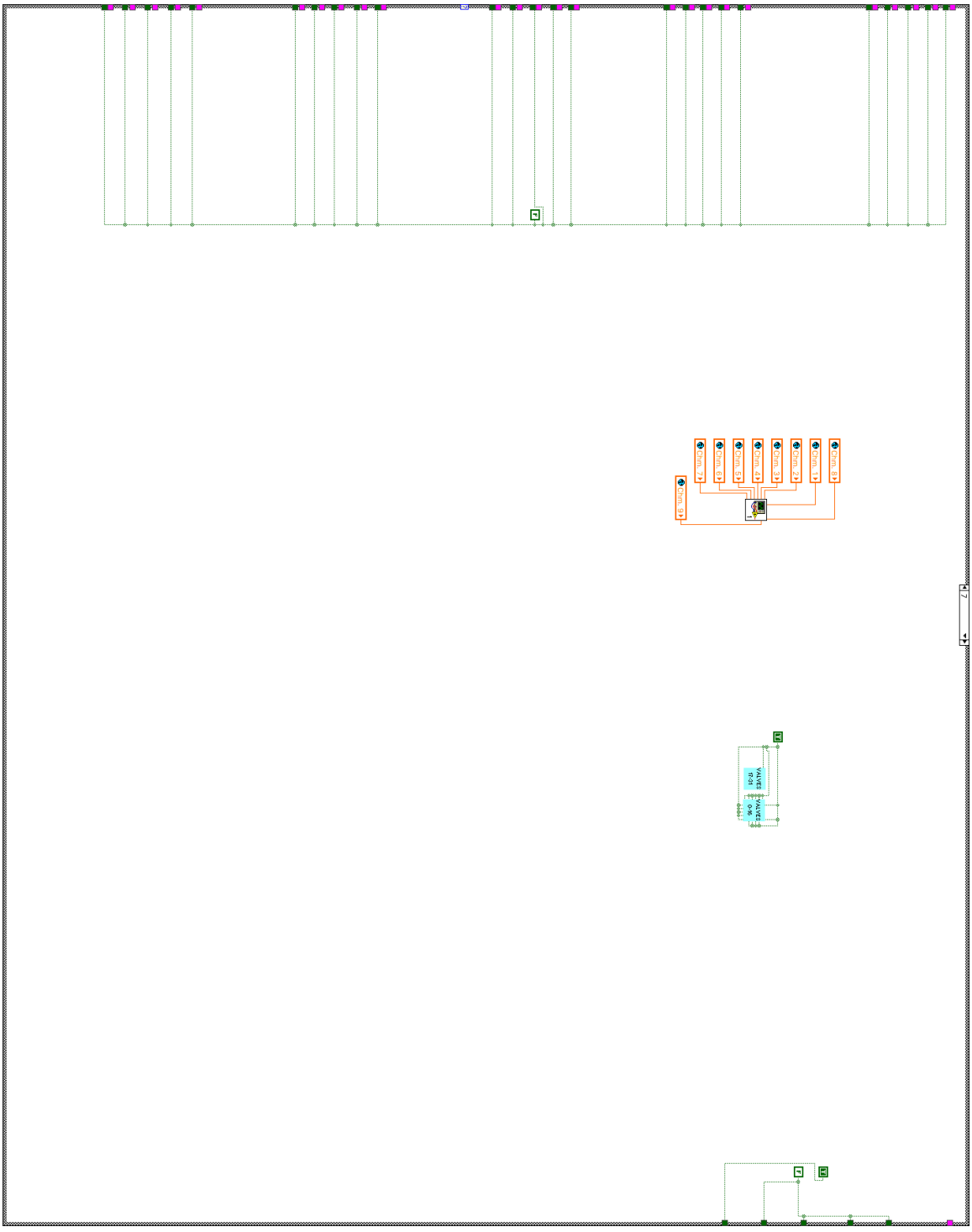

Figure E6: IMU block diagram – “Calibrated” hold of pressure for all actuators. Only “mode” 7 is show, however this same code is employed for “modes” 1, 3, and 5.

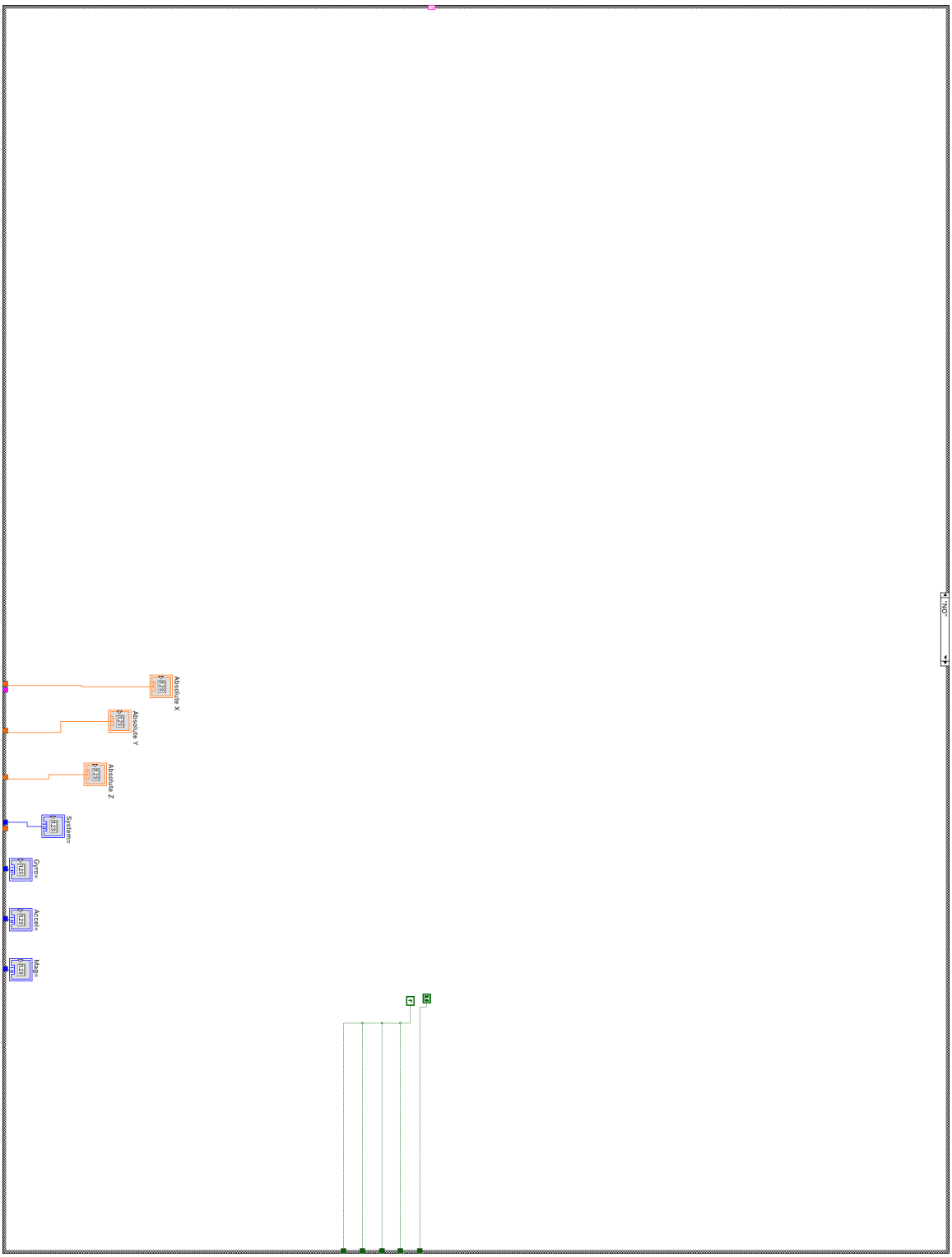

Figure E7: IMU block diagram – “Not calibrated” input.

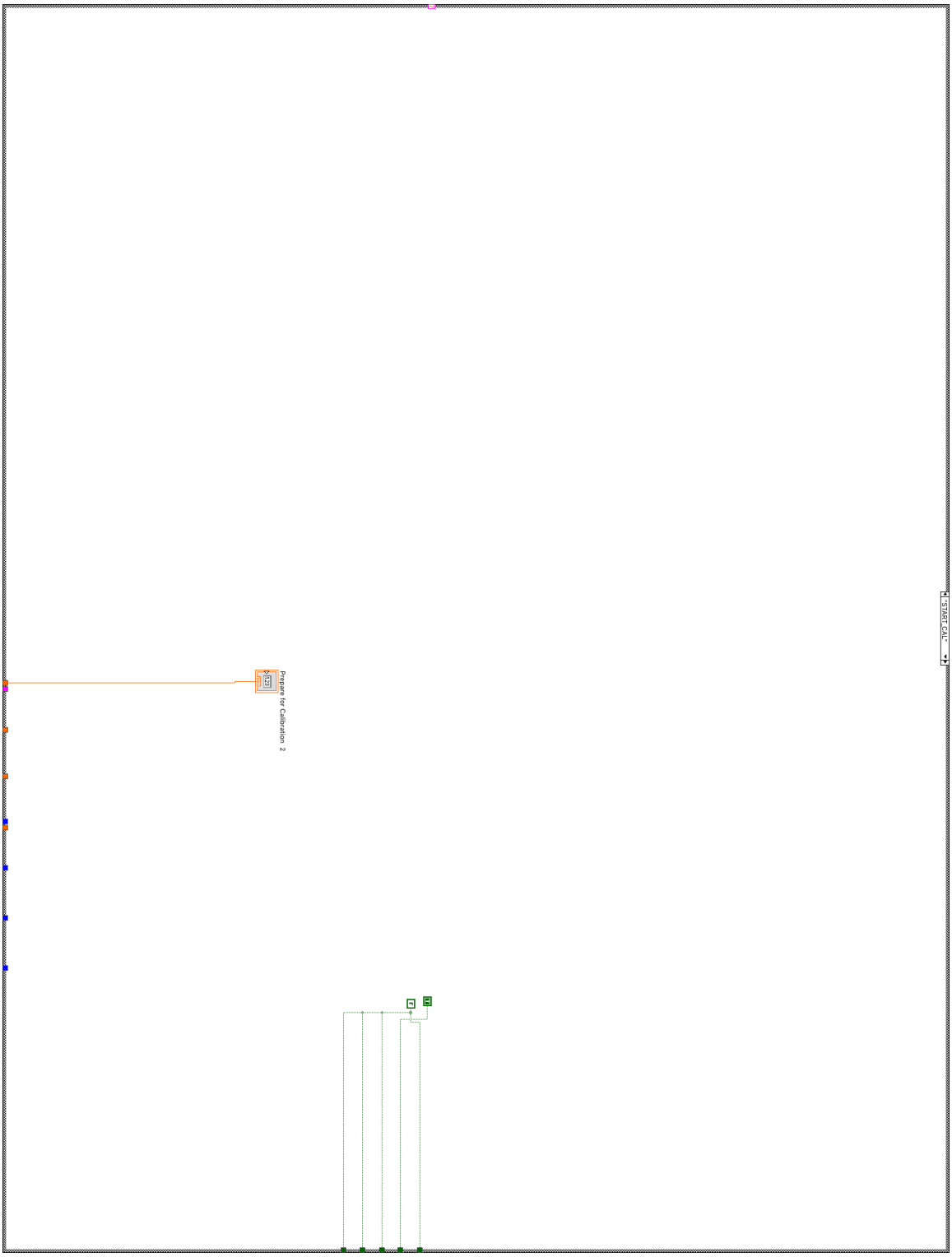

Figure E8: IMU block diagram – “Prepare for calibration” input.

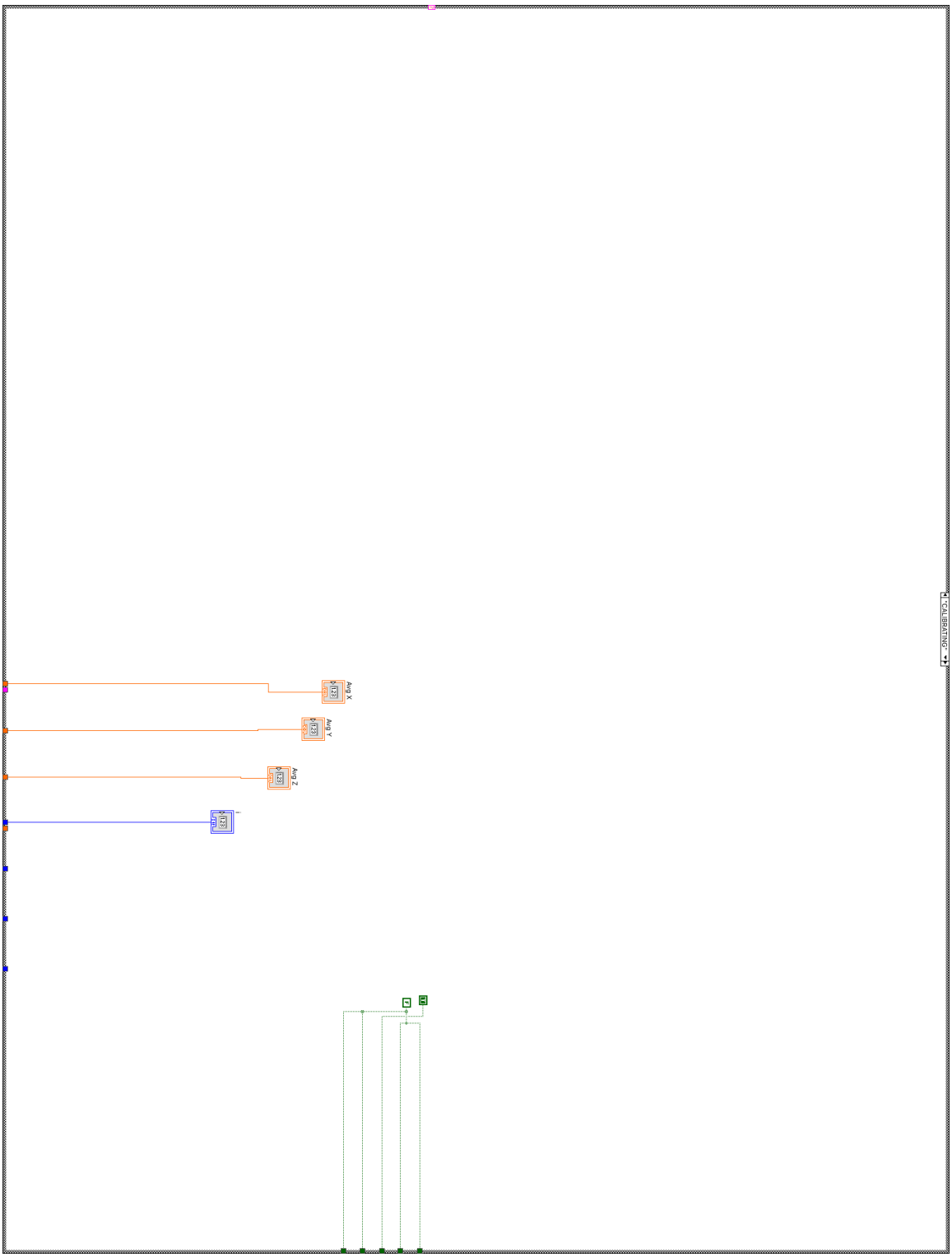

Figure E9: IMU block diagram – “Calibrating” the device.

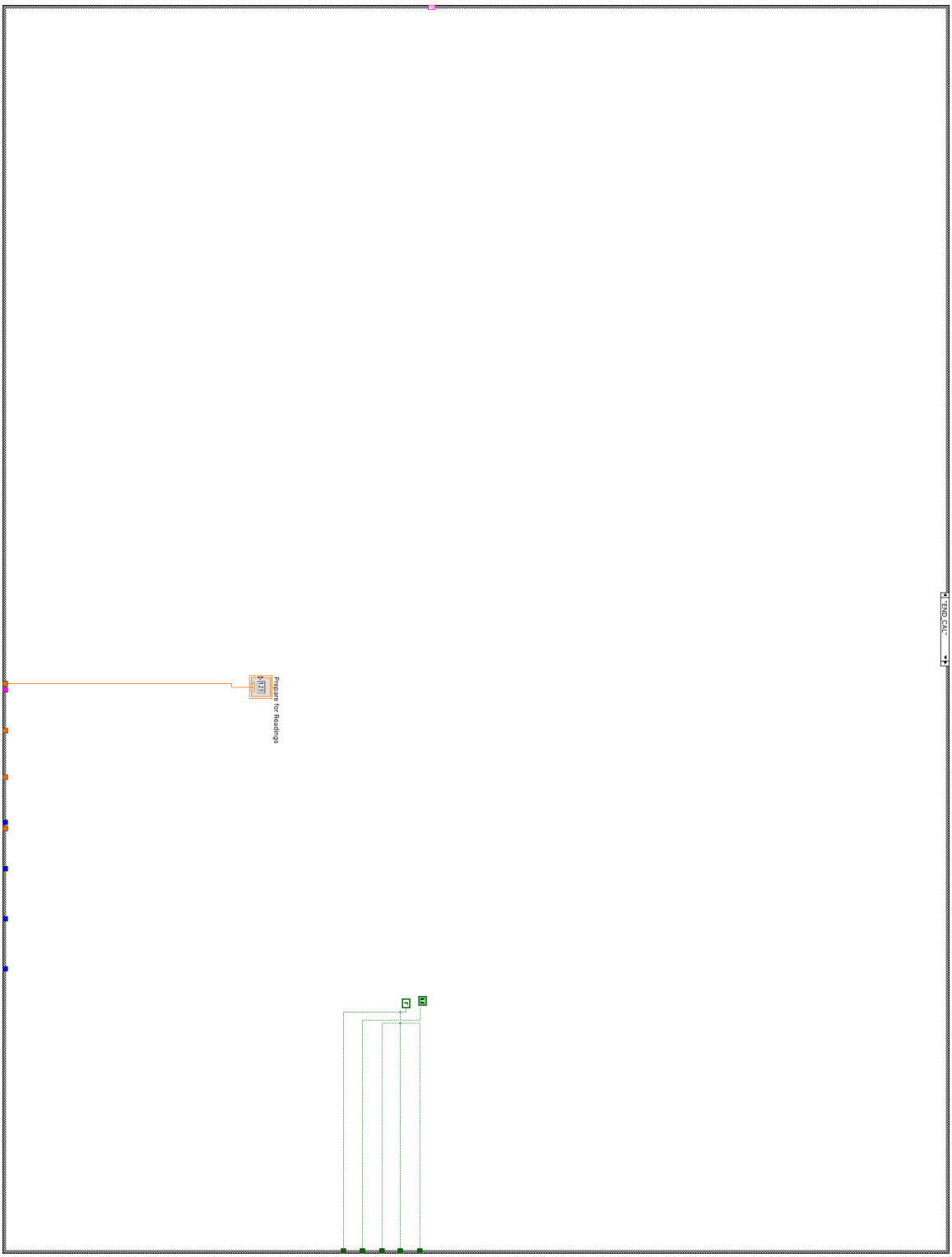

Figure E10: IMU block diagram – “End of calibration” input.

APPENDIX F

EMG LABVIEW CONTROL: FRONT PANEL AND BLOCK DIAGRAM

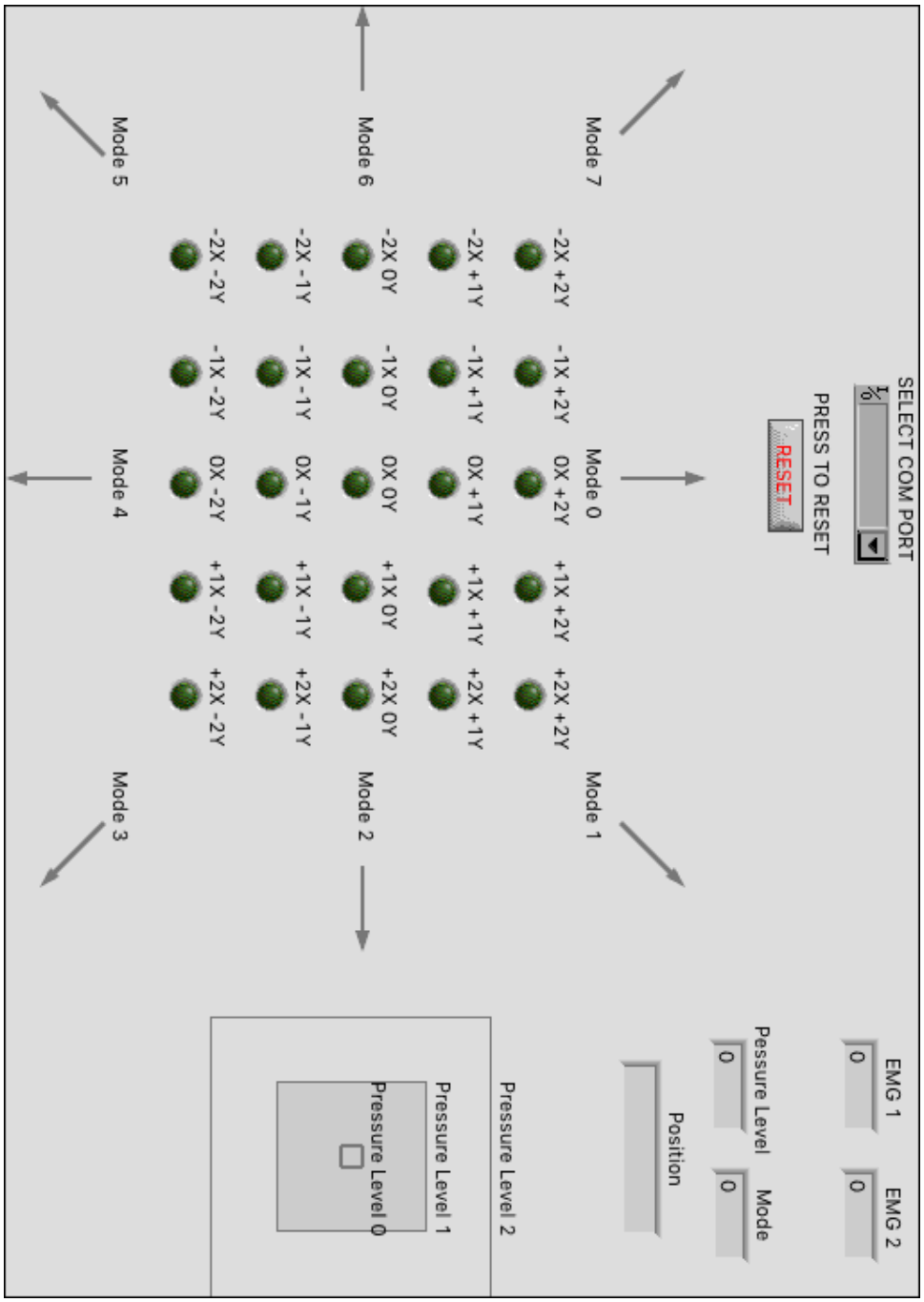

Figure F1: sEMG front panel.

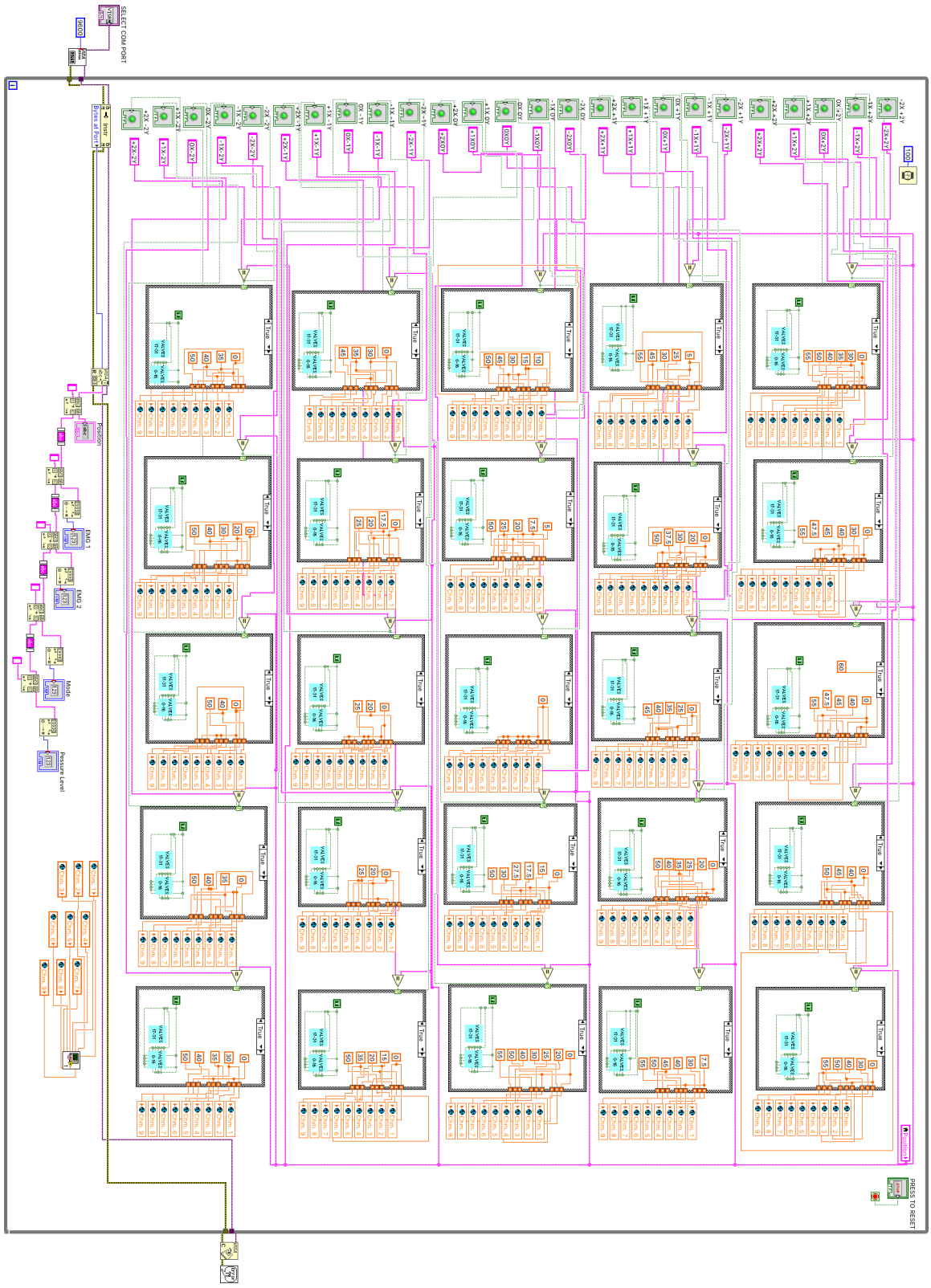

Figure F2: sEMG block diagram