2015

A Cost-Effective Haptic Device for Assistive and Rehabilitation Purposes

Archana Pradeep
Grand Valley State University

A Cost-Effective Haptic Device for Assistive and Rehabilitation Purposes

Archana Pradeep

A Thesis Submitted to the Graduate Faculty of

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Engineering

School of Engineering

December 2015

Acknowledgements

I would like to express my gratitude to my advisor, Dr. Nael Barakat for the useful comments, remarks and engagement through the learning process of this master thesis. Furthermore I would like to thank Mr. Deepak Dileepkumar for stimulating discussions and helping me understand the core concepts of hardware build. My sincere thanks to Mr. Phil Reuben for providing me an internship opportunity and leading me to work on related projects. I would also like to thank my family, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love.

Abstract

With the growing population of elderly, the need for assistance has also increased considerably especially for the tasks such as cleaning, reaching and grasping objects among others. There are numerous assistive devices in the market for this group of people. However, they are either too expensive or require overwhelming user effort for manipulation. Therefore, the presented research is primarily concerned with developing a low-cost, easy to use assistive device for elderly to reach and grasp objects through intuitive interface for the control of a slave anthropomorphic robotic arm (tele operator). The system also implements haptic feedback technology that enables the user to maneuver the grasping task in a realistic manner.

A bilateral master-slave robotic system combined with the haptic feedback technology has been designed, built and tested to determine the suitability of this device for the chosen application. The final prototype consists of primarily off the shelf components programmed in such a way as to provide accurate teleoperation and haptic feedback to the user. While the nature of the project as a prototype precluded any patient trials, testing of the final system has shown that a fairly low cost device can be capable of providing the user an ability to remotely control a robotic arm for reaching and grasping objects with accurate force feedback.

Keywords: Teleoperation, Bilateral master-slave system, assistive device

Table of contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background

The global population is rapidly ageing. By the year 2050, more than 1 in every 5 persons throughout the world is projected to be aged 60 or over, while nearly 1 in every 6 is projected to be at least 65 years old [1]. This growth in the elderly population will bring a corresponding surge in the number of elderly people with functional limitations. Functional limitations are physical problems that limit a person's ability to perform routine daily activities, such as eating, bathing, dressing, paying bills, and preparing meals. On average, about one-third of the people aged 65 or older report functional limitations of one kind or another. The proportion of older adults experiencing limitations in activities of daily living increases with age ranging from 26% at 65 to 74 years to 63% at 85 years or older. One study estimates that more than two-thirds of 65-year-olds will need assistance to deal with a loss in functioning at some point during their remaining years of life [2]. If those rates of prevalence continue, the number of elderly people with functional or cognitive limitations, and thus the need for assistance, will increase sharply in the coming decades.

Out of the everyday tasks that are challenging for the elderly to perform, picking up miscellaneous objects, especially from the floor or a shelf, and carrying objects, are listed as high priority tasks as shown by Table 1.1. The same has been confirmed through a recent interview conducted at the local assisted living place (Appendix A).

Functional limitations are frequently compensated by informal care. A proportion is also addressed through assistive technology. According to the disability statistics report [3], over 6.8 million Americans use assistive devices to compensate for the impairment. Out of this, nearly 64% of the assistive device use is by persons aged 65 and over [3]. An assistive device is anything that helps a homecare patient with activities of daily living. The technology related assistance for individuals with disabilities act of 1988 defined an assistive technology device as "any item, piece of equipment, or product system, whether acquired commercially off the shelf, modified or customized, that is used to increase, maintain or improve functional capabilities of individuals with disabilities". These devices help a patient perform activities that might otherwise be difficult or impossible.

Table 1.1: User task priorities as surveyed by Stanger et al. [4]

Priority	Task
High	• Picking up misc. Objects, esp. from floor or shelf • Carrying objects
Moderate to High	• Eating/Drinking • Preparing Food and Drinks • Personal hygiene • Leisure and Recreation

Based on the survey results shown in the table 1.1 and interview with the elderly, the most challenging everyday task is to reach and grasp objects from shelves. Therefore, the scope of the thesis is limited to research and development of optimal solution for reaching and grasping object from shelves.

1.2. Problem statement and purpose of the study

There are many robotic assistive devices in the market that enable the elderly to reach and grasp things. However, they are either expensive or require overwhelming user effort (both mental and physical) for manipulation. Therefore, there is a need for a cost effective assistive device which can be manipulated with minimal effort and guidance.

The goal of this study was therefore to develop an effective assistive device for executing some of the high priority tasks challenging the elderly population with minimal effort, while taking into consideration cost, simplicity, safety and acceptance. The parameters for developing a new assistive device were obtained through in depth interviews with elderly, literature review and researching existing products in the market.

Furthermore, greater levels of activity are linked to successful ageing [5]. Literature suggests that machine mediated therapy offered through haptic interfaces have the potential to improve the outcome of stroke patients engaged in rehabilitation for upper limb motor impairment [6]. Therefore, the purpose of this study also included validating the incorporation of haptics feedback technology into the assistive device that provides a sense of force feedback to the users. This would not only protect the objects held from being damaged but would also provide the users with therapy benefits especially for stroke patients.

1.3. Hypothesis

Based on the literature review, it is hypothesized that a bilateral master slave robotic system would be a suitable design for this application wherein master unit would consist of a partial wearable exoskeleton which will manipulate a remote mobile robotic arm (slave unit) and in turn obtain a force feedback sensed by the master unit indicating reactions experienced by the slave unit.

It is also hypothesized that the product based on this design would function considerably well by using off the shelf inexpensive components. Various qualitative and quantitative parameters are hypothesized and are detailed in section 3.

1.4. Overview of related concepts

1.4.1. Teleoperation for robot manipulation

The idea of teleoperation [9] has been around since the 1970s, a time when it was totally unfeasible to program adaptive robots, instead it was easier to allow human beings to control the robots from afar. The main advantage of this is that human beings are adaptive and so are better able to deal with unstructured environments.

Teleoperation has an important role in manipulating remote objects interactively using robotic manipulators, especially in hostile environments [10].

Robotic arms with prehensile functions are now extensively used in telemedicine, such as endoscopic teleoperation [11]. Da Vinci Surgical System from Intuitive Surgical [12] is an example of a commercially available sophisticated robotic manipulator which translates the surgeon's hand, wrist and finger movements into precise, real-time movements of the surgical instruments inside the patient.

There are numerous methods of manipulating the robotic arm remotely. The basic method would be using remote controller sensors such as joystick, capacitive sensor keypad, rotary encoders, and potentiometers. The research in this area is ongoing and aims at minimizing the input required from the operator in order to strengthen the human-machine interaction. Most commonly, data gloves are employed to control the joints of robotic hands through the use of flex sensors. Other, more advanced data gloves employ acoustic, resistive or magnetic induction sensors to track the motion of the phalanx [13].

1.4.2. Master-Slave system

Human hands are capable of doing skilled dexterous manipulations with high precision and can grasp objects with high grasping forces. These characteristics make them excellent manipulators. Skillful manipulations of human hands are often needed in extreme environments, for example in radioactive environments or outer space. Here, human presence can be avoided with effective master-slave tele operator systems.

In the master-slave control methodology, which has been used in this work, the slave robot (tele operator) replicates the movements of the master. Methods for controlling master-slave robot systems may be divided into two categories – unilateral control system and a bilateral control system [14]. In a unilateral control system, conceptually shown in Fig 1.1(a), no force feedback is available from the slave unit. The only form of feedback to the master unit operator is in the form of vision data. Such a system has the merit of having a simple controller and mechanism; however dexterous manipulation is difficult. Fig 1.1 (b) shows a bilateral control system in which a force feedback signal, usually electrical, is available from the slave to the master control unit. Although the controller and other mechanisms become more complex, dexterous manipulation is possible using such a bilateral system.

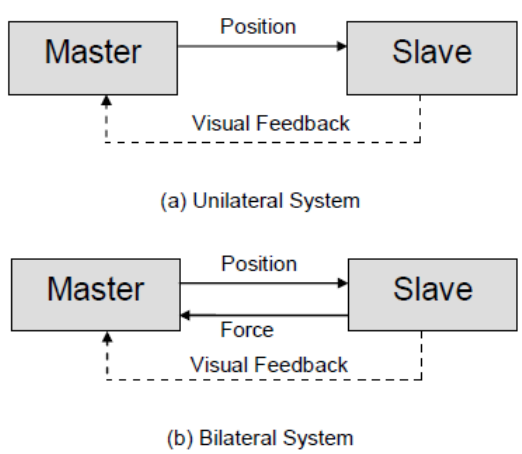


Figure 1.1: Concept of Master-Slave control system [15]

1.4.3. Need and importance of haptics technology for grasping tasks

Haptics is an emerging technology that is growing rapidly. Haptic interfaces are a particular group of robots that are attractive due to their ability to safely interact with humans.

The word haptics refers to the capability to sense a natural or synthetic mechanical environment [16]. Haptic interfaces enable man- machine communication through touch, and most commonly in response to user movements. Also, it provides simultaneous information exchange between a user and a machine. A complete haptic interface usually includes one or several electromechanical transducers (sensors and actuators) in contact with a user in order to apply mechanical signals to distinct areas of the body and to measure other mechanical signals at the same distant areas of the body. These signals could refer to forces, displacements or a combination of these and their time derivatives [16]. Haptics lend themselves to being excellent candidates for improving the level of activity and related senses for the elderly.

Force feedback is classified into two categories- Passive and active. Active and passive force feedback can be differentiated by whether or not ENERGY is added to the system by the controller. Active force feedback controllers apply forces to the user by adding energy into the human-machine system. Passive force feedback controllers apply forces to the user by removing energy from the human-machine system. For example, an active controller might use servo motors to generate feedback forces. The strength of the forces could be directly regulated by the computer which regulates power to the motors. A passive controller might use energy dissipation elements such as a friction brake or a magnetic particle brake. These devices can not directly apply forces to the user, rather they can only apply resistance to the user's motion. The advantage of active force feedback control is that it is inherently general. When using active elements such as servo motors, the system can produce any general force sensation. The advantage of using passive force feedback control is that it is inherently stable and inherently safe for the user. This is because energy dissipation elements only resist motion but do not induce motion. Thus the tradeoff between active and passive is a tradeoff between performance and safety [17]. The passive approach to interaction control is very simple and cheap, because it does not require force/torque sensors. The response of passive compliance mechanism is much faster than active repositioning by the computer control algorithm. However, since no forces are measured in the passive feedback approach, it cannot guarantee that the high contact forces will never occur [17].

The following sections provides a survey of the relevant related research regarding existing assistive devices.

Chapter 2 Literature review

2.1 Existing assistive devices for grasping objects

There are numerous assistive devices in the market that enables the user to reach and grasp objects. One example is MANUS arm developed by Exact dynamics Inc. [7] which is a wheelchair mounted robotic arm that allows a variety of activities to be carried out on a day to day basis. The MANUS can be operated using a wide range of input devices that include, but are not limited to, a keypad (sixteen-buttons in a 4x4 grid), or a joystick (e.g. the joystick of the wheelchair). This has a payload capacity of 3.3 lbs [18].

MANUS has also been used by many research groups to improve manipulation. Kim et al. [8] developed a system for integrating various processes needed for end to end implementation of the assistive robotic manipulator, MANUS. The primary objective of this research was to design an easy to use modular assistive robotic system, UCF-MANUS as shown in Figure 2.1, that can be utilized for a robust operation in unstructured environments without any major revision to its structure, regardless of the user's level of disability, preference, or residual functional ability. The system was tested in a simulated environment of daily living tasks to quantitatively evaluate the absolute and relative efficacy of the various control modes and input modalities at the interface. The results of this research indicated that users experienced difficulty to manipulate the robot in manual mode as it required numerous command inputs.

The pick and place tasks carried out using the interfaces: Cartesian control using 16-button keypad and Cartesian control using TS and GUI (manual mode) are very time consuming and requires lots of effort from the user in terms of manipulation. Mechanical design is not suitable for reaching the top shelves and would be unsafe for the user to manipulate the robot in the manual mode.

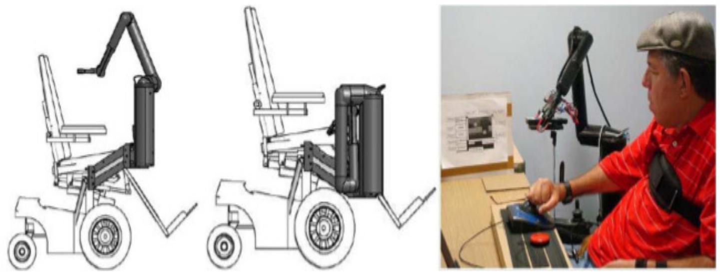


Figure 2.1: Left and middle: Manus Arm in folded out and folded in configuration; Right: MANUS under user control-picking up a remote control from shelf at about a coffee table height using a trackball switch combination [8].

2.2 Ongoing research in teleoperation and haptics

A haptics based assistive device prototype was developed by Gohil et al. [18]. It was controlled using a human arm by means of haptics technology. A haptic glove was developed that fitted over the user's hand like an exoskeleton with potentiometers installed on finger and wrist which picked up the change in resistance with hand movement. In other words, the sensors on the haptic device worked as transducers and converted hand motions into electrical signals. These hand movements were replicated using a robotic arm as shown in figure 2.2. However, communication lag, safety aspects and data transmission from the robotic arm to the human arm has not been discussed.

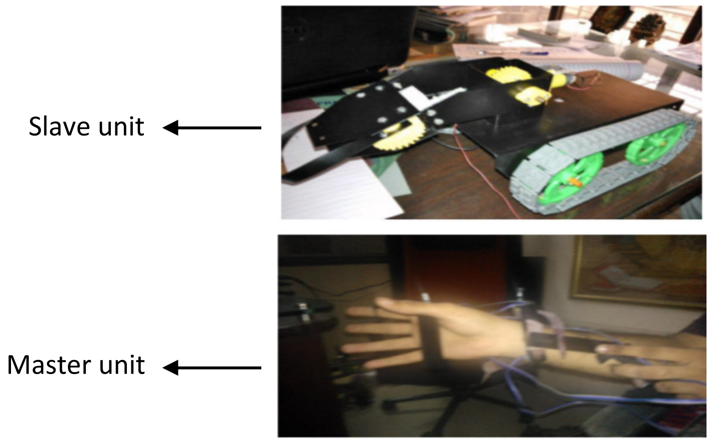


Figure 2.2: Robotic arm and haptic glove [18]

Minas et al. [19] developed a similar system for teleoperation and tele manipulation with the five fingered robot hand DLR/HIT II. A Cyberglove II was used to capture human hand kinematics and a modified version of the joint-to-joint mapping methodology was used to map human to robot hand motion. A force feedback device based on RGB LEDs and vibration motors that can provide real-time feedback of the forces exerted by a robot hand was used so as for the user to be able to perceive the forces exerted by the robot fingertips as shown in figure 2.3. Each led in the RGB led has three different color intensity values (one for each color) and was controlled through the Arduino platform. The value of each color ranges from 0 (off state) to 255 (higher state) so in order to create the different color variations, different intensity levels of different colors were fused to enable the user to determine the amount of pressure to be applied on the object. Similarly vibrational motors were used to provide haptic feedback.

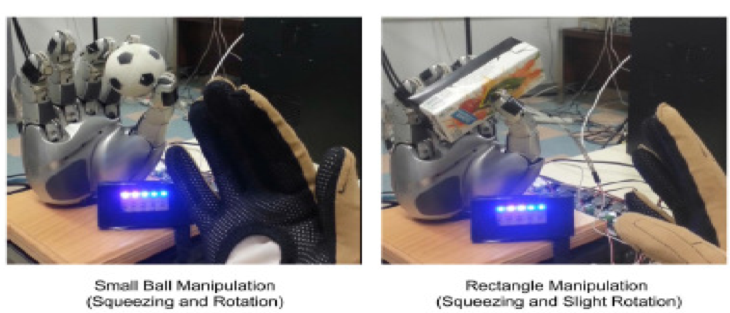


Figure 2.3: 5 fingered robotic hand manipulated through Cyberglove [19]

Lelieveld et al. [20] developed a portable haptic interface with passive force feedback as shown in the figure 2.4. The concept utilizes a mechanical tape brake at the rolling-link mechanism (RLM) for passive force feedback. The brake of a desired joint is actuated to lock the movement of the corresponding RLM when a virtual object is touched. The locking of the relevant RLM is dependent upon the phalange that touches the object. Due to the locked RLM, no rotation of the relevant finger joint is possible. The circular links make it possible to exert a perpendicular force on the finger phalanges during the complete flexion and extension motion. The operator will sense a virtual object. The brake is released when the operator moves away from the virtual object. The brake allows some extension movement of the finger, due to the low stiffness of the flexible outer cable. Therefore, the operator can move the finger away from the object. The brake is released when a rotation of 0.5° is sensed at the potentiometer located at the brake drum for a smooth and low friction movement. Therefore, no strain gauges are required for sensing the force direction to cancel out the braking command, which allows a less complex structure of the device.

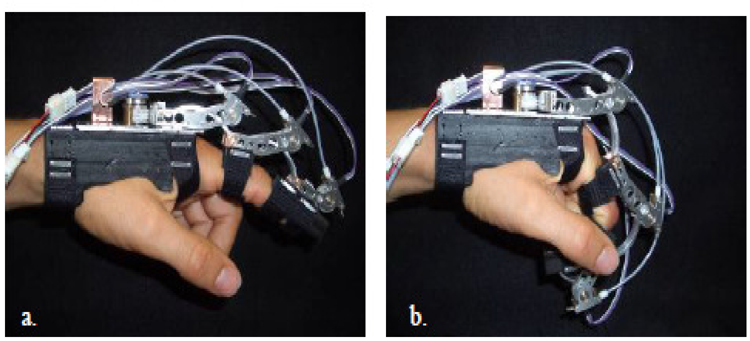


Figure 2.4: Side view of finger with haptic interface in stretched and bent position [20].

Commercially available wheelchair mounter robotic arms (WMRAs) are expensive ($12,500-$50,000), and require the user to drive the wheelchair to the desired object in order for the arm to perform tasks. Users have reported that the size of WMRAs may hinder their ability to reach a table or maneuver the wheelchair through narrow passages [21]. Also, most of the current studies in the teleoperation and haptics field focuses on testing the functionality of the system/device and very limited information has been presented regarding the cost, safety issues, ethical and societal impacts, data communication, and detailed signal transmission data between the master and slave robotic system with an integrated haptics feedback technology.

Chapter 3 Methodology

It is hypothesized that a bilateral master slave robotic system would be a suitable design for this application wherein master unit would consist of a partial wearable exoskeleton which will manipulate a remote robotic arm (slave unit) and in turn obtain a force feedback sensed by the master unit indicating reactions experienced by the slave unit. This hypothesis will be tested through the design and build of a prototype device that integrates all of the desired aspects of a practical device. The constraints for the proposed prototype are detailed in table 3.1.

Table 3.1: Qualitative and quantitative constraints of the prototype

Qualitative constraints	Quantitative constraints
1. How extensive is the input needed by the user?	1. Time taken to complete a given task should be less than 30 seconds
2. Is the device safe?	2. The robotic arm must be able to lift 1 lbs. at the slave unit without tipping over
3. Is the device aesthetically appealing and socially acceptable?	3. The master unit wearable part should accurately manipulate the end effector and in turn obtain force feedback for objects with weights ranging from 0.2 lb to 2 lb
4. Does the device irritate the user or occupy extra space?	4. Success rate of grasping, moving and placing the held object should be 95%
5. Does the device potentially offer therapy benefits?	5. The robotic performance should map to the motion simulations using the kinematic model for accurate control
6. Does the device grip the objects firmly and allow the transfer of objects?	6. The device should be cost effective
7. Is the device upgradeable?	

The following sections describe the mechanical design and the control system setup.

3.1 System design, analysis and verification

3.1.1. Concept design

Based on the previous literature review, it was found that grasping and reaching for things are some of the most challenging and common tasks for the elderly to perform on a daily basis. Therefore an initial concept was developed based on tele-operation which involves manipulation of the robotic device from a certain distance. The design includes a bilateral master and slave system wherein an anthropomorphic slave robotic arm replicates the movements of the master, and the master unit, in turn, receives force feedback for safe manipulation.

The master unit represents the tool and mechanism used by the operator to provide the slave with position commands. Figure 3.1 shows the model of the human arm and various parameters that are mapped to the proposed master unit.

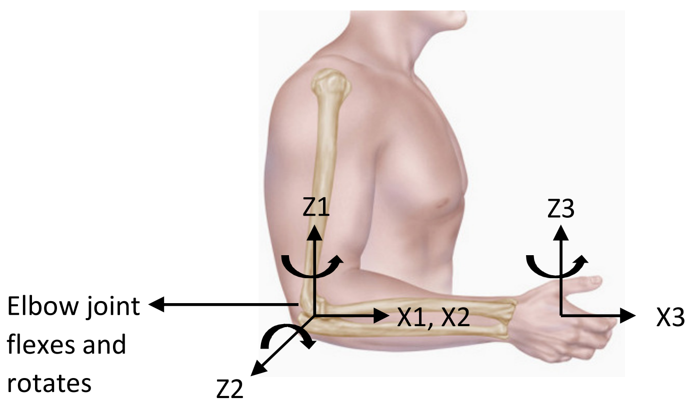


Figure 3.1: Mapping the joints of a human arm to the master unit of the proposed system

3.1.2. Master unit design

The design of the master unit is shown in figures 3.2 and 3.3. The master unit has two parts:

1. Partial hand exoskeleton representing the gripping part of the hand and manipulating the end effector at the slave unit, as well as providing haptic feedback (Figure 3.3)

2. Robotic unit representing the first two joints supporting the hand in a human arm for manipulating the corresponding joints of the slave unit (Figure 3.2)

The partial hand exoskeleton and robotic unit are kept as separate units so that it can be used by users with different arm sizes and also reduce the amount of material and bulkiness. This would ultimately reduce the amount of system load on the elderly user's arm.

The robotic part has two mapping joints simulating the elbow and gripper to reduce the complexity of the system. The position of each of the joints is measured using potentiometers mounted at the axis of rotation of the joint.

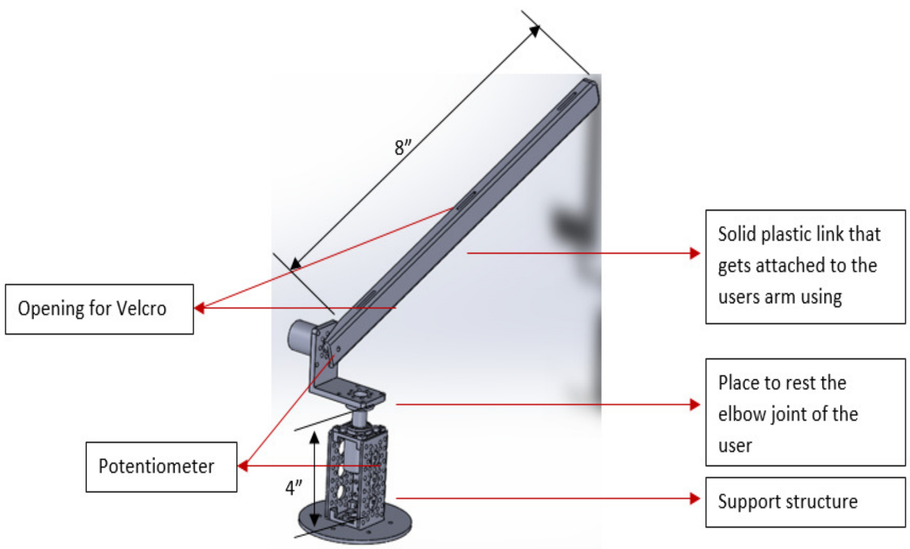


Figure 3.2: Master unit elbow design

The wearable exoskeleton is shown in the figure 3.3. This unit gets strapped on the users hand by means of velcro. The unit consists of fingercap to place the forefinger and a link that is connected to the servo motor which provides the haptic feedback. The flex sensor will be attached on top of the rubber link that passes on the forefinger. This sensor enables the manipulation of end effector by offering resistance change values.

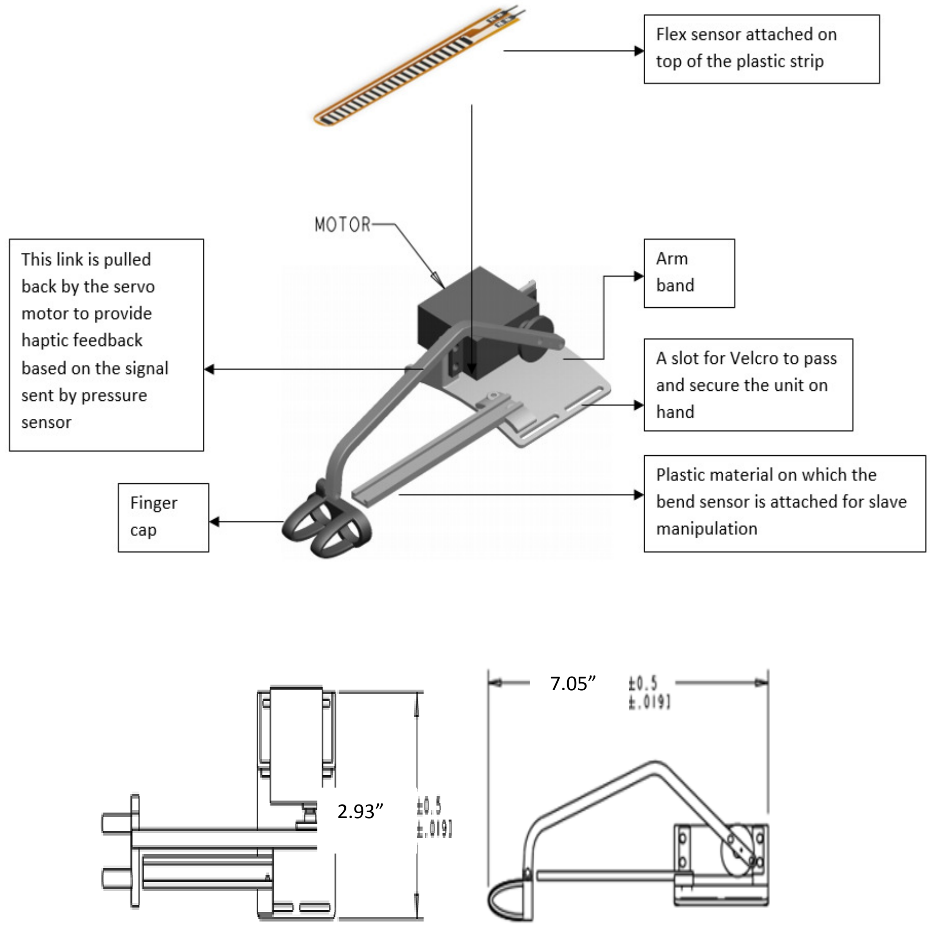


Figure 3.3: Master unit hand wearable exoskeleton

3.1.3. Slave unit design

The design and exploded view of the slave robotic arm is as shown in figure 3.4 as a two degree of freedom robotoic arm with two rotational joints. The angular range for both joints is from 0 degrees to 180 degrees. This unit simulates the front part of the human arm. The robotic arm length is chosen by considering the average human arm length for reaching things on shelves. Anthropometric dimensional data [22] shown in the table 3.2 has also been used for choosing appropriate length of the arm, according to which the average human lower arm and hand length is approximately 16 inches. However, for this length, a higher torque would be required to account for moment arm. Therefore, the length of link 2 is taken as 10".

Table 3.2: Standing and sitting dimensions in meters [22]

Name	Dimension	Male 5th%	Male 50th%	Male 95th%	Female 5th%	Female 50th%	Female 95th%
Stature	A	1.649	1.759	1.869	1.518	1.618	1.724
Eye height(Standing)	B	1.545	1.644	1.748	1.427	1.52	1.63
Mid shoulder height	C	1.346	1.444	1.564	1.21	1.314	1.441
Waist height	D	0.993	1.102	1.168	0.907	0.985	1.107
Buttocks height	E	0.761	0.839	0.919	0.691	0.742	0.832
Sitting height	F	0.859	0.927	0.975	0.797	0.853	0.911
Eye height(Sitting)	G	0.743	0.8	0.855	0.692	0.743	0.791
Upper arm length	H	0.333	0.361	0.389	0.306	0.332	0.358
Lower arm + hand length	I	0.451	0.483	0.517	0.396	0.428	0.458
Upper leg length	J	0.558	0.605	0.66	0.531	0.578	0.628
Lower leg length	K	0.506	0.553	0.599	0.461	0.502	0.546

The length of link 1 is taken as 3.75 inches as this would enable the user in the wheelchair to rest their arm and maneuver the slave unit comfortably. Aluminum channels are used for building the links as it is a light weight material with density of 0.095 lb/in^3 and high tensile strength of 483 MPa as opposed to steel with a higher density of 0.284 lb/in^3. This allows the reduction of the overall weight of the robotic arm while providing excellent strength.

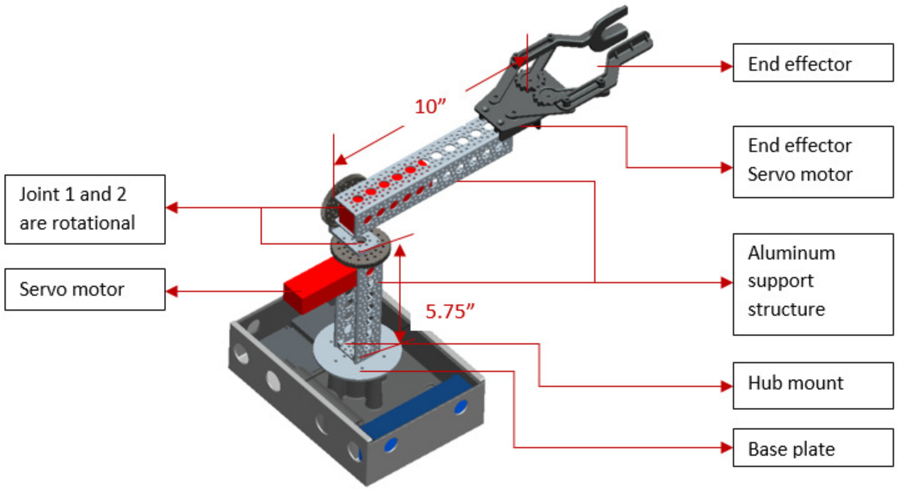


Figure 3.4: Exploded view of the slave part of the robotic system

3.1.4. Modeling

The idea behind this concept was to develop a slave robotic arm that would mimic the human arm motion. As part of the concept development phase of the project, forward kinematic model of the arm was developed for the slave unit of the system. The base of the model is the arm shown in figure 3.5 which is used for the kinematic modeling and analysis. This model allowed for simulation of the device motion as a baseline to verify that the designed parameters allow the device to function as desired. Appendix B includes the details of the model.

Based on the forward kinematics model (Appendix B), workspace covered by this proposed device has been calculated as shown in Figure 3.6. The work envelope enables the visualization and modification of the design according to the needs of the application and restriction of the work environment. This is also very important for the safety of the end user. Moreover, this simulation exposes any possible singularities in the device motion allowing prevention of any possible dysfunctionality of the device.

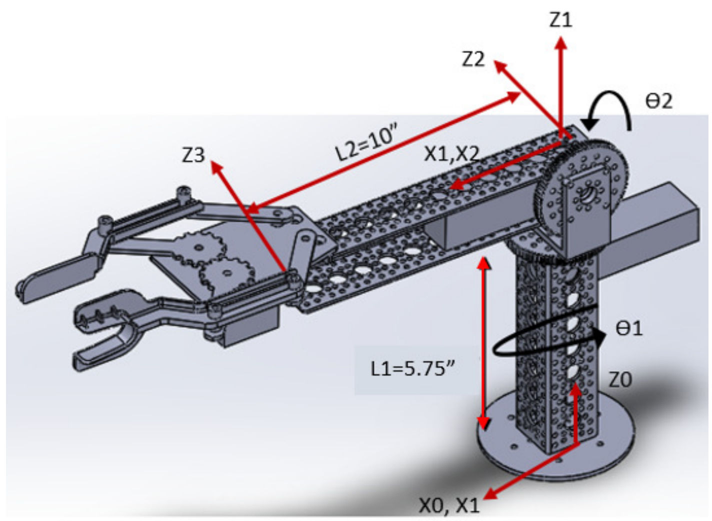


Figure 3.5: Kinematic model parameters of the proposed slave robotic arm

Angular limits at each of the joints was specified in the program for generating the work envelope. Joint 1 and joint 2 of the master robotic unit are controlled by the elbow joint of the human arm. Therefore, there is limitation in extent to which the human arm can rotate about the elbow joint. The angular limit was measured using a digital protractor by resting the arm on the master robotic unit. Five trials were taken by moving the arm in left and right direction. The safe average angular range of motion was determined as -35 degrees to 135 degrees in horizontal direction and 45 degrees to 135 degrees in vertical direction. These values were input into the program for generating the work envelope. The projection of the work envelope in XY, ZX and ZY plane are shown in the figures 3.6, 3.7 and 3.8. The 3 dimensional work envelope is shown in the figure 3.9. The work envelope for the proposed device is a part of a spherical shell with the thickness equivalent to the length of the end effector.

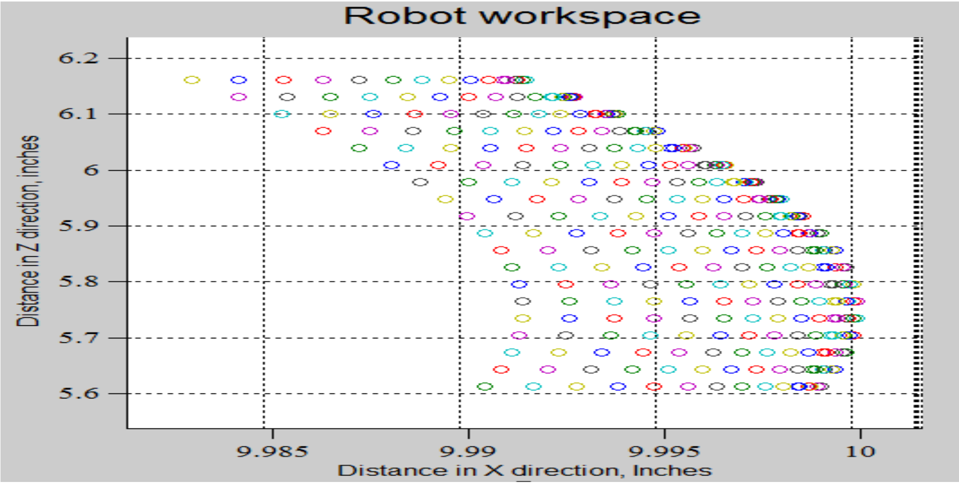


Figure 3.6: Work envelope of slave robotic arm in XZ plane

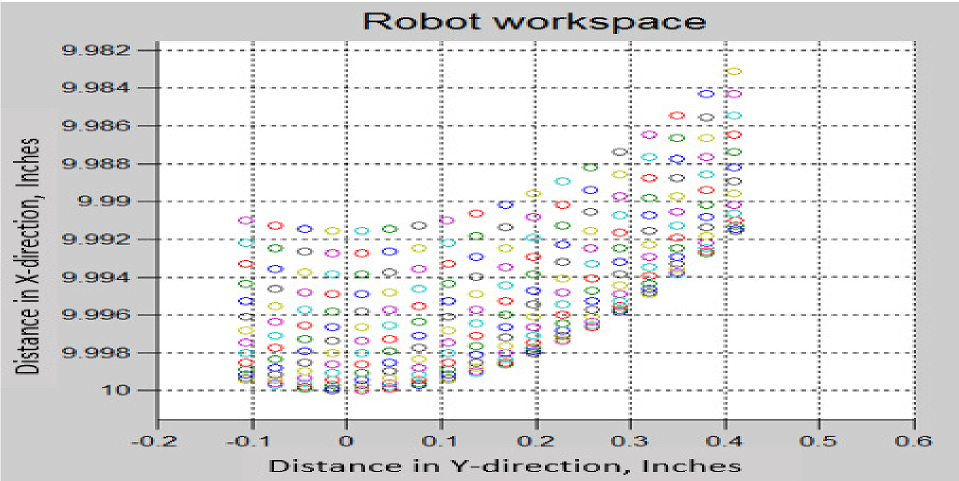


Figure 3.7: Work envelope of slave robotic arm in XY plane

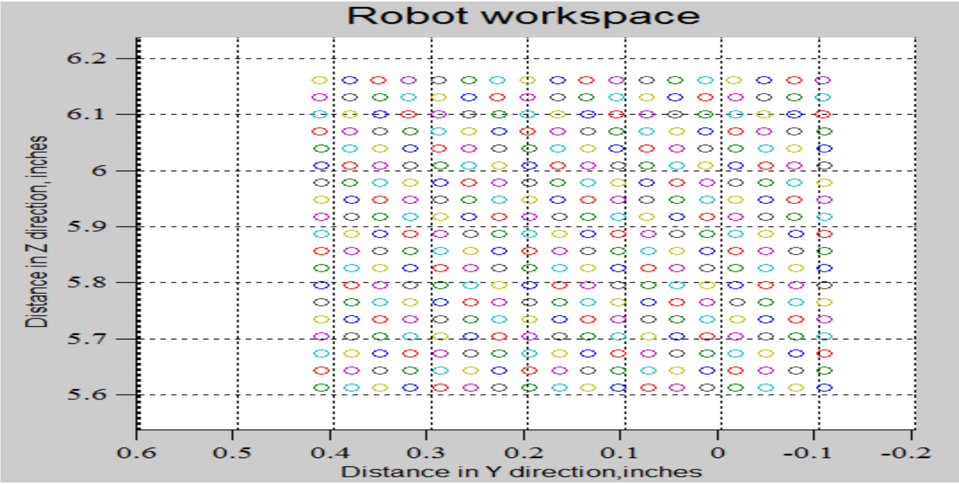


Figure 3.8: Work envelope of slave robotic arm in ZY plane

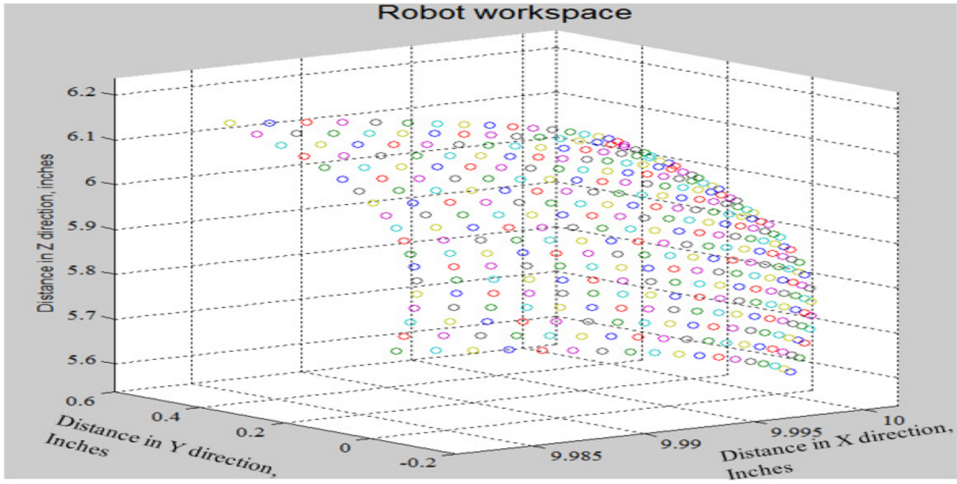


Figure 3.9: 3D view of work envelope of the proposed device

The inverse kinematics model was also derived in Appendix B using the forward kinematics model. The inverse kinematics models allows the calcuation of the vector of joint values corressponding to a designed end effector goal state.

The joint angles, $\theta 1$ and $\theta 2$ were calculated and are as follows:

$$\theta 2 = Atan2(\sin(\theta 2), \cos(\theta 2)) \quad (3.1)$$

$$\theta 1 = atan2(y, x) \quad (3.2)$$

Where x and y are the cartesian space values.

Hard coding these equations into software allows the joint angles to be calculated in real time as x-y coordinates are given to the controller. This inverse kinematic model can be used to programmatically provide the information for how to move each joint of the prototype to achieve a preset x-y position for the user hand relative to the location of their elbow.For control purposes, this would be an ideal break down method of common commands by actuator of where the actuator should move to, and could be used to, create smooth motion profiles as desired, within the microprocessor.

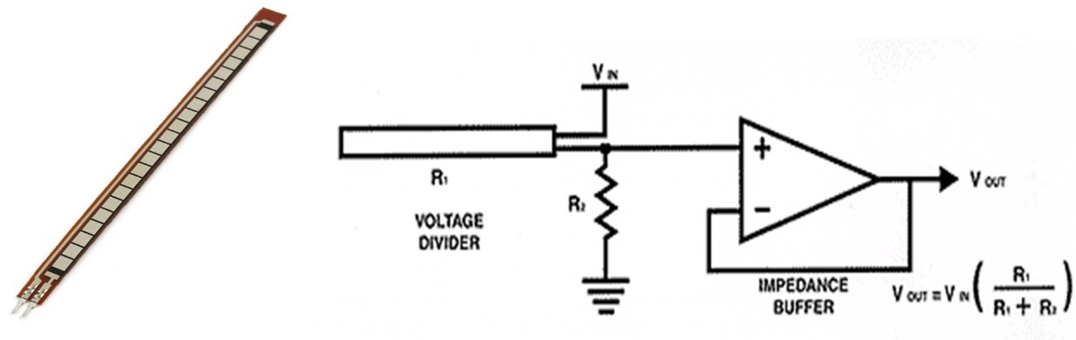
3.2. Control system and electrical circuitry design

The literature review in section 1 shows numerous methods for manipulating the robots remotely such as joystick and teach pendant. Similarly, a variety of sensors are used for providing haptic feedback for various applications. Since the goal of this research was to develop a cost effective assistive device, off-the-shelf electronic components were used to build the prototype. The following sections summarize the overall system architecture and main features of the components used.

3.2.1. Flex sensor

Since the proposed technique for manipulating the end effector is an exoskeleton glove, flex sensors seem to be a perfect match for this application as they can run on top of the fingers and offer variable resistance when the fingers are moved back and forth. Flex sensors (such as strain gages) mounted inside the gloves generate an electrical signal proportional to the bending amount of each phalange. Phalanges are the bones that makes up the fingers of the hand and toe. A computer interface incorporated in the loop, converts these signals into angular measurements which are then communicated to the robotic hand to mimic the gestures.

One side of the sensor is printed with a polymer ink that has conductive particles embedded in it. When the sensor is straight, the particles give the ink a resistance of about 30k Ohms. When the sensor is bent away from the ink, the conductive particles move further apart, increasing this resistance to about 50k Ohms when the sensor is bent to 90º, as in the figure 3.10. When the sensor straightens out again, the resistance returns to the original value. By measuring the resistance, the corresponding angle of the motion can be determined.



(a) (b)

Figure 3.10: 4.5” flex sensor (a) and basic electrical circuit (b) [23]

3.2.2 Flexi force sensor

There are numerous methods of producing feedback in a mechanical system. One basic method would be to place a precision spring in the system and measure the detection [24]. More advanced methods would include placing a load cell or strain gage in the system and reading the resulting force. The method that is especially popular in haptic systems is applying carefully controlled current to motors. Current can be directly calibrated to torque output from a given motor through mathematical formulas [24]. One device that shows promise in the field of haptics, but so far has been used in very limited applications, is the force sensitive resistor (FSR). The operation of a force sensitive resistor (FSR) is analogous to a piezo-electric device. Applying a force to a FSR changes the resistance of the device. FSRs can be considered normally open where, at rest they exhibit an extremely high resistance [25]. As a load is applied the resistance of the sensor decreases. This relationship is shown by a typical plot in figure 3.11. The change in resistance can be converted to force through formulas specific to the FSR and the driving circuitry based on actual calibration.

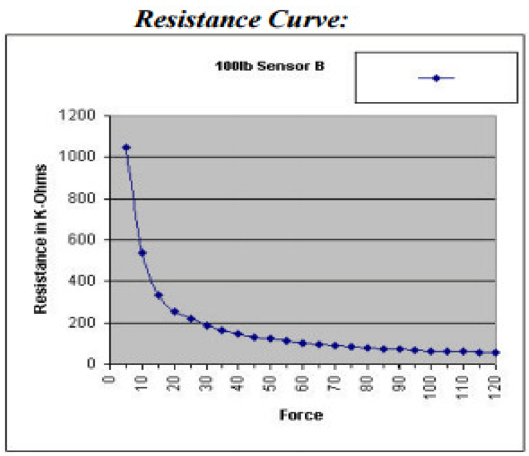


Figure 3.11: Calibration curve for Flexi force sensor [25]

FSRs are manufactured in a wide variety of form factors and load ranges [4]. Strip FSRs are long, trimmable strips that allow sensing along the entire length of the strip. More advanced strip FSRs offer feedback as to where along the strip the load was applied. Certain manufacturers also offer square FSRs with a large, square sensing area. These sensors provide x-y coordinates to define where the force was applied and are gaining popularity as sensors under touch pad type computer mice. One model FSR, manufactured by Tekscan, is shown in figure 3.12. Tekscan produces an FSR called the FlexiForce. This FSR is a thin film resistor with a thickness of 0.008 inches [25].

The FlexiForce FSRs are constructed of two layers of polyester film, a layer of silver applied to each, and pressure sensitive ink between the silver layers. The silver layer extends to the end of the resistor, which can be trimmed to any length [25]. A schematic diagram of the construction of the FlexiForce FSRs is shown in figure 3.13. Tekscan recommends that the FSR be placed in a driving circuit as shown in figure 3.12. The FSR is driven by an excitation voltage, the force output is determined through a combination of this voltage and the reference resistor

RF. In the case of the driving circuit reducing RF will decrease the sensitivity of the circuit and increase the maximum force the sensor can read [25].

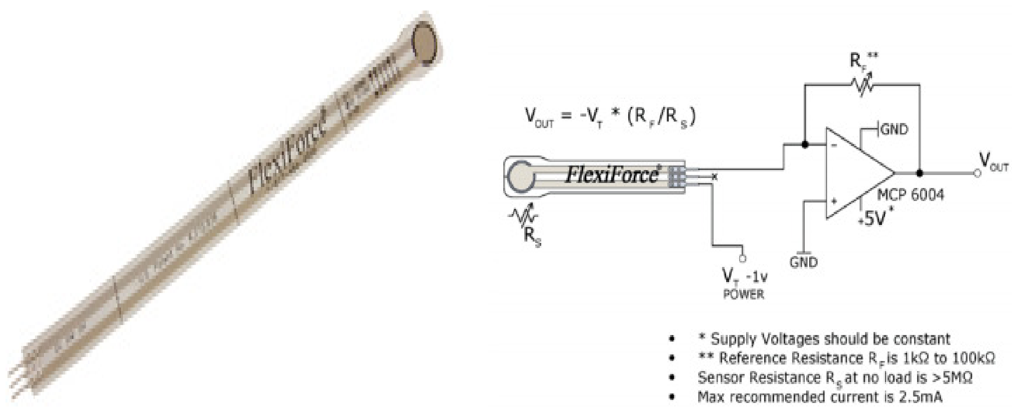


Figure 3.12: Flexiforce sensor and basic driving circuit [25]

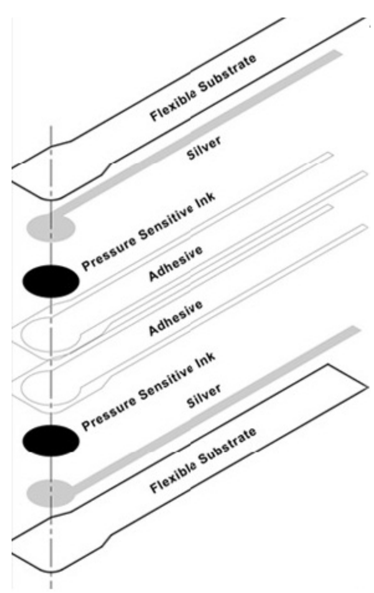


Figure 3.13: FlexiForce sensor construction [25]

The FlexiForce sensors were chosen for this project primarily because of their sensing area and their relatively inexpensive cost. The silver sensing pad in the FlexiForce sensor is a circle with 0.375 inches in diameter. Unlike strip FSRs or square FSRs of comparable price and performance, the FlexiForce FSRs allow a much smaller and controlled sensing area. The most important area of force application is the end effector gripping surface. Because the area of interest is well defined it is suitable to use an FSR with relatively small sensing area.

Obtaining accurate force readings from any FSR requires calibration and testing to account for the driving circuitry. Tekscan recommends a conditioning and calibration procedure for their FSR products, found in the FlexiForce user manual [26]. Tekscan recommends using this calibration procedure for new FlexiForce resistors as well as resistors that have not been used for a significant period of time. This procedure will reduce drift and hysteresis effects on the FSRs [25].

3.2.3. Transistor for haptic motor control

A TIP 120 NPN (Negative-Positive-Negative) transistor is used as a switch to control the power supplied to the motor. The design and electrical schematic of this transistor can be found in the product manual [27]

The motor that provides haptic feedback is attached to a link on the partial hand exoskeleton. When the master unit is powered, this motor locks its position and continues to remain in that position until it receives a PWM signal from the microcontroller. This hinders the ability to move the hand. Therefore, a NPN transistor was introduced to control the power supplied to the motor. It is programmed in such a way that the motor turns off when the hand unit is powered and allows the hand to move freely. The motor turns on when it receives a signal from pressure sensor located on the slave robotic unit and provides a resistive force feedback to the user's hand by pulling the link by a small angle of 2 degree and then switches off again. The hand can keep moving further even when the haptic feedback is ongoing. This is controlled in the program to be repeated at a certain frequency based on the pressure intensity to mimic vibration feedback (appendix E).

3.2.4. Actuators

The proposed slave unit consists of a 2 degrees of freedom robotic arm. Both degrees of freedom is achieved through a rotational joint and therefore requires motor to drive the links attached to the joints. Servo motors were chosen over other motors due to its ability to move to fixed positions based on the input signals from the sensors. Servo motors can position the motor shaft at a specific position (angle) using control signal. The motor shaft will hold at this position as long as the control signal is not changed. This is very useful for controlling the robot arms, or any object that is required to move at certain angle and stay at its new position [27]. Since servos are fully self-contained, the velocity and angle control loops are very easy to implement, while prices remain very affordable [28].

Numerous parameters like range of motion and velocity were analyzed for selecting appropriate servo motors for two joints of the slave robotic arm and is shown in the table 3.3. The rationale for choosing these ranges are detailed in section 3.1.3.

Table 3.3: Joint 1 and joint 2 angular range of motion

Joint 1 angular range of motion	-35 degrees to 135 degrees
Joint 2 angular range of motion	-45 degrees to 135 degrees

Caroline et al. [29] carried out studies to document the major changes that occur in the control and coordination of movement with respect to aging. It has been shown that the reaction time and movement time of older adults are slower than those of young adults and older adults produce movements with 30-70 percent lower peak velocity compared with young adults. Therefore, based on this, average velocity of the arm was determined as 70 degrees/sec by resting the arm on the master robotic unit and a 30% velocity was deducted from this value to account for lower movement time in elderly. Therefore, the resulting velocity of 49 degrees/sec was considered as a safe speed range for and elderly person.

Torque and power information was further analyzed to decide the type of motor and the calculations are detailed in Appendix A. The torques at joint 1 was determined to be 56 kg-cm including the factor of safety. Based on the motor manufacturer recommendation, a safety factor of 2.0 was applied. This led to selecting a motor: HS 7950 (3:1 gear ratio) and the specifications are detailed in table 3.4. The speed of this motor was found to be sufficient for mimicking the elderly human arm motion.

Table 3.4: Specifications of HS 7950 motor

Speed of the motor/60 degrees	0.39 seconds
Torque	1206 Oz-in at 6V
Speed of the motor/second	153 degrees

Figure 3.14: HS-7950 servo motor [30]

Gripping force is the maximum effort applicable by the end-effector. This force is normally used for the claw grippers which are the end-effectors, representing the force that the fingers can apply on a part. The slave robot gripper applies this force on the object to retain it from slipping, especially during movement.

The calculation of the gripping force and its limitations that the robot gripper must apply depends on the mass of the part that must be moved, the friction coefficient between the finger material, part material, and the gravitational acceleration constant [31].

To make sure the part doesn't slip during static pretension, the gripping force for the proposed end effector design was calculated as 0.89 Kgf including a factor of safety of 1.2. The gripper torque was determined as 12.46 Kg-cm. Detailed calculation of gripping force and torque are shown in Appendix A. Based on these results, a 15 Kg-cm torque analog servo motor from Pololu was chosen (Figure 3.15). The specifications of this motor are detailed in Appendix H. This motor is also used for providing the force feedback on the master partial hand exoskeleton as the resisting force on the end effector should be proportional to the force experienced by the master unit.

Figure 3.15: 1501 MG analog servo motor [32]

3.2.5. Power supply

The motors for joint 1 and joint 2 of the slave robotic unit are rated at 6-7.4V and 4.8 A. The motor for end effector joint and haptic feedback are rated at 6V, 3 Amps. Due to varying current and voltage rating, Venom Ni-Mh rechargeable batteries were used to power the motors and is shown in the figure 3.16.

Figure 3.16: 6V, 3A Ni-mh battery (left) and 6V, 5A Ni-mh Venom battery

3.2.6. Master-slave parts wireless communication

To eliminate possible wiring issues including safety and possible maintenance, and to allow for maximum flexibility, a wireless communication method was chosen for this system. The wireless communication is performed using a specialized chip, nRF24L01 (figure 3.17). This is an inexpensive single chip 2.4 GHz transceiver with an embedded baseband protocol engine, suitable for ultra-low power wireless communications. One of this chip is mounted on the master unit and one on the slave unit for a two way data communication. The maximum distance range covered by this chip is 100 meters. This chip was chosen as Arduino has libraries that supports the function and also covers a wide range of data communication distance of 100 meters which is well above the required specification for the prototype. The pin configuration and block diagram for this chip can be found in product manual [33].

Figure 3.17: nRF24L01 wireless module [34]

3.2.7 Controller and software interface

Arduino uno is chosen as a main controller interface for the assistive device. All the electronic components are connected to this microcontroller for data communication. Detailed description and specifications of the controller are provided in appendix I. One Arduino is used to control the components on the master unit and one on the slave unit. This enabled the separation of the master and slave unit for wireless communication. Arduino also has libraries to control servo motors, wireless transceivers and sensors.

A protoshield (Appendix I) was used to facilitate prototyping as it allows for easy connections between a breadboard and an Arduino. The protoshield provides Arduino with easy access to a prototyping area, two general use LEDs, access to a BlueSMiRF socket, a general use push button, and most important of all the Arduino reset button. This shield sits on top of the Arduino and offers extra pins and eases the process of soldering.

The complete system architecture and the data flow between different components is shown in the figure 3.18.

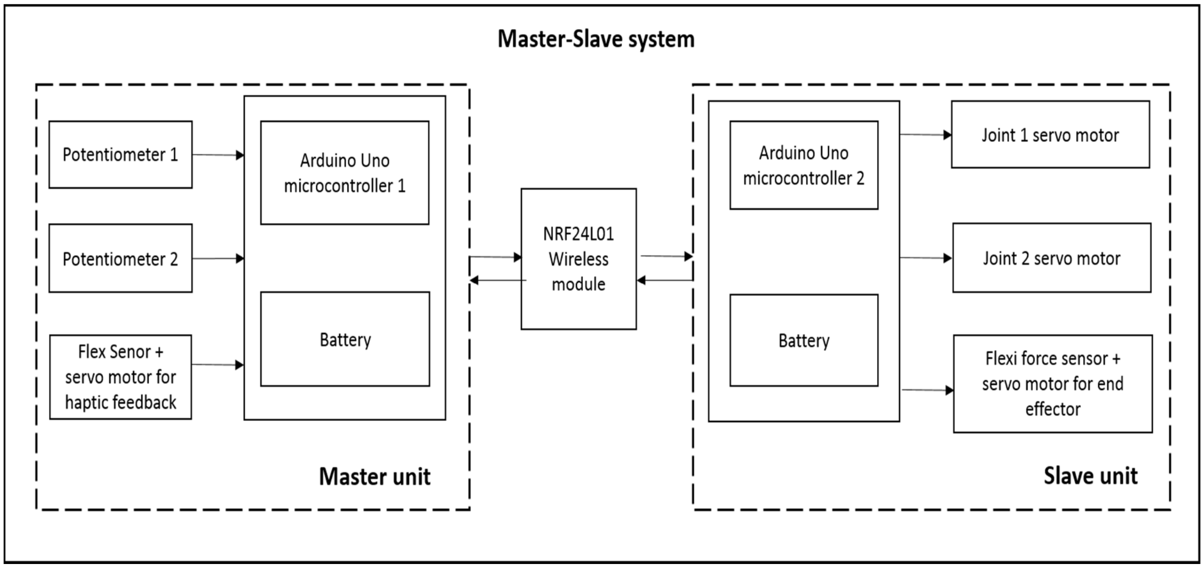


Figure 3.18: Complete system architecture of the proposed haptic device

Data and signal transmission between different components and units for transmitter and receiver is shown by detailed algorithms in the figure 3.19 and 3.20.

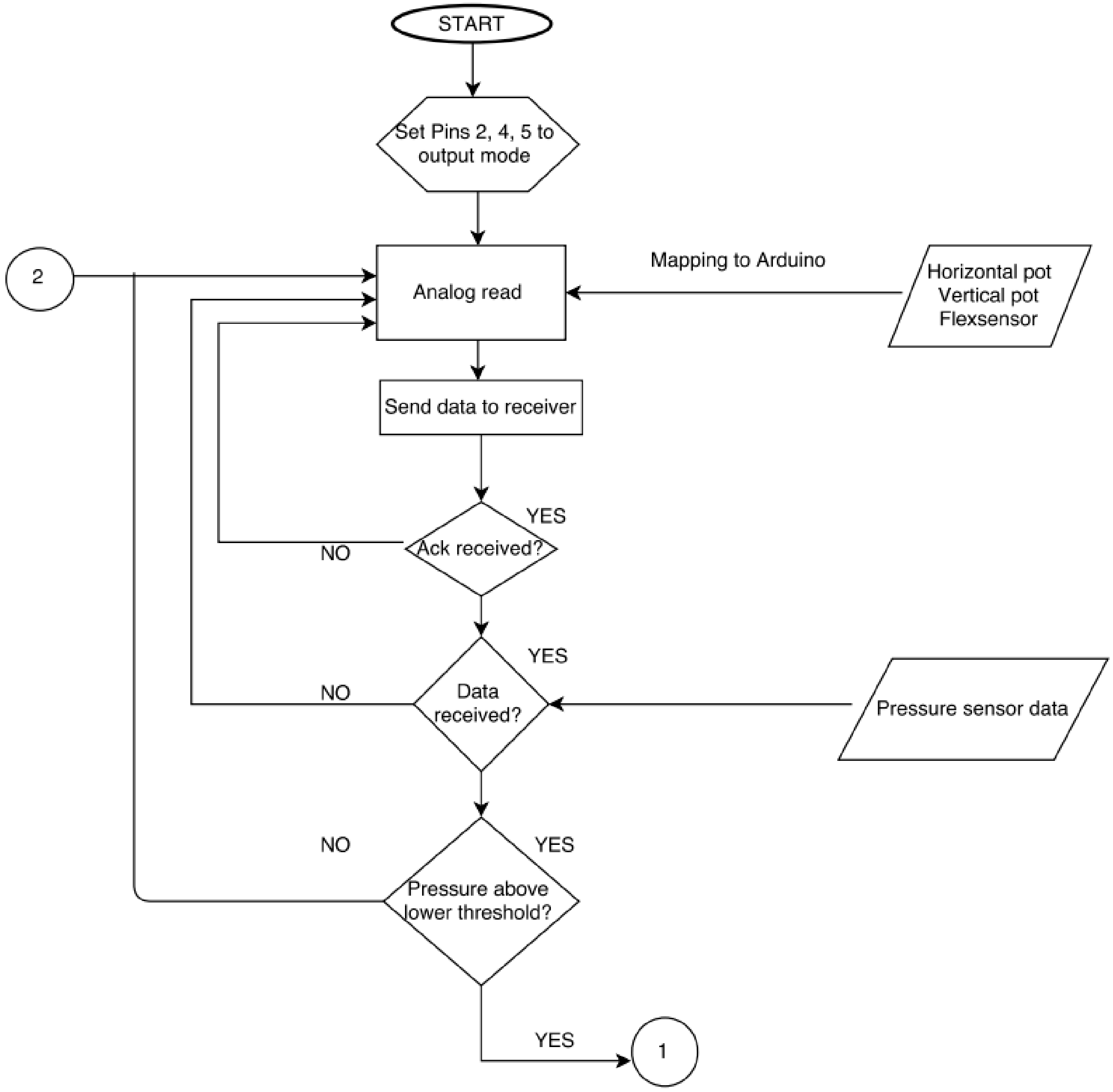


Figure 3.19: Flowchart of transmitter program logic

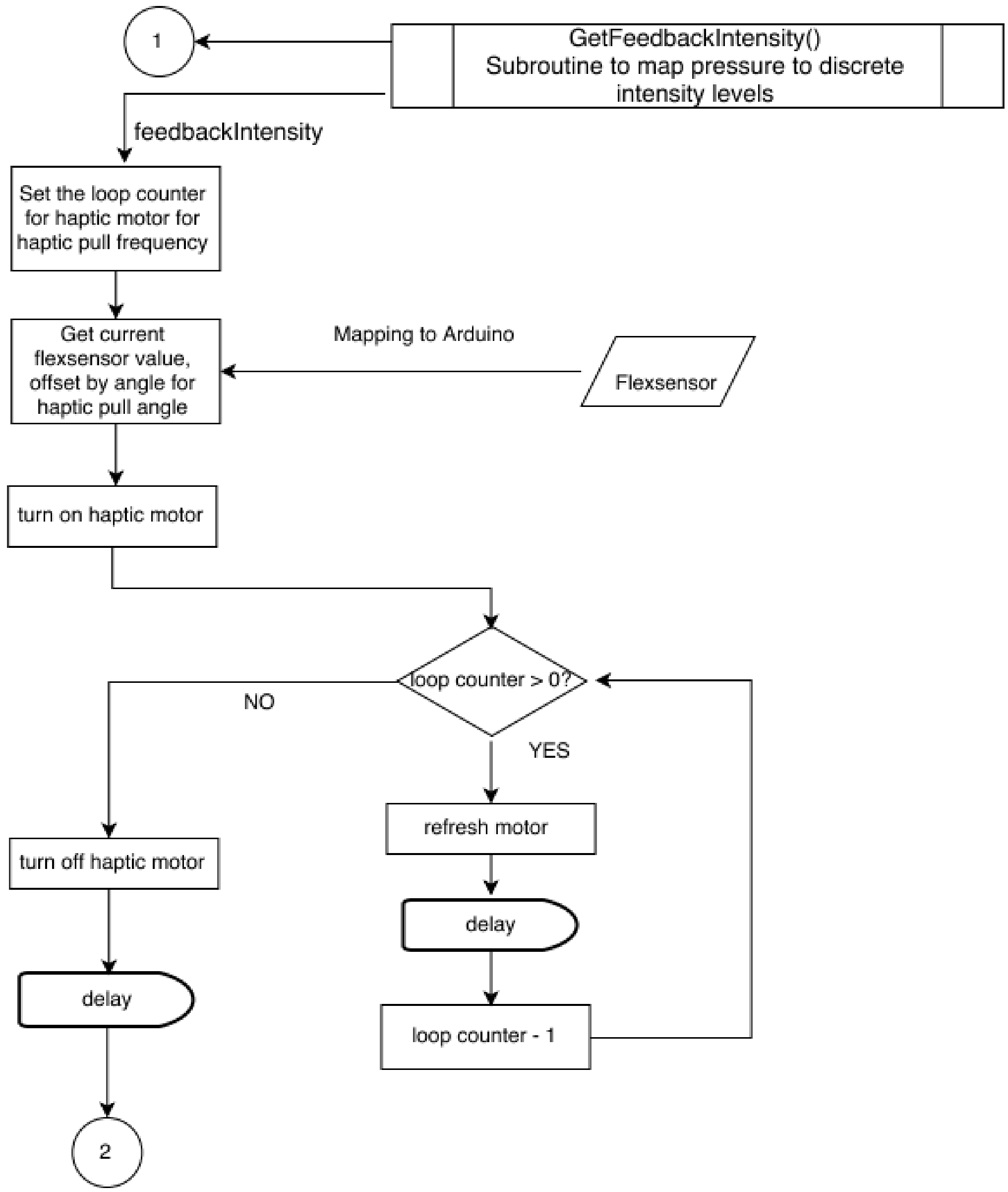


Figure 3.19: Flowchart of transmitter program logic (Continued)

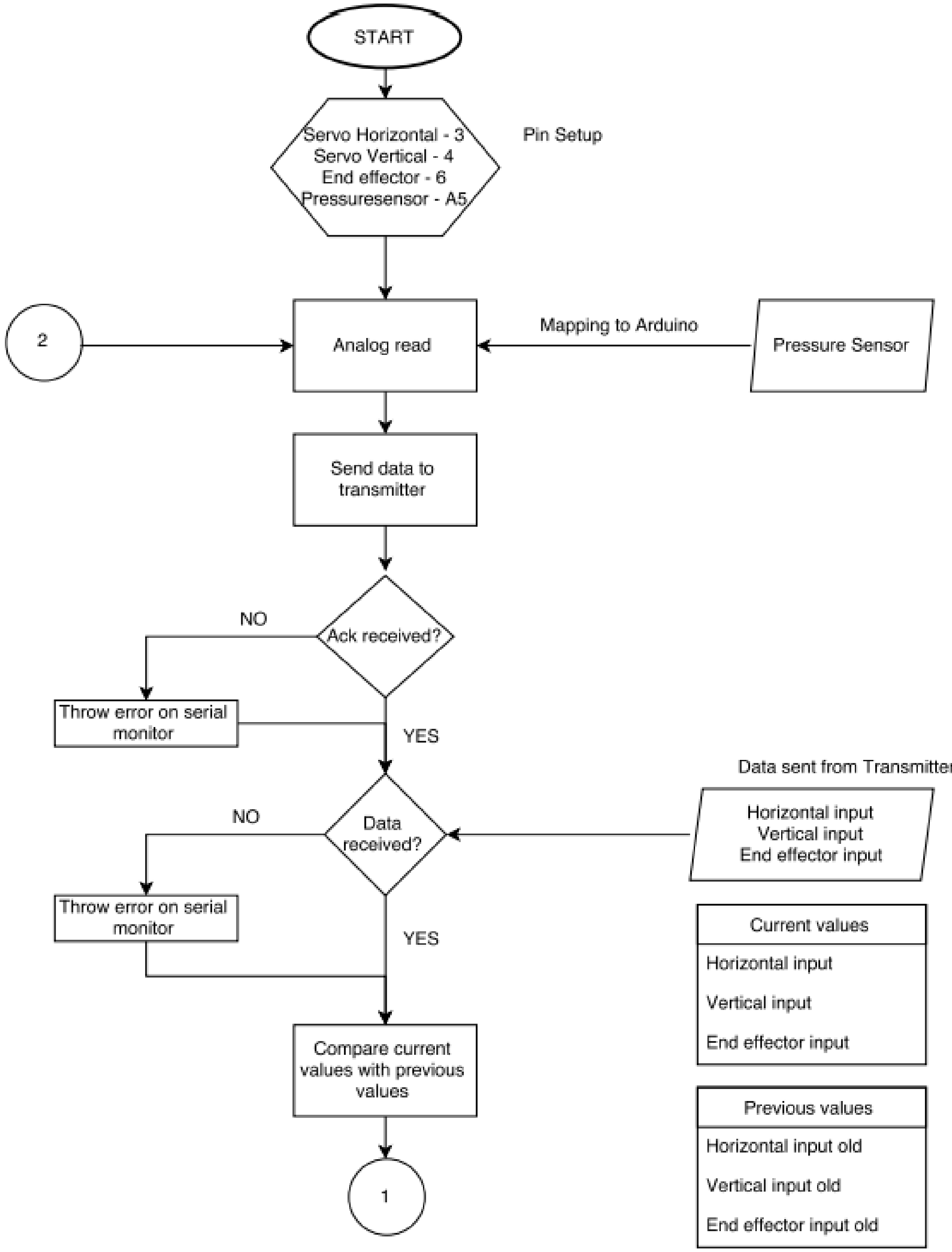


Figure 3.20: Flowchart of receiver program logic

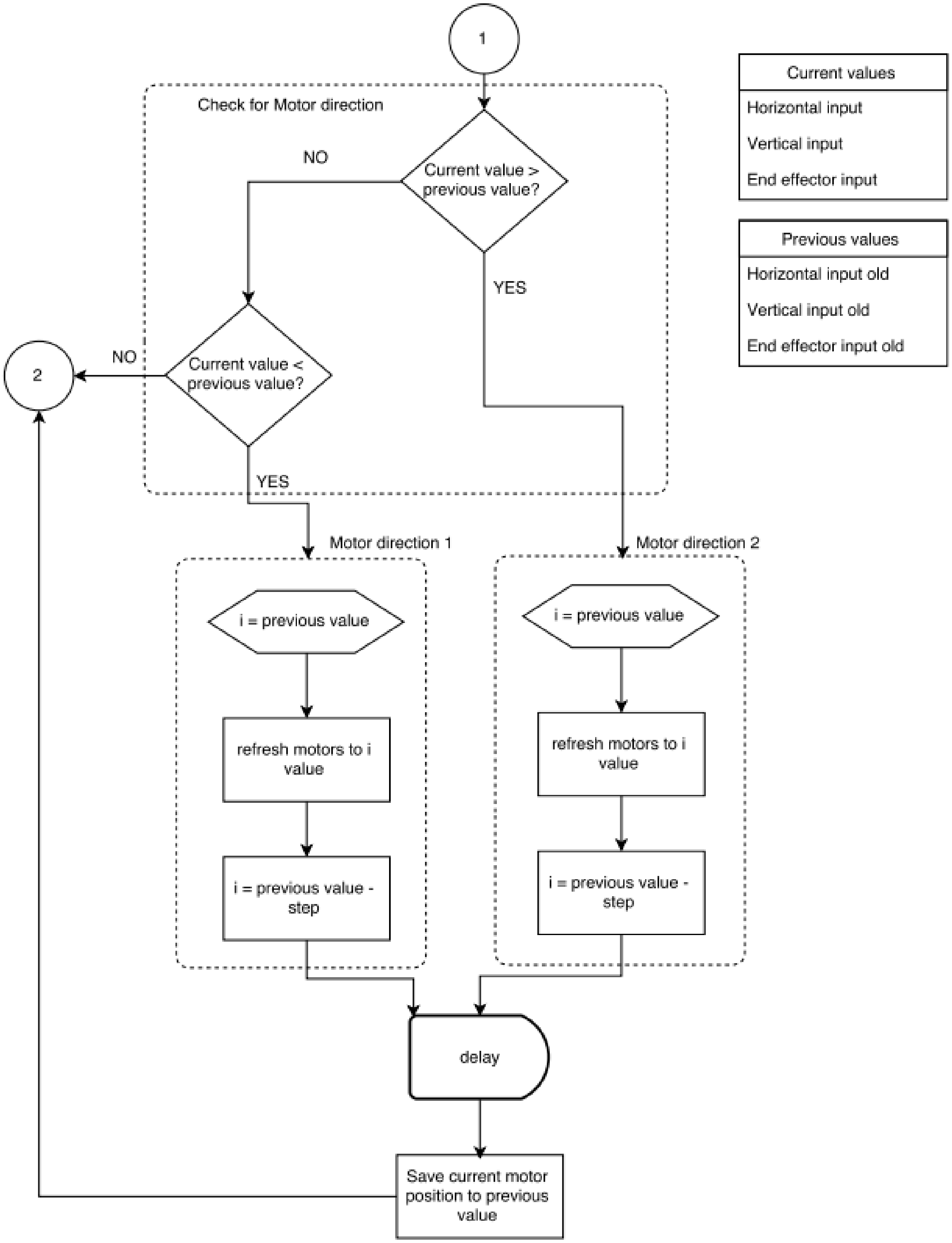


Figure 3.20: Flowchart of receiver program logic (Continued)

3.3 Proof of concept and physical prototype

3.3.1 Proof of concept

A prototype system for testing the teleoperation and haptic feedback was built as a proof of concept as shown by figure 3.21. The schematic of the electrical circuit for this proof of concept was created using Eagle software and is described in appendix D. The master unit is a glove which has the conductive rubber stitched on top of it, which when stretched, changes resistance causing rotation of the motor attached to the end effector, proportionally. The flexi force sensor is attached to one of the links of the end-effector which when holding an object, senses pressure and sends the signal to the micro servo motor attached to the glove around the wrist. This concept has been videotaped [35] which clearly demonstrates the functionality of both flex sensor and pressure sensor. Arduino micro controller was used for controlling all the components on the master and slave unit. The end effector was designed based on the four bar mechanism principle. A prototype was built for testing purposes using acrylic material. It was found that the design worked well for holding objects.

Building the prototype enabled testing of the required functionality to determine various glitches that can be treated before developing the actual system. Some of the things that were observed from this prototype were:

1. The glove was made of fabric and therefore did not stay firm on the hand.
2. The conductive rubber was not very sensitive. It had to be stretched to a greater extent for obtaining a small resistance change. This made the maneuver of the end effector challenging.
3. Only one microcontroller was used for controlling the system. This led to clumsy

wiring which in turn restricted the hand motion.

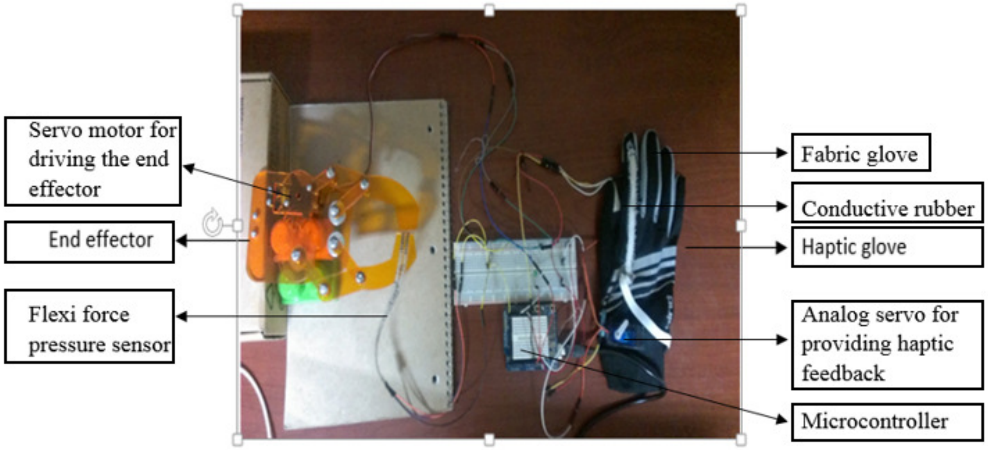


Figure 3.21: Proof of concept for the proposed assistive device

3.3.2 Physical system of master-slave unit

The physical prototype of the complete system is shown in the figure 3.22. The master and slave units are described in detail in the following sections.

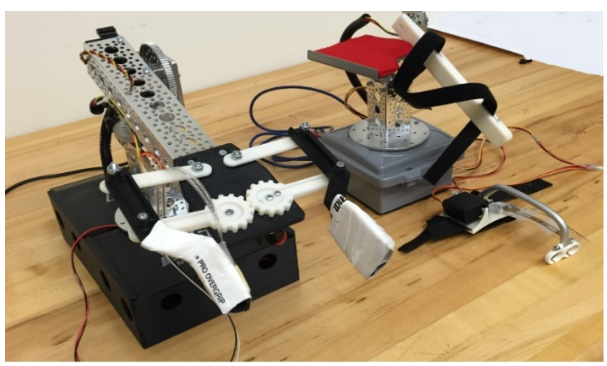


Figure 3.22: Physical system of the master-slave unit

3.3.3 Physical system of Master unit

Based on the experience obtained from the prototype, the proposed system was built. The master unit is shown in the figure 3.23 and the slave unit is shown in the figure 3.27. The master unit includes two rotating joints. A potentiometer has been attached to each of these joints. A

0.5” bore flat bearing mount is placed on top of joint 1 to obtain the swivel motion. A mounting hub is placed on top of this and is secured to the potentiometer using a set screw, as shown in the figure 3.34. Similarly, the knob of the potentiometer at joint 2 is attached to a 3D printed plastic part which in turn will get attached to the arm of the user through Velcro. The two joints are connected using a L bracket. The elbow of the user is placed on the L bracket base and can move sideways or up and down. Aluminum channels with holes are used as a supporting structure for the maser unit.

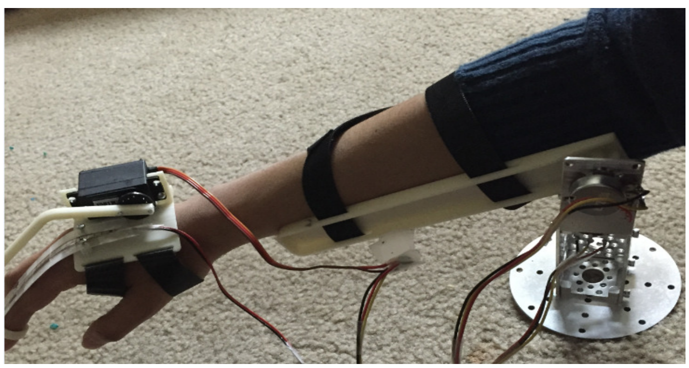


Figure 3.23: Complete master unit with user’s hand in place

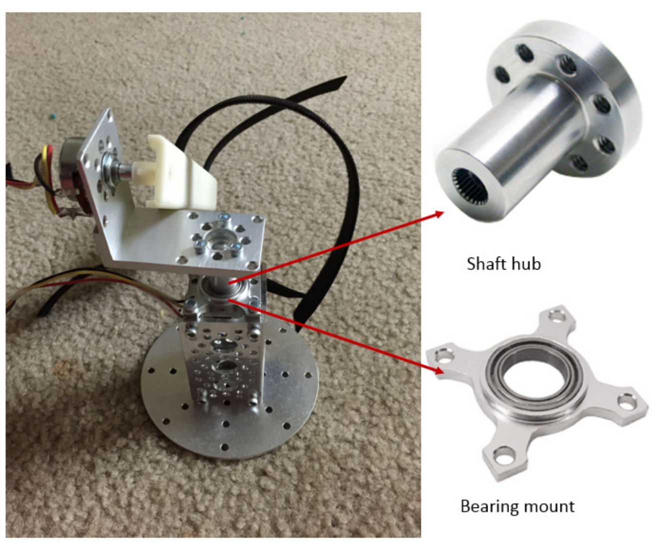


Figure 3.24: Master unit parts

The partial hand exoskeleton unit in figure 3.25 is secured to the hand using Velcro. This part is 3D printed with openings for placing the two fingers. A thin plastic sheet runs over the two fingers and is mounted on both ends. A flex sensor is mounted on this sheet. The unit also consists of a link for providing haptic feedback which is attached to the shaft of the motor on one end and to the finger cap on the other end. The home position of the hand unit is defined as a position where the angle between the four fingers and thumb is 90 degrees. At this position, the resistance offered by the flex sensor would be zero. The maximum angle to which the unit can move from the home position is 44.5 degrees as shown in the figure 3.26.

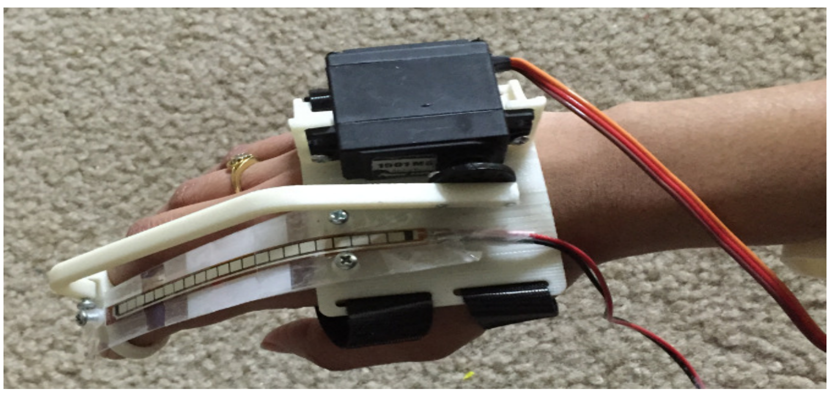


Figure 3.25: Partial master unit hand exoskeleton

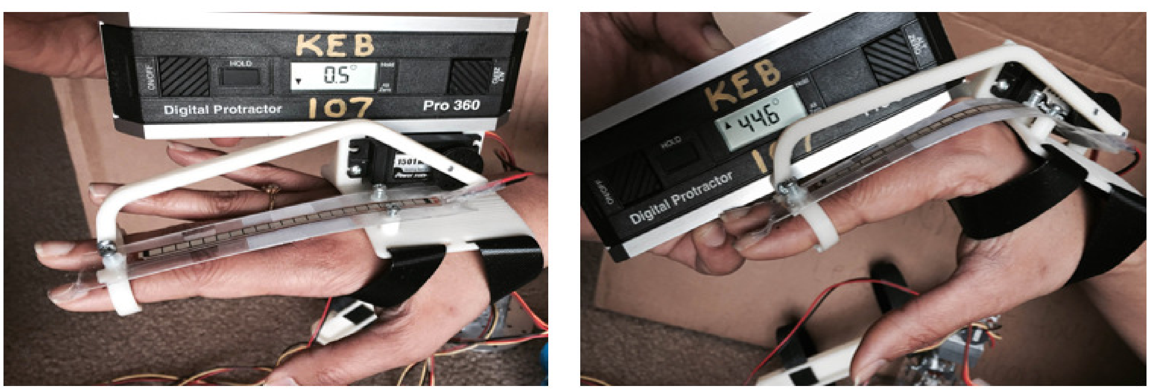


Figure 3.26: Hand exoskeleton in home position (left) and maximum stretched position (right)

3.3.4 Physical system of slave unit

The slave robotic unit consists of two degree of freedom robotic arm structure based on two rotating joints shown in figure 3.27. A servo motor for joint 1 is attached to a 3.5" long aluminum channel and gear of the servo motor is mounted to an L bracket through a mounting hub to obtain swivel motion in the x-plane. Similarly, servo motor gear at joint 2, enclosed in a 9" aluminum channel, is mounted to the top end of the L bracket to obtain a swivel motion in y-plane. The end effector is mounted at the end of the 9" channel. Therefore, the overall reach of the robotic arm in the x- axis is 19.67" and y-axis is 11.67". The working area of the robotic arm based on these lengths is 140.39 in^2 and is plotted using the kinematic model detailed in Appendix B. The home position for this unit is defined as 90 degree between link 1 and link 2. Detailed dimensions of this unit can be found in Appendix C.

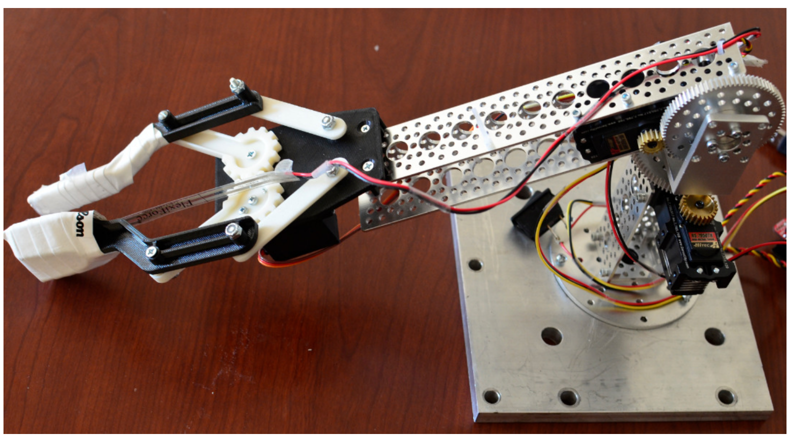


Figure 3.27: Proposed system of slave unit

4 Experimental Results and Discussion

The various components of the proposed master-slave system were tested separately at various stages of the development process. The integrated system was also tested after all the components were obtained and assembled. The results of all testing performed during the project and the relevant discussion are provided in the following sections.

4.1 Verification of forward kinematic model

The forward kinematic model is verified against the actual system by providing the same input of joint space values to both the model and the slave unit of the actual system. Results of the cartesian space values (end-effector coordinates) from both sides were gathered and compared, as shown in table 4.1. A testing was done to verify the accuracy of the coordinates against the values obtained through theoretical model. Different joint angles were input to joint 1 and joint 2 servo motors through Arduino IDE. Once the robot moved to this position, the x, y and z coordinates were recorded using a tape measure.

Table 4.1: Comparison of Cartesian space values of the model and actual system

Trials	Joint space values in degrees		Cartersian space values (End-effector)-model in inches			Cartersian space values (End-effector)-Actual in inches			Error in inches		
	ϴ1	ϴ2	X	Y	Z	X	Y	Z	X	Y	Z
1	0	0	10	0	5.75	9.5	0	5	0.5	0	0.75
2	0	45	7.071	0	12.821	7	0	12.5	0.071	0	0.321
3	45	0	7.07	7.07	5.75	6.75	7	5.5	0.32	0.07	0.25
4	45	45	5	5	12.82	5	4.75	12.5	0	0.25	0.32
5	0	90	0	0	15.75	0	0	15.75	0	0	0
6	-45	10	6.96	-6.96	7.84	6.75	-6.75	7.5	0.21	-0.21	0.34
7	60	30	4.33	7.5	10.75	4	7.25	10.5	0.33	0.25	0.25
8	70	40	2.62	7.19	12.17	2.5	7	12	0.12	0.19	0.17
9	-20	20	8.8	-3.21	9.17	8.5	-3	8.75	0.3	-0.21	0.42
10	-80	30	1.5	-8.5	10.75	1.25	-8.25	10	0.25	-0.25	0.75

It can be observed from the table 4.1 that the difference in the Cartesian space values of the model and the actual system on an average is 0.21, 0.143, and 0.357 inches in x, y and z direction respectively. These differences can be considered minimal over the given range. However, some of these difference could be due to human error involved in measuring the distances using the tape measure and also due to the varied voltage supply of the battery that might affect the performance of the servo motor which in turn hinders the ability of the link to move to the required location.

4.2 Verification of inverse kinematic model

The inverse kinematic model is verified by inputting the x, y and z coordinates to the slave unit and measuring the corresponding joint angles using digital protractor. These joint angle values are then compared to the theoretical model values. This test enabled to validate the forward kinematic model. It can be seen from the table 4.2 that difference in model joint space values and the actual joint space values is within 1 degree which is very minimal and therefore the inverse kinematics model is verified.

Table 4.2: Comparison of joint space values of the model and the actual system

Trials	Cartesian space values in inches			Joint space value (Model) in degrees		Joint space values (Actual) in inches		Error in degrees	
	X	Y	Z	ϴ1	ϴ2	ϴ1	ϴ2	ϴ1	ϴ2
1	10	0	5.75	0	0	0.9	0	0.9	0
2	7.071	0	12.821	0	45	0	44	0	-1
3	7.07	7.07	5.75	45	0.9957	43.9	0.5	-1.1	-0.4957
4	5	5	12.82	45	45	44.9	46	-0.1	1
5	0	0	15.75	0	90	0.05	90.8	0.05	0.8
6	6.96	-6.96	7.84	-45	10.169	-45.7	9.6	-0.7	-0.5686
7	4.33	7.5	10.75	60.0007	30.001	60.2	30.6	0.1993	0.5993
8	2.62	7.19	12.17	69.9785	40.071	69	41	-0.9785	0.9291
9	8.8	-3.21	9.17	-20.0406	20.493	-21	20	-0.9594	-0.4925
10	1.5	-8.5	10.75	-79.992	30.33	-79	29.6	0.992	-0.7297

4.3 Verification of the integrated system

The complete system was tested for grasping performance and feedback capability. In order to evaluate the performance of the slave unit in approaching and grasping, objects from 25 categories that were ranked most important for robotic retrieval by motor impaired patients from the Emory ALS center [36] were considered. Out of these objects, a few of them were chosen for testing the grasping capabilities of the end effector.

A grasp was deemed successful if the robot picked up the object and held it there for about 10 secs without dropping. The robotic arm was moved from -45 degrees to 135 degrees and the corresponding distance was about 24 inches in y-axis where the object was kept. The qualitative analysis of this experiment is shown in table 4.3. The robotic arm was able to lift 4 out of 5 objects successfully. Also, all the objects were of different shapes and size.

Table 4.3: Overall performance of the system

Sl. No	Object	Dimensions, inch	Grasp (√ or ×)	Haptic feedback	Time in seconds
1	cookie packet	5 x 2	√	×	25
2	Plastic bottle	6 x 3	√	√	15
3	Perfume bottle	4 x 3	√	√	20
4	Glass	6 x 3.5	√	√	13
5	Pill box	4 x 1	√	×	15

The time taken to pick and place each of the objects was also recorded as shown in the table 4.3. The time taken by the robotic arm to pick and place most of the chosen objects were less than 30 seconds which goes in line with the defined objectives in section 2. The maximum time was taken for picking up a cookie packet as the end effector joint is rigid and has no degree of freedom. The system was unable to provide haptic feedback for cookie packet and pill box.

The robotic arm had to reach very close to the ground to pick these objects and the object did not come in complete contact with the flexi force sensor.

4.4 Repeatability of flex sensor

Repeatability of the flex sensor was quantified by the following experiment. The haptic motor on the master unit was programmed to bend the flex sensor to a specific angle.

Table 4.4: Repeatability of flex sensor

Angle of flex sensor in degrees	Arduino analog reading of flex sensor				Average	Standard deviation
	Trial 1	Trial 2	Trial 3	Trial 4		
1	213	212	212	214	212.75	0.96
2	212	213	212	214	212.75	0.96
3	213	214	213	213	213.25	0.5
4	209	213	211	215	212	2.58
5	210	212	212	215	212.25	2.06
6	206	208	211	212	209.25	2.75
7	206	204	213	210	208.25	4.03
8	209	203	211	209	208	3.46
9	208	201	210	209	207	4.08
10	206	200	209	206	205.25	3.77
11	206	197	207	205	203.75	4.57
12	205	197	207	204	203.25	4.35
13	203	194	205	201	200.75	4.79
14	200	194	203	202	199.75	4.03
15	200	195	203	201	199.75	3.4
16	197	193	201	194	196.25	3.59
17	196	191	200	196	195.75	3.69
18	193	189	198	196	194	3.92
19	194	190	197	196	194.25	3.1
20	191	188	193	192	191	2.16

The setup was automated to flex the sensor in steps of one degree within the allowed range of 45 degrees. At each step, the corresponding analog values were recorded from the Arduino serial monitor. A smaller standard deviation within the tabulated values at a particular angle would indicate that the sensor has good repeatability. The recorded analog values from the flex sensor and standard deviation are tabulated in table 4.4. It can be noted that the standard deviation remains consistent over the range of 20 degrees and varies between 0 and 5 which is an acceptable range for the application. The calibration procedure and technical specifications of this sensor is detailed in the user manual in appendix F.

4.5 Wireless communication response time between master and the slave unit

Due to wireless communication, the response time between the master and slave units might vary and affect the performance of the complete system. In order to determine the response time, the time difference between the master unit's start of motion at random initial positions and the corresponding actuation of the slave unit was calculated. This was measured for both horizontal and vertical axis actuators. A stop watch, with least count of 1 millisecond, was used to record the time for this experiment. The results are detailed in table 4.5 and it can be inferred that there is a delay in response time between the master and slave unit. Much of the delay is due to the communication between the receiver and transmitter radio manager software which is a part of the Arduino library. However, the maximum delay over five trials is 1.578 secs which can be considered very minimal as it is designed to be used by elderly who would require the system to respond slowly and steadily.

Table 4.5: Wireless communication delay between master unit and slave unit's horizontal actuator

Trials	Master unit's start time (stop watch measured value)	Slave unit's start time (stop watch measured value)	Delay in response time, secs
1	00:00:00	00:00:01.237	1.237
2	00:00:00	00:00:00.504	0.504
3	00:00:00	00:00:00.756	0.756
4	00:00:00	00:00:01.578	1.578
5	00:00:00	00:00:01.000	1

4.6 Calibration testing of Flexiforce pressure sensor

When the force sensor is unloaded, its resistance is very high. When a force is applied to the sensor, this resistance decreases. The resistance is measured by a multimeter, when applying a force to the sensing area. Each of the FlexiForce FSRs was conditioned prior to being installed in the prototype. Tekscan recommends performing this conditioning procedure on new FSRs in order to reduce the effects of drift and hysteresis on the sensors [25]. After the conditioning procedure was completed the sensors were calibrated in order to correlate the analog sensor value to the applied force.

The flexiforce sensor was calibrated according to the procedure described in Tekscan manual [25]. Since the maximum weight lifted by the end effector was defined as 2 lb, the calibration was done using the standard weights ranging from 0.022 lb to 0.44 lb (figure 4.1). Each of these weights were placed on the sensor and the respective changes in resistance were recorded using a digital multimeter. The graph of the same is shown in figure and it can be seen that the trend in which the curve is changing matches the actual calibration curve given by the manufacturer as described in the section 4.

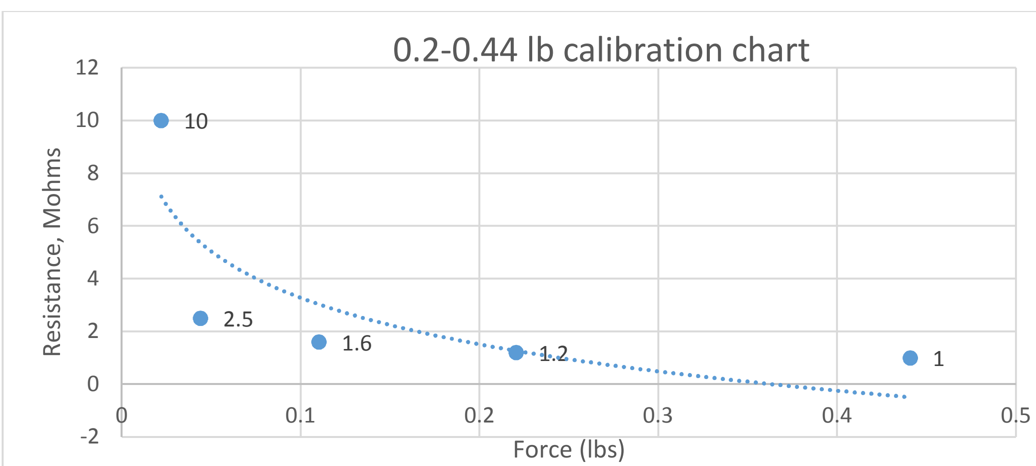


Figure 4.1: 0.022 lb-0.44 lb Calibration curve for flexi force sensor

The sensors were calibrated in order to correlate the analog sensor value to the applied force. The product manual [25] for the 1120 FlexiForce adapter board contains conversion factors for this purpose. Equation 4.1 was taken from the 1120 manual as the conversion factor for the 0-1 lb FlexiForce FSR.

$$Force = sensor\ value/461 \qquad (4.1)$$

This calibration was done by connecting one lead of the flexi force sensor to analog input through a 150 Kohm resistor and the other end to the ground of Arduino microcontroller. The analog input from the sensor was recorded by placing different weights on the sensor. A 0.25" diameter puck was placed on the sensing area of the sensor to concentrate the load applied and obtain accurate results. This data was plotted in excel using a built in function. An equation for converting the raw analog signal to applied force was derived from the best fit equation shown in figure 4.2. This equation is given as equation 4.2.

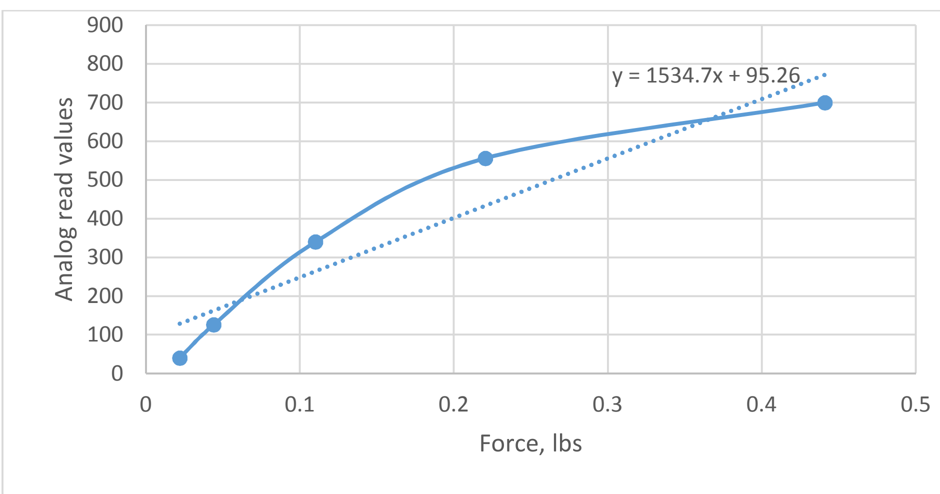


Figure 4.2: Plot of the FlexiForce calibration results showing best fit line and R2 value

$$Force = sensor\ value/1534.7 \qquad (4.2)$$

This equation differs slightly from the equation provided by Phidgets for the 1120 adapter. Because of this discrepancy equation 4.2 was used for the conversion in the Arduino program for providing feedback.

5 Conclusion

Teleoperation and bilateral system with haptic feedback was incorporated in the proposed system that would enable the user to move his/her hand in a natural way for manipulating the robot and in turn obtain force feedback for realistic grasping experience.

The proposed assistive device that has been designed and built, demonstrated to meet all of the quantitative and qualitative objectives set forth in section 2. The device is able to pick and place certain selected objects with minimal effort and within 30 seconds. Once the master unit is secured to the user and the power supply is turned on, the user is not required to look at the screen or use joystick to maneuver the robotic arm but instead can move their hand intuitively and reach for objects. Haptic feedback was obtained for most of the selected objects except cookie packet and pill box which was due to insufficient contact between pressure sensor surface and the object. The sensors used for this device are all tested for accuracy and have an error percentage of less than 5%.

The wireless data communication on an average showed a delay in response time of 1.015 secs and a standard deviation of 0.416. However, this delay is very minimal as the proposed device is designed to be used by elderly people which demands a slower motion of the system.

Bill of material was created for building this prototype (Appendix K) which indicates that the overall cost for the build was around $1205.60. Some of the commercially available wheelchair mounted assistive device costs around $12500-50,000 dollars [21] which clearly indicates that the prototype is 90% cheaper. Moreover, the prototype uses off the shelf components which are much more cost effective as opposed to the custom made parts. However, the overall cost for the build is excluding the mobile base for the slave robotic unit which was one of the hypothesis. A rough bill of material for building the mobile base is detailed in Appendix K which has an overall cost of 2276 $. The total cost of the device considering the mobile base would therefore be 3481$ which is still 72% cheaper than the commercially available assistive devices. The estimated cost for the prototype is based on a payload capacity of 2 lbs as opposed to the 3.3 lbs on the commercially available MANUS arm. Based on this, the cost of the prototype may rise a little due to the requirement of higher capacity motors and power supplies for meeting the higher load capacity. On the other hand, the estimated cost of the prototype is including the haptics technology which is an added feature as opposed to the existing assistive devices. In conclusion, the proposed device is considerably cost effective which is built using inexpensive off the shelf components along with haptics feedback technology.

The safety of the slave robotic arm was ensured by carrying out the forward kinematic model and defining safe working limits for both user and the robotic arm in the Arduino program. The force limits for haptic feedback were set by carrying out numerous experiments and therefore the device has been prototyped by considering various safety aspects for the chosen environment. Emergency stops can be added to each of the motors in the system to provide further safety against any possible hardware glitches.

Research suggests that haptic feedback offer therapy benefits, and therefore clinical trials would be an expansion to this project. Nevertheless, various factors need to be taken into consideration for carrying out such tests. Some of these aspects are discussed in the following sections.

On the whole, the proposed haptic feedback assistive device is a cost effective and reliable assistive device that enables the user to reach and grasp objects. It may be able to provide affordable assistance in the near term, especially with tasks that involve small, lightweight objects.

5.1. Ethical considerations

The proposed haptic assistive device was designed and built as a proof of concept prototype. As such if used for patient trials even though safety aspects have been taken into consideration, it would require a more rigorous testing and design to ensure adherence to safety and national standards. A much detailed analysis is needed to testify the credibility of the complete system. As professionals engineers are bound by professional codes of ethics [37], in full clinical trials, or any trial involving actual patients, consideration must be given to the safety of the patient using the device. In a strict sense an engineer beginning human trials on a device such as this should insist on testing being monitored by a trained personnel. This will ensure minimum risk to any human subjects

5.2. Future work

The haptic feedback project was treated as a prototype from the initial concept phase. With that basis there are several features that are not included in this initial prototype that may be included in future iterations of the design.

5.2.1. Range of motion of master unit

The master unit is currently designed to be mounted to a wheelchair so that the user can operate the slave robotic arm easily. However, the user's hand will be constrained to the unit and since only elbow joint is considered for two rotating motions, the range of motion of the user is restricted to a maximum of -30 degrees to 60 degrees. This might also strain the users hand as the whole hand movement is concentrated on the elbow joint. Therefore, future design revisions should include shoulder joint and also incorporate sensors such as accelerometer for mapping the hand data to the remote robotic arm. The partial hand exoskeleton also has a range of motion restriction of 45 degrees. This limits the ability of the user to hold small objects. The design can further be iterated to incorporate flexibility to handle a wide range of objects.

5.2.2 Haptic feedback capability

The proposed system consists of only one pressure sensor and the experimental trials indicated that the hand did not obtain haptic feedback when holding certain type of objects. The reason behind this issue is the point of contact between the object and the end effector. Therefore, the future design needs to incorporate more sensors to exactly position the end effector around the object for obtaining accurate and precise haptic feedback.

5.2.3 Single source power supply

The current haptic feedback assistive prototype incorporates four servo motors run on separate power supplies. The motor for joint 1 and 2 on the slave unit requires 6V and 4.8A stall current, whereas the end effector and haptic motor requires 6V, 3A current. For testing purposes, Ni-mh batteries were used to supply varying current rating. This means that there are four batteries that needs to be charged to operate the prototype. Therefore, future revisions of this device should incorporate a single dedicated power supply with different current rating that matches the motor specifications.

5.2.4. Location of the remote object by slave robotic unit

Aligning the end effector to the objects can be quite challenging with the existing prototype as the user has to really be really attentive for doing this task which in turn can be very strenuous. A possible future solution for this could be to attach a video camera to the slave robotic unit and a display unit to the wheel chair to enable the user to clearly see the objects remotely. The robotic arm can be programmed based on the visual feedback to precisely locate the object of interest. There are numerous researches going on in this area. The robot Obrero uses vision at the beginning of the task to direct the attention of the robot and to get a rough estimation of the position of the object. Next, the robot moves its limb towards the object and explores with the hand the area around it. During exploration, the robot exploits tactile feedback to find the actual position of the object and grasp it [41].

5.2.5 Revised end effector design

The end effector design of the slave robotic unit has only one degree of freedom and hinders the ability of the user to hold different shaped objects. A five fingered robot as described in the section 2 could be one of the future revisions. However, maneuvering this could be more dexterous and fabrication of such an arm can be expensive.

Appendix A

Torque, speed and end effector gripping force calculations

A simplified beam model has been created to carry out the torque calculation as shown in the figure A1.

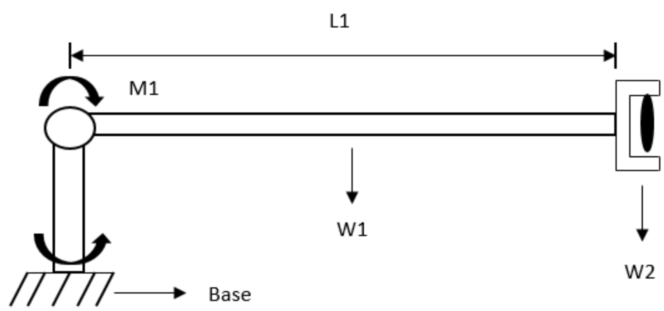


Figure A1: Simplified beam model of the slave robotic arm

Parameters

Material chosen: Aluminum alloy

Aluminum is a light weight material with a density of $0.095\ lb/in^3$ and high tensile strength of 483 MPa. This enables to reduce the overall weight of the robotic arm along with providing excellent strength.

L1= Length of link 1=10 in

Robotic arm lengths are chosen by considering the average human arm lengths for reaching things in shelves.

W1=Weight of link 2=0.77 lb

W2=Weight of end effector = 0.25 lb

W0=Weight of the object = 1.5 lb

Considering a factor of safety of 20% for the weight of the object, maximum weight of the object would be

1.5 lb+(1.5 lb*0.2)=1.8 lb

Therefore total weight of end effector and object = 1.8 lb+ 0.25 lb=2.05 lb

Torque at joint 1

$$M1 = 2.05\ lb * 10\ in + (0.77\ lb * 5\ in) = 48.7\ lb - in = 5.5\ N - m = 56\ kg - cm$$

779.2 oz-in approximately equal to 800 oz-in

Servo city, manufacturer of a variety of servo motor recommends to double the calculated torque for safety purposes. Based on this, a HS 7950 (3:1 gear ratio) with a speed of 0.39 seconds/60 degrees and torque of 1206 Oz-in at 6V was chosen for both joint 1 and 2.

The angular velocity of this servo motor was calculated as follows:

Given torque=56 kg-cm=4.05 pound foot and speed=0.39 sec/60 deg

For 0.39 sec, rotation is 60 deg, for 1 sec, rotation=60/0.85=153 degrees

Based on the torque and speed calculation, it is found that this servo motor meets all the set objectives and specifications.

The calculation of the minimal gripping force that the robot gripper must apply will include the mass of the part that must be moved, the friction coefficient between the finger material and the part material and the gravitational acceleration constant [31].

F: Gripping force [N]
u: Coefficient of static friction=1.1 (Rubber material)
m: Mass of the part [kg] =0.816 kg
g: Gravitational acceleration [9.81 m/s^2]
a: Acceleration (if it is significant)

To make sure the part doesn't slip during static pretension, the gripping force should be higher than the weight of the part itself.

$$F > m(g+a)/u \qquad (A1)$$

$$F = 0.816 \times \frac{9.8}{1.1} = 7.26\ N$$

Factor of safety=F+20% *F=7.26 N+ (0.2*7.26 N)=8.712 N=0.89 Kgf

Factor of safety = (allowable load-Actual load)/allowable load (based on torque limits of motor)

Gripper torque=gripper Force*Length of the jaw (It is the distance from the gripper face)

Gripper torque=0.89 Kg*14 cm=12.46 Kg-cm

Appendix B

Forward and inverse kinematic modeling

As a part of the concept development phase of the project, a kinematic model of the arm was developed. The model of the arm shown in figure B1 was used for the kinematic analysis. This will allow for simulation of the device to verify that the designed parameters function as desired. Frames have been affixed to all parts of mechanism and the relationship between these frames has been described. Frame 0 defines the 0 location for all angles and distances in the kinematic model. ϴ1 is the angle between frame 0 and frame 1. Frame 1 is attached to the elbow joint and rotates with the joint. L1 is the distance between the x-axes of frame 0 and frame 1. ϴ 2 is the angle between the z-axis of frame 1 and the z-axis of frame 2. L2 is the distance between the z-axes of frame 2 and frame 3. Frame 3 is attached to the end of the link 2 representing the center of the end effectors grip.

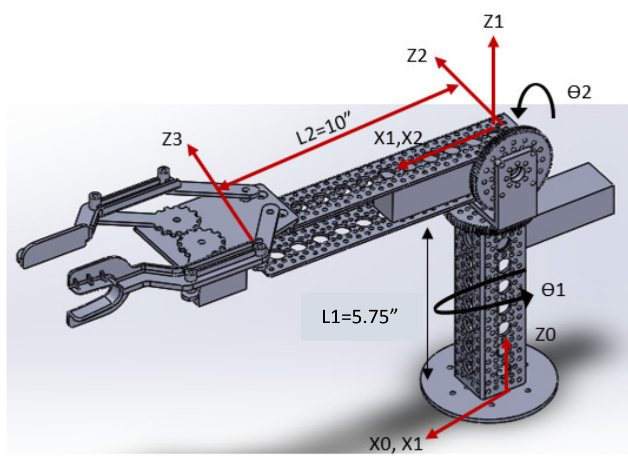


Figure B1: Kinematic model of the proposed device

Link is considered as a rigid body that defines the relationship between two neighbouring joint axes of the robotic arm. The mechanisms are defined using the link parameters.

Table B1: Kinematic Parameters Defined from Attaching Frames to the Model

i	$\boldsymbol{\propto_{i-1}}$	$\boldsymbol{a_{i-1}}$	$\boldsymbol{d_i}$	$\boldsymbol{\theta_i}$
0	0	0	0	0
1	0	0	L1	θ_1
2	90°	0	0	θ_2
3	0	L2	0	0

In general, the transformation matrix that defines the frame {i} relative to the frame {i-1} is

$$ {}^{i-1}_{i}T = \begin{bmatrix} C\theta i & -S\theta i & 0 & a_{i-1} \\ S\theta i . C\alpha_{i-1} & C\theta i . C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} . di \\ S\theta i . S\alpha_{i-1} & C\theta i . S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} . di \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(B1)} $$

Where, $\theta, \alpha, a \ and \ d$ are link parameters which are shown in the figure B1

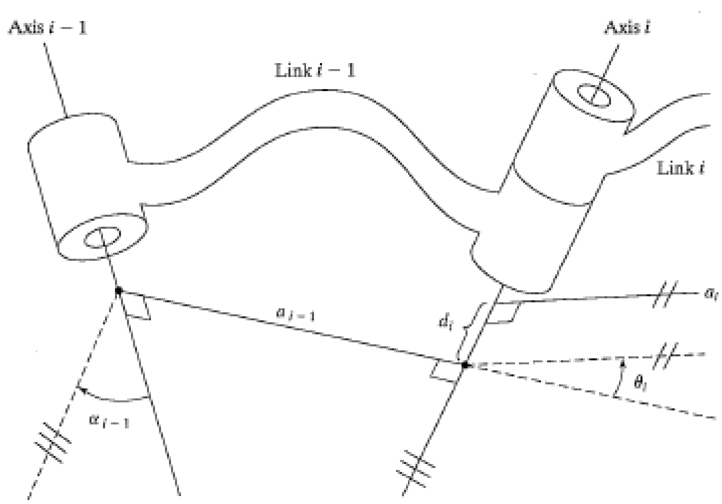


Figure B2: Definition of link parameters

By substituting the individual link parameters into the equation B1, the individual transformartion for each link of the proposed robotic arm (Figure B1) is calculated as follows:

$$ {}^{1}_{0}T = \begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(B2)} $$

$$ {}^{1}_{2}T = \begin{bmatrix} C2 & -S2 & 0 & 0 \\ 0 & 0 & -1 & -L1 \\ S2 & C2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(B3)} $$

$$ {}^{2}_{3}T = \begin{bmatrix} C3 & -S3 & 0 & L2 \\ S3 & C3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(B4)} $$

The final transformation matrix = ${}^{1}_{0}T \times {}^{1}_{2}T \times {}^{2}_{3}T$ (B5)

$$ {}^{0}_{3}T = \begin{bmatrix} Cos(\theta 1) * Cos(\theta 2) & -Cos(\theta 1) * Sin(\theta 2) & Sin(\theta 1) & L2 * Cos(\theta 1) * Cos(\theta 2) \\ Cos(\theta 2) * Sin(\theta 1) & -Sin(\theta 1) * Sin(\theta 2) & -Cos(\theta 1) & L2 * Cos(\theta 2) * Sin(\theta 1) \\ Sin(\theta 2) & Cos(\theta 2) & 0 & L1 + L2 * Sin(\theta 2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(B6)} $$

MATLAB code for slave robotic arm work envelope

```
%Determining the transformation matrix
%Written by: Archana Pradeep
%Instructor: Dr. Nael Barakat

clc;
clear all;

%Define the link lengths

L1=3.75;
L2=10;

%Define the angle limits for link 1 and 2

th1min=-45*3.14/180;
th1max=135*3.14/180;
```

```
th2min=0;
th2max=70*3.14/180;

hold off

% Determine the transformation matrix for the mechanism

for th1=th1min:10*3.14/180:th1max
for th2=th2min:10*3.14/180:th2max

T01 = [cos(th1) -sin(th1) 0 0;sin(th1) cos(th1) 0 0;0 0 1 L1;0 0 0 1];

T12 = [cos(th2) -sin(th2) 0 0;0 0 -1 0; sin(th2) cos(th2) 0 0;0 0 0 1];

T23 = [1 0 0 L2;0 1 0 0;0 0 1 0;0 0 0 1];

T03=T01*T12*T23
disp(T03)

%defining the position variables
X=T03(1,4);
Y=T03(2,4);
Z=T03(3,4);

% Plot the workspace

scatter3(X,Y,Z)
title('Robot workspace', 'fontsize',16)
xlabel('Distance,mm')
ylabel('Distance,mm')

hold on
end
end
```

The inverse kinematics models allows the calcuation of the vector of joint values corressponding to a designed end effector goal state.

The inverse kinematics model is given by the following equations.

$$X = L2 * Cos(\theta 1) * Cos(\theta 2) \quad \text{(B7)}$$

$$Y = L2 * Cos(\theta 2) * Sin(\theta 1) \quad \text{(B8)}$$

$$Z = L1 + L2 * Sin(\theta 2) \quad \text{(B9)}$$

The first step to finding the value of the angles is to square and add these two equations. The resulting eqaution is shown below.

$$x^2 + y^2 = L2^2 * cos^2\theta 2 \quad \text{(B10)}$$

Solving this equation for the cosine of $\theta 2$ yields:

$$\cos(\theta 2) = \frac{\sqrt{x^2+y^2}}{L2} \qquad \text{(B11)}$$

By trignometric identity:

$$\text{Sin}(\theta 2) = \sqrt{1 - cos^2\theta 2} \qquad \text{(B12)}$$

Therefore the final value of $\theta 2$ can be found using the Atan2 function.

$$\theta 2 = Atan2(\sin(\theta 2), \cos(\theta 2)) \qquad \text{(B13)}$$

It is important to note that in software, equations B7, B8 and B9 must be calculated in order to determine the value of $\theta 2$. At this point one of the variables, $\theta 2$, is known. Therefore the value for $\theta 1$ can be solved using the equations for x and y above. Through algebraic manipulation the equations for x and y can be re-written as follows:

$$\frac{y}{x} = \tan(\theta 1) \qquad \text{(B14)}$$

Therefore, $$\theta 1 = atan2(y, x) \qquad \text{(B15)}$$

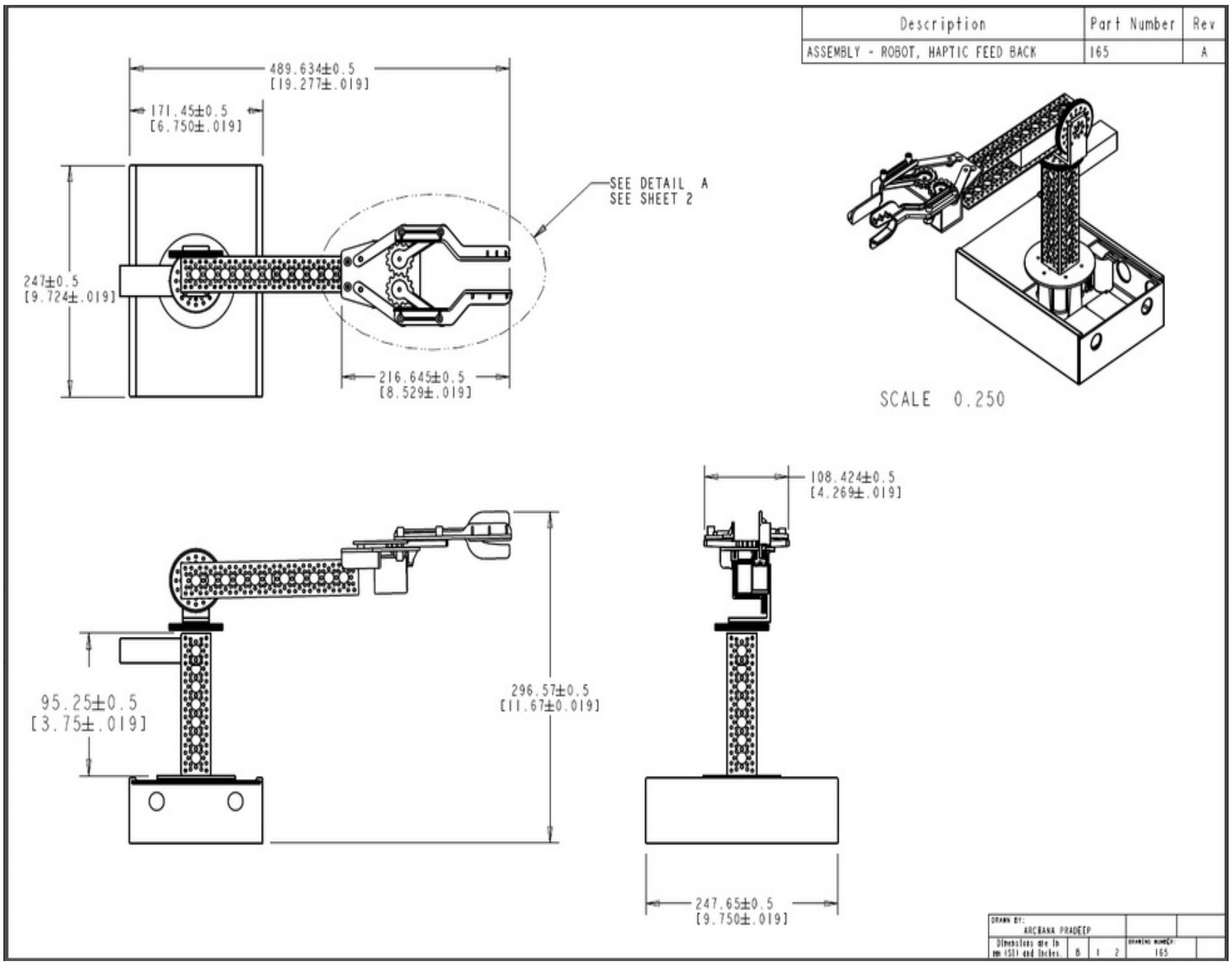
Where all the variables are the same as was used in the table B1

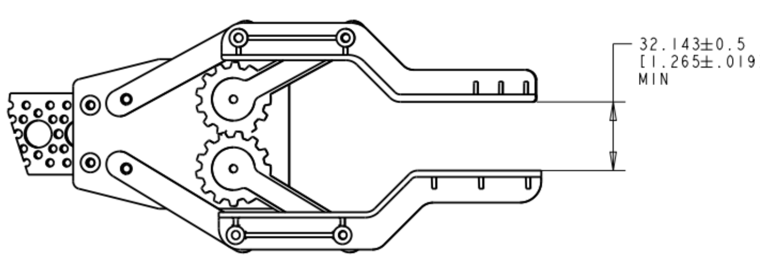
Hard coding these equations into software allows the joint angles to be calculated in real time as x-y coordinates are given to the controller. This inverse kinematic model can be used to programmatically provide the information for how to move each joint of the prototype to achieve a preset x-y position for the user hand relative to the location of their elbow.

Appendix C

Mechanical drawings

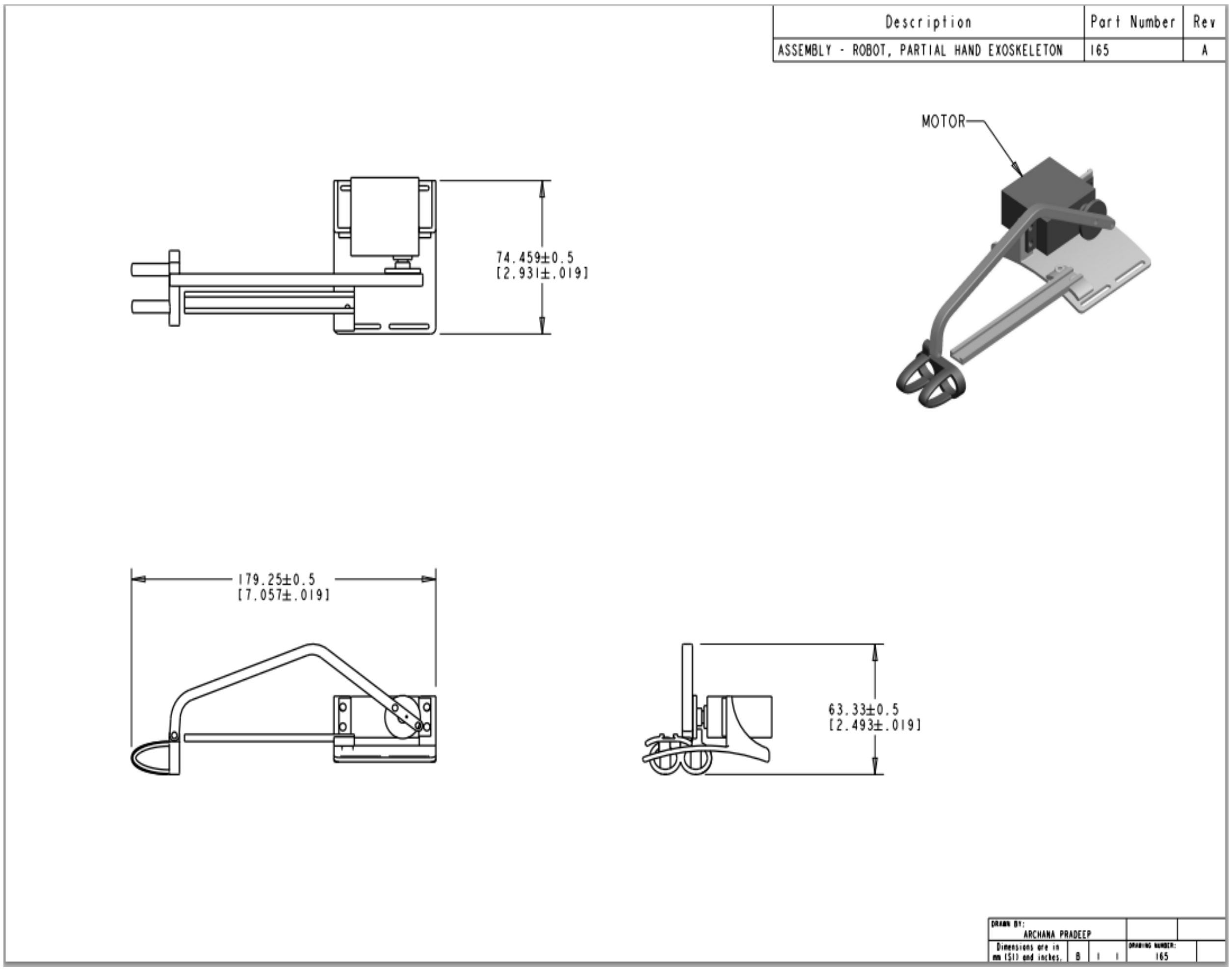
Slave unit



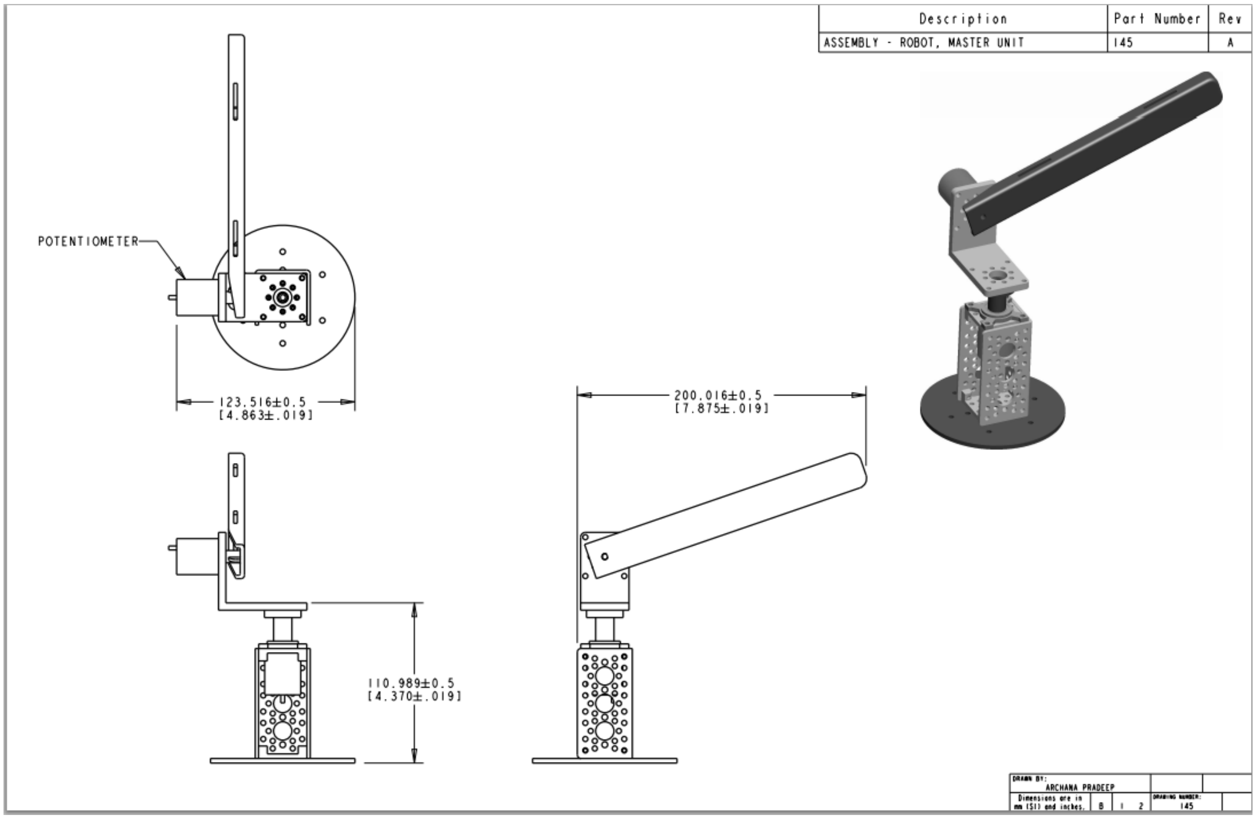


DETAIL A
SCALE 0.550

Partial hand exoskeleton



Master unit


Description
Part Number
Rev
ASSEMBLY - ROBOT, MASTER UNIT
145
A
POTENTIOMETER
123.516±0.5
[4.863±.019]
200.016±0.5
[7.875±.019]
110.989±0.5
[4.370±.019]
DRAWN BY:
ARCHANA PRADEEP
Dimensions are in
mm (SI) and inches.
B
1
2
DRAWING NUMBER:
145

Appendix D

Electrical schematic

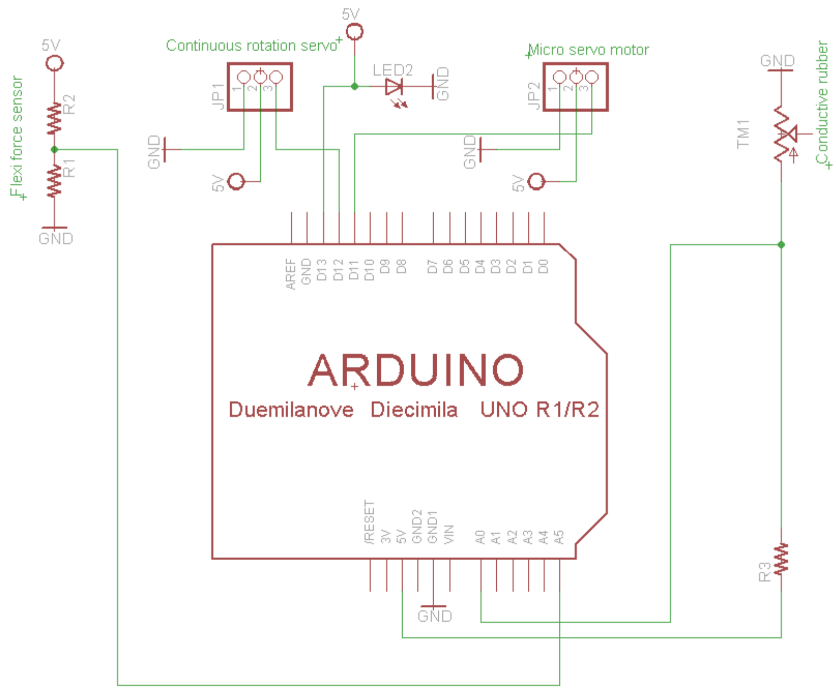
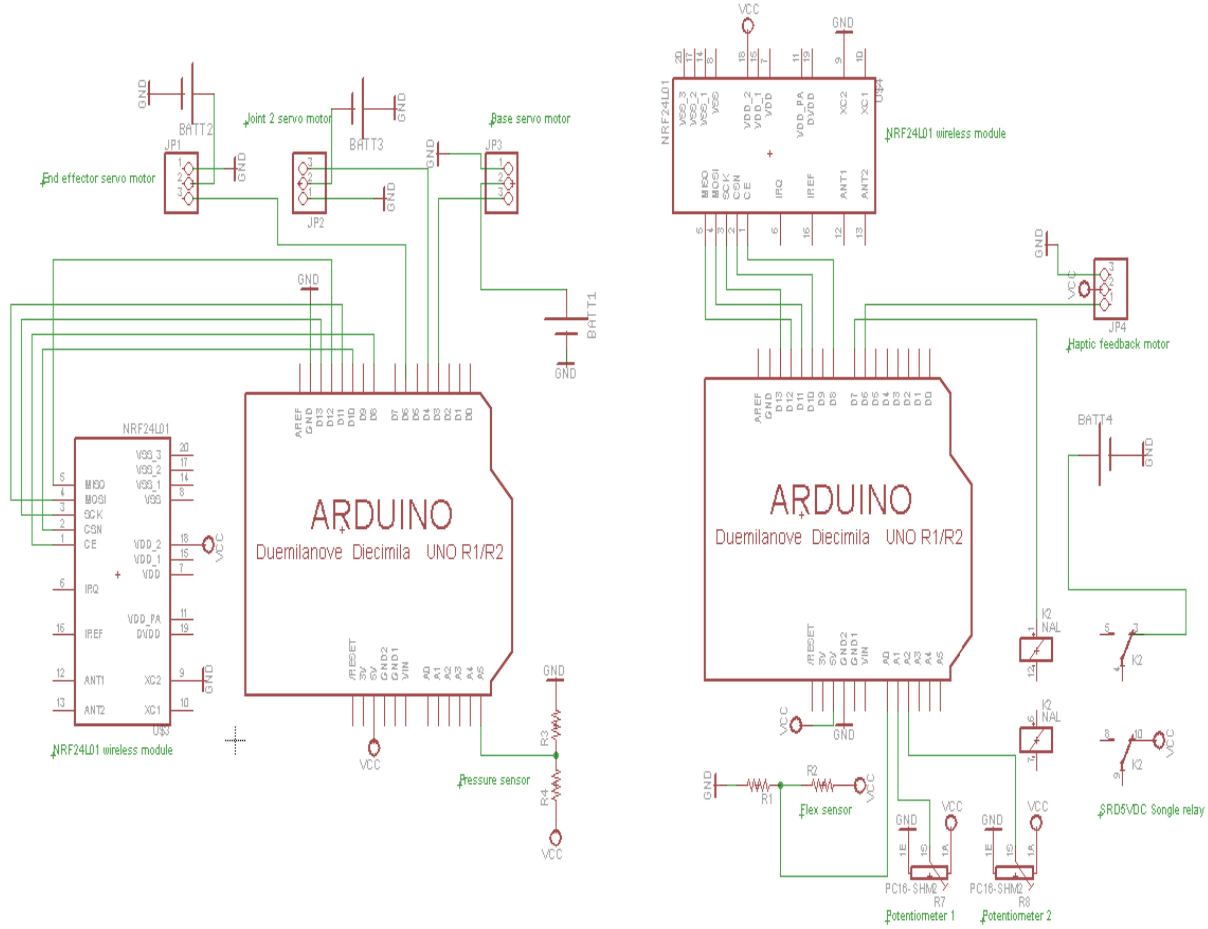


Figure 7: Schematic of the electronic circuit developed for the proof of concept

Working Prototype schematic



Appendix E

Arduino code for the complete system

Transmitter Code

```
/* - CONNECTIONS: nRF24L01 Module:
 1 - GND
 2 - VCC 3.3V !!! NOT 5V
 3 - CE to Arduino pin 8
 4 - CSN to Arduino pin 10
 5 - SCK to Arduino pin 13
 6 - MOSI to Arduino pin 11
 7 - MISO to Arduino pin 12
 8 - UNUSED

/*-----( Import needed libraries )-----*/
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>
#include <SoftwareServo.h>

/*-----( Declare Constants and Pin Numbers )-----*/
#define JoyStick_X_PIN     A1  //Pin Numbers
#define JoyStick_Y_PIN     A2
//#define ClickPIN           4
#define flexsensor_pin     A0
#define hapticmotor_pin    7
#define CLIENT_ADDRESS 1      // For Radio Link
#define SERVER_ADDRESS 2


SoftwareServo servo1;
// Create an instance of the radio driver
RH_NRF24 RadioDriver;

// Create an instance of a manager object to manage message delivery and receipt, using the driver declared above
RHReliableDatagram RadioManager(RadioDriver, CLIENT_ADDRESS);// sets the driver to NRF24 and the client adress to 1


int pressureJoystickReceived;
int hapticmotorposition;

/*-----( Declare Variables )-----*/
uint8_t joystick[3];  // 2 element array of unsigned 8-bit type, holding Joystick readings
uint8_t ReturnMessage[]="pressure sensor data Received";
// Predefine the message buffer here: Don't put this on the stack:
uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];    // Actually: 28 bytes (32 minus 4 byte header)
int angleOffset= -8;
int feedbackLoopCount=0;

void setup()  /****** SETUP: RUNS ONCE ******/
{
  pinMode(2,OUTPUT);
  digitalWrite(2,HIGH);
  pinMode(4,OUTPUT);
  digitalWrite(4,HIGH);
  pinMode(5,OUTPUT);
  digitalWrite(5,LOW);
```

```
  // begin serial to display on Serial Monitor. Set Serial Monitor to 115200
  Serial.begin(115200);
  servo1.attach(7);
  // NOTE: pinMode for Radio pins handled by RadioDriver

  if (!RadioManager.init())   // Defaults after init are 2.402 GHz (channel 2), 2Mbps, 0dBm
    Serial.println("init failed");

  }


void loop() /****** LOOP: RUNS CONSTANTLY ******/
{
  int haptic_feedback;

  //Read the joystick values, scale them to 8-bit type and store them in the joystick[] array.
  // Take the value of Joystick voltages which are 0 to 1023 (10 bit), and convert them to 0 to 255 (8 bit)
  joystick[0] = map(analogRead(JoyStick_X_PIN), 155, 530, 109, 70 );
  joystick[2] = map(analogRead(JoyStick_Y_PIN), 260, 662, 90, 40);
  joystick[1]= map(analogRead(flexsensor_pin), 130, 195, 100, 180);
  //haptic_feedback=map(analogRead(flexsensor_pin), 130, 211, 45, 0);
  Serial.print("analog");
  Serial.print(analogRead(JoyStick_Y_PIN));

  //Display the joystick values in the serial monitor.
  Serial.print("Sending data --");
  Serial.print("x:");
  Serial.print(joystick[0]);
  Serial.print("; y:");
  Serial.print(joystick[2]);
  Serial.print("; motor 2:");
  Serial.println(joystick[1]);



 if (RadioManager.sendtoWait(joystick, sizeof(joystick), SERVER_ADDRESS))
 {
   // Now wait for a reply from the server
   uint8_t len = sizeof(buf);
   uint8_t from;
   if (RadioManager.recvfromAckTimeout(buf, &len, 2000, &from))
   {
     Serial.print("got reply from :0x");
     Serial.print(from, HEX);
     Serial.print("; PressureSensorValue:");
     Serial.println(buf[0]);

     if (buf[0] > 5)
     {
       digitalWrite(5, HIGH);
       Serial.print("; analog read flexsensor:");
       Serial.print(analogRead(flexsensor_pin));
```

```
    int feedbackIntensity = GetFeedbackIntensity(buf[0]);
    switch(feedbackIntensity)
    {
    case 0:
      feedbackLoopCount = 0;
      break;
    case 1:
      feedbackLoopCount = 9;
      break;
    case 2:
      feedbackLoopCount = 10;
      break;
    case 3:
      feedbackLoopCount = 11;
      break;
    case 4:
      feedbackLoopCount = 12;
      break;
    }

    Serial.print("; feedbackLoopCount:");
    Serial.println(feedbackLoopCount);


    haptic_feedback=map(analogRead(flexsensor_pin), 130, 211, 45, 0);
    int angle = haptic_feedback - angleOffset;

    if (angle < 0)
    {
      angle = 0;
    }
    if (angle > 45)
    {
      angle = 45;
    }

     Serial.print("; angle:");
     Serial.println(angle);
     servo1.write(angle);

     for (int i=0; i <= feedbackLoopCount; i++)
     {
       SoftwareServo::refresh();
       delay(20);
     }
     digitalWrite(5, LOW);
     delay(200);
   }
   else
   {
     digitalWrite(5, LOW);
   }
}
else
{
  Serial.println("No reply, is nrf24_reliable_datagram_server running?");
}
```

```
  }
  else
    Serial.println("sendtoWait failed");

 //delay(50); // Wait a bit before next transmission
 // SoftwareServo::refresh();
}


//Function to get feedback intensity level based on the pressure sensor input
int GetFeedbackIntensity(int pressurevalue)
{
  int feedbackMotorIntensity;

  if ((pressurevalue >= 6) && (pressurevalue <= 15))
  {
    feedbackMotorIntensity = 1;
  }
  else if ((pressurevalue >= 16) && (pressurevalue <= 30))
  {
    feedbackMotorIntensity = 2;
  }
  else if ((pressurevalue >= 31) && (pressurevalue <= 40))
  {
    feedbackMotorIntensity = 3;
  }

  else
  {
    if (pressurevalue >= 41)
    {
      feedbackMotorIntensity = 4;
    }

    if (pressurevalue <= 5)
    {
      feedbackMotorIntensity = 0;
    }
  }

  return feedbackMotorIntensity;
}
```

Receiver Code

```
/* nrf24l01 module connections
 1 - GND
 2 - VCC 3.3V !!! NOT 5V
 3 - CE to Arduino pin 8
 4 - CSN to Arduino pin 10
 5 - SCK to Arduino pin 13
 6 - MOSI to Arduino pin 11
 7 - MISO to Arduino pin 12
 8 - UNUSED

/*-----( Import needed libraries )-----*/

#include <SoftwareServo.h>  // Regular Servo library creates timer conflict!
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>

/*-----( Declare Constants and Pin Numbers )-----*/
#define CLIENT_ADDRESS     1
#define SERVER_ADDRESS     2

#define ServoHorizontalPIN 3   //horizontal motor
#define ServoVerticalPIN   4   //vertical motor
#define EndEffectorPIN     6   //end effector
#define pressuresensorPIN  A5

/*-----( Declare objects )-----*/
SoftwareServo HorizontalServo;
SoftwareServo VerticalServo;  // create servo objects to control servos
SoftwareServo EndEffectorServo;


// Create an instance of the radio driver
RH_NRF24 RadioDriver;

// Create an instance of a manager object to manage message delivery and receipt, using the driver declared above
RHReliableDatagram RadioManager(RadioDriver, SERVER_ADDRESS);

/*-----( Declare Variables )-----*/
int HorizontalInputReceivedTemp;    // Variable to store received Horizontal values
int HorizontalInputReceivedNew;
int HorizontalInputReceivedOld;
int HorizontalServoPosition;    // variable to store the servo position

int VerticalInputReceivedTemp;    // Variable to store received Vertical values
int VerticalInputReceivedNew;
int VerticalInputReceivedOld;
int VerticalServoPosition;      // variable to store the servo position

int EndEffectorInputReceivedTemp;
int EndEffectorInputReceivedNew;
int EndEffectorInputReceivedOld;
int EndEffectorServoPosition;      // variable to store the servo position

int HorizontalSteps = 2;          //variable to hold in how many degree steps the 3 motors move
int VerticalSteps = 3;
int EndEffectorSteps = 2;
int MasterDelay = 20;           //delay to let the servos reach position
int idx;                        //temp variable
unsigned long time;
```

```
uint8_t ReturnMessage[] = "JoyStick Data Received";  // 28 MAX
// Predefine the message buffer here
uint8_t pressuresensor[1];
uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];


//--------------------------------( SETUP Runs ONCE )------------------------------------
void setup()
{

  /*-----( Set up servos )-----*/
  HorizontalServo.attach(ServoHorizontalPIN);  // attaches the servo to the servo object
  VerticalServo.attach(ServoVerticalPIN);      // attaches the servo to the servo object
  EndEffectorServo.attach(EndEffectorPIN);

  // begin serial to display on Serial Monitor. Set Serial Monitor to 115200
  Serial.begin(115200);

  if (!RadioManager.init()) // Initialize radio. If NOT "1" received, it failed.
  {
    Serial.println("init failed");
  }

  // Defaults after init are 2.402 GHz (channel 2), 2Mbps, 0dBm
  //This part is to get the servo input values into the local variables
  //when the transmitter is turned on.
  //Wait for a message addressed to us from the client.

  while (!RadioManager.available())
  {
    continue;
  }


  uint8_t len = sizeof(buf);
  uint8_t from;
  if (RadioManager.recvfromAck(buf, &len, &from))
  {
    EndEffectorInputReceivedOld = buf[1] + 1;
    EndEffectorInputReceivedNew = buf[1];

    HorizontalInputReceivedOld  = buf[0] + 1;
    HorizontalInputReceivedNew  = buf[0];

    VerticalInputReceivedOld    = buf[2] + 1;
    VerticalInputReceivedNew    = buf[2];

    EndEffectorServo.write(EndEffectorInputReceivedNew);
    HorizontalServo.write(HorizontalInputReceivedNew);
    VerticalServo.write(VerticalInputReceivedNew);

  }
} // END Setup

//--------------------------------( LOOP runs continuously )------------
void loop()
{
  uint8_t c_value=200;
  pressuresensor[0] = map(analogRead(pressuresensorPIN),0,1023,0,50);
//Serial.println("pressure sensor : ");
 //Serial.println(pressuresensor[0]);

  if (RadioManager.available())
  {
    // Wait for a message addressed to us from the client
    uint8_t len = sizeof(buf);
    uint8_t from;
```

```
if (EndEffectorInputReceivedNew > EndEffectorInputReceivedOld)
{
  idx = EndEffectorInputReceivedOld + EndEffectorSteps;

  if(idx <= EndEffectorInputReceivedNew)
  {
    EndEffectorServo.write(idx);
    Serial.print("End effector servo : ");
    Serial.println(idx);
    EndEffectorInputReceivedOld = EndEffectorInputReceivedOld + EndEffectorSteps;
  }
}

else if (EndEffectorInputReceivedNew < EndEffectorInputReceivedOld)
{
  idx = EndEffectorInputReceivedOld - EndEffectorSteps;

  if(idx >= EndEffectorInputReceivedNew)
  {
    EndEffectorServo.write(idx);
    Serial.print("End effector servo : ");
    Serial.println(idx);
    EndEffectorInputReceivedOld = EndEffectorInputReceivedOld - EndEffectorSteps;
  }
}

//---------------------------------------------------------------------------
if (HorizontalInputReceivedNew > HorizontalInputReceivedOld)
{
  idx = HorizontalInputReceivedOld + HorizontalSteps;

  if(idx <= HorizontalInputReceivedNew)
  {
    HorizontalServo.write(idx);
    Serial.print("Horizontal servo : ");
```

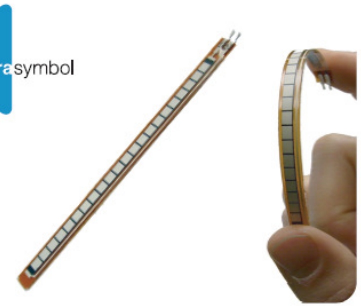
```
 if (RadioManager.recvfromAck(buf, &len, &from))

 {
   EndEffectorInputReceivedNew = buf[1];
   HorizontalInputReceivedNew  = buf[0];
   VerticalInputReceivedNew    = buf[2];

   Serial.print("New HorizontalServoPosition : ");
   Serial.print(buf[0]);
   Serial.print("New VerticalServoPosition : ");
   Serial.println(buf[2]);
   Serial.print("New EndEffectorServo : ");
   Serial.println(buf[1]);

   time=millis();
   Serial.print("start:");
   Serial.print(time);

   // Send a reply back to the originator client, check for error
   if (!RadioManager.sendtoWait(pressuresensor, sizeof(pressuresensor), from))
   {
     Serial.println("sendtoWait failed");
   }

   time=millis();
   Serial.print("end:");
   Serial.print(time);
   Serial.println("");
 }// end 'IF Received data Available

}// end 'IF RadioManager Available
```

```
      Serial.println(idx);
      HorizontalInputReceivedOld = HorizontalInputReceivedOld + HorizontalSteps;
    }
  }

  else if (HorizontalInputReceivedNew < HorizontalInputReceivedOld)
  {
    idx = HorizontalInputReceivedOld - HorizontalSteps;

    if(idx >= HorizontalInputReceivedNew)
    {
      HorizontalServo.write(idx);
      Serial.print("Horizontal servo : ");
      Serial.println(idx);
      HorizontalInputReceivedOld = HorizontalInputReceivedOld - HorizontalSteps;
    }
  }

  //--------------------------------------------------------------------------------
  if (VerticalInputReceivedNew > VerticalInputReceivedOld)
  {
    idx = VerticalInputReceivedOld + VerticalSteps;

    if(idx <= VerticalInputReceivedNew)
    {
      VerticalServo.write(idx);
      Serial.print("Vertical servo : ");
      Serial.println(idx);
      VerticalInputReceivedOld = VerticalInputReceivedOld + VerticalSteps;
    }
  }

  else if (VerticalInputReceivedNew < VerticalInputReceivedOld)
  {
    idx = VerticalInputReceivedOld - VerticalSteps;

    if(idx >= VerticalInputReceivedNew)
    {
      VerticalServo.write(idx);
      Serial.print("Vertical servo : ");
      Serial.println(idx);

      VerticalInputReceivedOld = VerticalInputReceivedOld - VerticalSteps;
    }
  }
  //--------------------------------------------------------------------------------

  SoftwareServo::refresh();        //refreshes servo to keep them updating
  delay(MasterDelay);              // wait for the servo to reach the position
 }

}// END Main LOOP
```

Appendix F

User manual: Flex sensor

spectrasymbol



FLEX SENSOR FS

Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
  - Robotics
  - Gaming (Virtual Motion)
  - Medical Devices
  - Computer Peripherals
  - Musical Instruments
  - Physical Therapy
- Simple Construction
- Low Profile

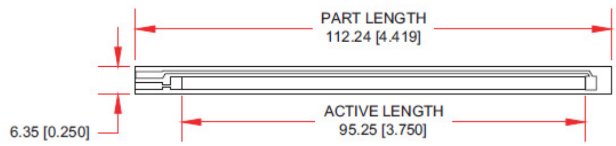
Mechanical Specifications

-Life Cycle: >1 million
-Height: ≤0.43mm (0.017")
-Temperature Range: -35°C to +80°C

Electrical Specifications

-Flat Resistance: 10K Ohms
-Resistance Tolerance: ±30%
-Bend Resistance Range: 60K to 110K Ohms
-Power Rating : 0.50 Watts continuous. 1 Watt Peak

Dimensional Diagram - Stock Flex Sensor

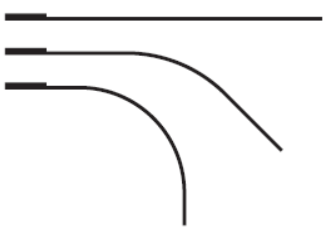


How to Order - Stock Flex Sensor

FS – L – 0095 – 103 – ST

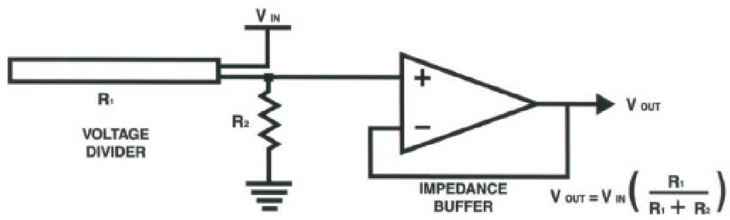
Series	Model	Active Length	Resistance	Connectors
FS = Flex Sensor	L = Linear	0095 = 95.25mm	103 = 10 KOhms	ST = Solder Tab

How It Works



Schematics

BASIC FLEX SENSOR CIRCUIT:

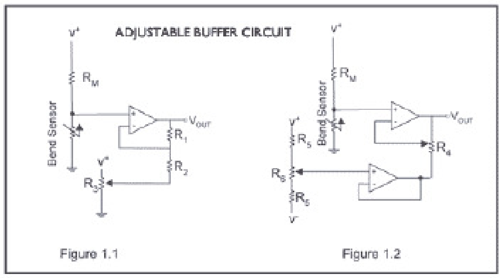


Following are notes from the ITP Flex Sensor Workshop

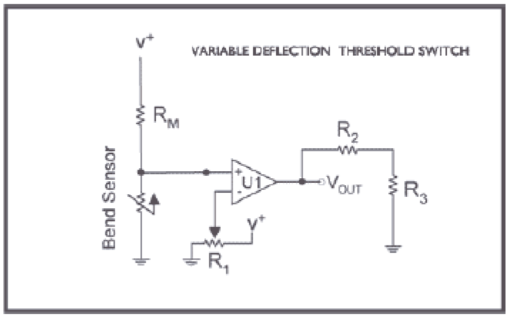
"The impedance buffer in the [Basic Flex Sensor Circuit] (above) is a single sided operational amplifier, used with these sensors because the low bias current of the op amp reduces errer due to source impedance of the flex sensor as voltage divider. Suggested op amps are the LM358 or LM324."

"You can also test your flex sensor using the simplest circut, and skip the op amp."

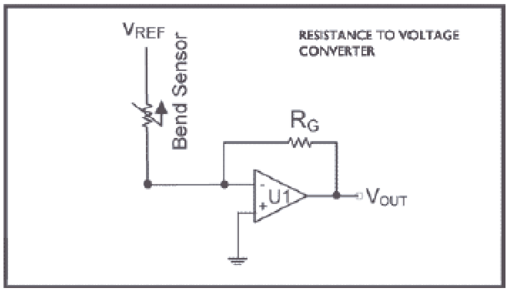
"Adjustable Buffer - a potentiometer can be added to the circuit to adjust the sensitivity range."



"Variable Deflection Threshold Switch - an op amp is used and outputs either high or low depending on the voltage of the inverting input. In this way you can use the flex sensor as a switch without going through a microcontroller."



"Resistance to Voltage Converter - use the sensor as the input of a resistance to voltage converter using a dual sided supply op-amp. A negative reference voltage will give a positive output. Should be used in situations when you want output at a low degree of bending."



Appendix G

Transistor

A TIP 120 NPN (Negative-Positive-Negative) transistor is used as a switch to control the power supplied to the motor and is shown in the figure 4.5. The NPN transistor is designed to pass electrons from the emitter to the collector (so conventional current flows from collector to emitter). The emitter emits electrons into the base, which controls the number of electrons the emitter emits. Most of the electrons emitted are collected by the collector, which sends them along to the next part of the circuit. The basic electrical schematic is shown in the figure 4.6.

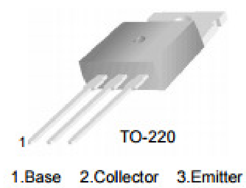


Figure 4.5: TIP120 NPN Darlington Transistor [39]

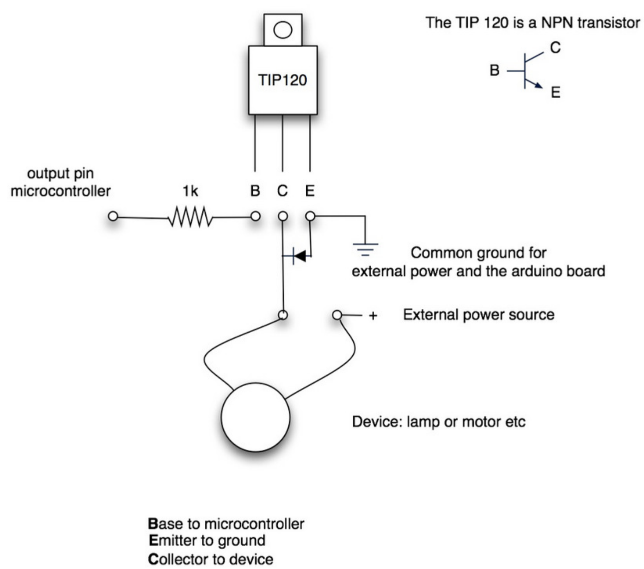


Figure 4.6: Electrical schematic of TIP120 transistor for switching applications [39]

Appendix H

Motor specifications

1501 MG analog servo motor specification

Dimensions

Size:	40.7 x 20.5 x 39.5 mm
Weight:	60 g

General specifications

Digital?:	N
Speed @ 6V:	0.14 sec/60°
Stall torque @ 6V:	17 kg·cm
Speed @ 4.8V:	0.16 sec/60°
Stall torque @ 4.8V:	15.5 kg·cm
Lead length:	11 in
Hardware included?:	Y

HS 7950 servo motor

Detailed Specifications

Control System: +Pulse Width Control 1500usec Neutral
Required Pulse: 3-5 Volt Peak to Peak Square Wave
Operating Voltage Range: 4.8-7.4 Volts
Operating Temperature Range: -20 to +60 Degree C (-4F to +140F)
Operating Speed (4.8V): 0.18 sec/60° at no load
Operating Speed (6.0V): 0.15 sec/60° at no load
Operating Speed (7.4V): 0.13 sec/60° at no load
Stall Torque (4.8V): 344oz/in. (22kg.cm)
Stall Torque (6.0V): 402oz/in. (29kg.cm)
Stall Torque (7.4V): 486oz/in. (35kg.cm)
Operating Angle: 45 Deg. one side pulse traveling 400usec
Continuous Rotation Modifiable: Yes
Direction: Clockwise/Pulse Traveling 1500 to 1900usec
Idle Current Drain (4.8V): 9mA at stop
Idle Current Drain (6.0V): 9mA at stop
Current Drain (4.8V): 220mA/idle and 3.8 amps at lock/stall
Current Drain (6.0V): 300mA/idle and 4.8 amps at lock/stall
Dead Band Width: 1usec
Motor Type: Coreless Carbon Brush
Potentiometer Drive: 6 Slider Indirect Drive
Bearing Type: Dual Ball Bearing MR106
Gear Type: Titanium Gears
Connector Wire Length: 11.81" (300mm)
Dimensions: 1.57" x 0.79"x 1.50" (40 x 20 x 38mm)
Weight: 2.40oz (68g)

Appendix I

Arduino Microcontroller Specifications

Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller. It can be simply connected to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started with the programming.

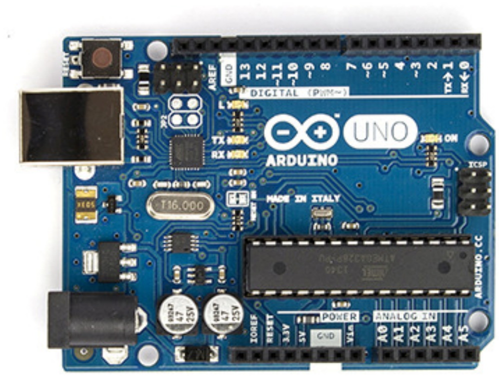


Figure K1: Arduino Uno microcontroller

The Arduino runs a simplified version of the C programming language, with some extensions for accessing the hardware. Programs are created in the Arduino development environment and then downloaded to the Arduino board. Code must be entered in the proper syntax which means using valid command names and a valid grammar for each code line. The compiler will catch and flag syntax errors before download.

Arduino protoshield

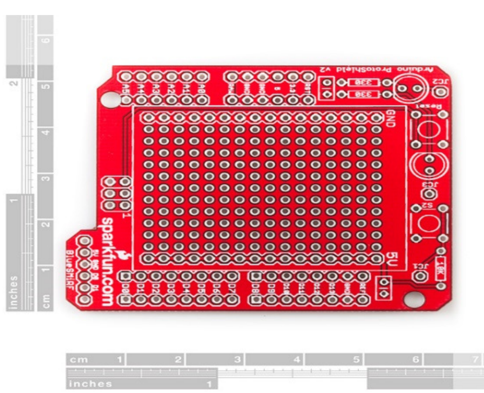


Figure K2: Protoshield for Arduino uno

Appendix J

Interview Questionnaire

An interview was conducted at Delaware manor which is an assisted living place in Grand rapids, MI. Five elderly people between the age of 60 and 70 were interviewed. Numerous questions were asked regarding the challenges faced by them in terms of day to day activities. The questionnaire and the answers recorded are shown in the table below:

Questions	Answers
1. What are the activities that are challenging on a day to day basis?	• Dusting • Reaching for things in shelves • Picking up objects from floor • Medicine reminder • Assistance in using internet
2. Would you be willing to have an assistive device for any of these activities?	• yes
3. What are your expectations from the assistive devices?	• The device should be easy to use with very minimum input • The device should be safe • Should be controllable from the wheelchair
4. Would you like the device to be fully automated or would you like to be involved in controlling it to some extent?	• 3 people mentioned that they would still like to have some physical activity while controlling it but should be absolutely less strainful. 2 people mentioned that they want the robot to do all the tasks for them.
5. Which is task that requires most assistance?	• Reaching for things in shelves and pick objects from different places
6. Would you be willing to have a mobile assistive device?	• Yes. It would be helpful in things such as grabbing the cell phone, Television remote control and turning on/off lights.

Appendix K

Bill of Materials

Component name	Quantity	Cost in $
Motor for base, HS 7950 (Servo city)	2	499.96
Servo motor for end effector, pololu	1	20
Motor for haptic feedback, pololu	1	20
Flex sensor, SEN-08606, sparkfun	1	13
Force resistive sensor	1	30
10 Kohm potentiometer	2	2
3.75” aluminum channel	1	4.59
9” aluminum channel	1	7.99
90 degree dual side mount (1 pair)	1	5.99
Round base A	1	6.99
90 degree hub mount bracket A	1	5.99
0.25” Socket head cap screw	25	1.69
32x1/4” Flat head screw	6	0.60
0.4375”socket head cap screw	25	2.09
6V, 3A Power supply	3	60
Arduino board	2	9.64
Wires	1	4
3D printed parts	12	62.82
Enclosure for circuitry	2	40
nRF24L01 wireless module	2	2
Shipping rough estimate	1	50.00
Publishing fees/expenses	1	200
Miscellaneous	1	155.71
Total		1205.06 dollars

Bill of material considering mobile slave robotic unit

Component name	Quantity	Cost in $
Vexnet cortex system bundle	1	400
Leeson Motors Gearmotor	2	454
DC motor drive	2	137.2
6 inch omni wheel	2	100
Caster	1	19
NEMA 3R enclosure	1	166
Miscellaneous		1000
Total		2276 dollars

Note: The parts for the mobile slave robotic unit base is chosen considering a load capacity of 80 lbs.

References

[1] Population division, DESA, "World population ageing 1950-2050," United Nations, Newyork, 2002.

[2] Congress of the United States, "Rising Demand for Long term-services and supports for elderly people," Congress of the United States, Washington D.C, 2013.

[3] H. S. Kaye, T. Kang and M. P. LaPlante, "Mobility Device Use in the United States," US Department of education, Washington,D.C, 2000.

[4] C. A. Stanger, C. Anglin, W. S. Harwin and D. P. Romilly, "Devices for Assisting Manipulation: A Summary of User Task Priorities," *IEEE TRANSACTIONS ON REHABILITATION ENGINEERING,* pp. 256-265, 1994.

[5] L. Mclnnes, "Importance of maintaining mobility to elderly health," *Aging Health,* pp. 165-167, 2011.

[6] R. LOUREIRO, F. AMIRABDOLLAHIAN, M. TOPPING, B. DRIESSEN and W. HARWIN, "Upper Limb Robot Mediated Stroke Therapy—GENTLE/s Approach," *Autonomous Robots,* pp. 35-51, 2003.

[7] Exact Dynamics, "Exact Dynamics," 2015. [Online]. Available: http://www.exactdynamics.nl/site/?page=about. [Accessed 12 03 2015].

[8] D.-J. Kim, Z. Wang, N. Paperno and A. Behal, "System Design and Implementation of UCF-MANUS—An Intelligent Assistive Robotic Manipulator," *TRANSACTIONS ON MECHATRONICS,* pp. 225-237, 2014.

[9] W. Uttal, "Teleoperators," *Scientific America,* no. 261, pp. 74-79, 1989.

[10] J. Batlle, P. Ridao and J. Salvi, "Integration of a Teleoperated Robotic Arm with Vision Systems using CORBA Compatible Software," *30th International Symposium on Automotive Technology and Automation,* Vols. Robotics, Motion and Machine Vision in the Automotive Industries, pp. 371-378, 1997.

[11] O. Ben-Porat, M. Shoham and J. Meyer, "Control Design and Task Performance in Endoscopic Teleoperation," vol. 9, pp. 256-267, 2000.

[12] Intuitive surgical, "Da Vinci Surgical System," [Online]. Available: http://www.intusurg.com/products/da_vinci.html. [Accessed 18 05 2015].

[13] P. Ben-Tzvi and P. Moubarak, A Mechatronic Perspective on Robotic Arms and End-Effectors, Croatia: InTech Europe, 2011, p. 248.

[14] W. Semere, M. Kitagawa and A. Okamura, "Teleoperation with sensor/actuator asymmetry: task performance with partial force feedback," *Proceedings of 12th International*

Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS, no. 04, pp. 121-127, 2004.

[15] I. Yamano, K. Takemura, K. Endo and T. Maeno, "Method for Controlling Master-Slave Robots using Switching and elastic Elements," *Proc. of the IEEE International Conference on Robotics and Automation,* pp. 1717-1722, 2002.

[16] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant and G. Robles-De-La-Torre, "Haptic interfaces and devices," *Sensor review,* p. 16, 2004.

[17] Siciliano and Khatib, Handbook of Robotics, Hiedelberg: Springer, 2008.

[18] V. J. Gohil, D. S. D. Bhagwat2, A. P. Raut and P. R. Nirmal, "ROBOTICS ARM CONTROL USING HAPTIC TECHNOLOGY," *International Journal of Latest Research in Science ,* pp. 98-102, 2013.

[19] M. V. Liarokapis, P. K. Artemiadis and K. J. Kyriakopoulos, "Telemanipulation with the DLR/HIT II Robot Hand Using a Dataglove and a Low Cost Force Feedback Device," in *IEEE Mediterranean Conference on Control and Automation* , Chania(Greece), 2013.

[20] M. Lelieveld and T. Maeno, "Design and Development of a 4 DOF Portable Haptic Interface with Multi-Point Passive Force Feedback for the Index Finger," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation Orlando, Florida,* pp. 3134-3139, 2006.

[21] C.-H. King, T. L. Chen, Z. Fan, J. D. Glass and C. C. Kemp, "Dusty: an assistive mobile manipulator that retrieves dropped objects for people with motor impairments," *Disability and Rehabilitation: Assistive technology,* vol. 7, no. 2, pp. 168-179, 2012.

[22] R. Huston, Principles of biomechanics, CRC Press, 2009.

[23] Spectra symbol, "Flex sensor," [Online]. Available: http://www.spectrasymbol.com/flex-sensor. [Accessed 26 06 2015].

[24] B. Pierson, "Haptic feedback arm for muscular rehabilitation," GVSU, Grand rapids, MI, 2012.

[25] Tekscan Inc, "Tekscan," [Online]. Available: file:///C:/Users/Archana/Downloads/FLX-FlexiForce-Sensors-Manual.pdf. [Accessed 15 03 2015].

[26] Tekscan Inc, [Online]. Available: https://www.tekscan.com/sites/default/files/FLX-FlexiForce-Sensors-Manual.pdf.. [Accessed 25 06 2015].

[27] Future electronics, "Servo motors control and arduino," 2015. [Online]. Available: http://www.inmoov.fr/wp-content/uploads/2015/02/Introduction-to-Servo-Motors-Arduino.pdf. [Accessed 26 05 2015].

[28] Society of Robots, "Society of robots," [Online]. Available: http://www.societyofrobots.com/actuators_servos.shtml. [Accessed 15 03 2015].

[29] C. J. Ketcham and G. E. Stelmach, "Movement control in older adult," *National academy of sciences,* 2004.

[30] ServoCity, "SPG400A-CM Channel mount gearboxes," [Online]. Available: https://www.servocity.com/html/spg400a-cm_channel_mount_gearb.html#.VSnBAPnF99A. [Accessed 11 04 2015].

[31] A. P. NANDA, "Design & Development of a Two-jaw parallel Pneumatic Gripper for Robotic Manipulation," National Institute of technology Rourkela, Orissa, 2010.

[32] "Pololu Robotics and Electronics," Pololu, [Online]. Available: https://www.pololu.com/product/1057. [Accessed 15 03 2015].

[33] Nordic semiconductor, "nRF24L01 Product Specification," [Online]. Available: http://www.nordicsemi.com/jpn/content/download/2730/34105/file/nRF24L01_Product_Specification_v2_0.pdf. [Accessed 25 06 2015].

[34] Nordic semiconductor, "nRF24L01 Single Chip 2.4GHz Transceiver Product Specification," [Online]. Available: file:///C:/Users/Archana/Downloads/nRF24L01_Product_Specification_v2_0%20(1).pdf. [Accessed 26 05 2015].

[35] A. Pradeep, Director, *Proof of concept.* [Film]. USA: GVSU, 2015.

[36] Y. S. Choi, T. Deyle and C. C. Kemp, "A List of Household Objects for Robotic Retrieval Prioritized by People with ALS (Version 092008)," Healthcare Robotics lab, Georgia, 2009.

[37] National Society of Professional Engineers, "Code of Ethics for Engineers," Virginia, 2003.

[38] w. N. and B. N., "Inclusion of bio-engineering into existing codes of ethics," *Proceedings of the 2009 American Society of Engineering Education (ASEE) North Central Section (NCS) spring conference Grand Rapids, MI, USA,* no. April, 2009.

[39] Adafruit, "TIP120/TIP121/TIP122 NPN Epitaxial Darlington Transistor," Fairchild semiconductor corporation, [Online]. Available: http://www.adafruit.com/datasheets/TIP120.pdf. [Accessed 26 05 2015].

[40] R. J. Hatke, T. R. Prohaska and S. E. Furner, "Older Adults and Assistive Devices: Use, Multiple-Device Use, and Need," *Journal of Aging and Health,* no. 99, p. 10, 1998.

[41] S. Allin, "Recent trends in the development and evaluation of assistive robotic manipulation devices," *The physical medicine and rehabilitation clinics of North America ,* pp. 59-77, 2010.

[42] A. J. Pearce, B. Adair, K. Miller, E. Ozanne, C. Said, N. Santamaria and M. E. Morris, "Robotics to Enable Older Adults to Remain Living at Home," *Journal of Aging Research,* vol. 2012, p. 10, 2012.

[43] K. GANDHI, H. KOTAK, S. JOSHI and V. PANDITA, "MOTION CONTROLLED ROBOTIC ARM," *International Journal of Electronics andCommunication Engineering (IJECE),* vol. 2, no. 5, pp. 81-86, 2013.