Real-time Control of Robot Arm Based on Hand Tracking Using Leap Motion Sensor Technology

A Major Qualifying Project
Submitted to the Faculty of
Worcester Polytechnic Institute
in
partial fulfillment of the requirements for the
Degree in Bachelor of Computer Science
and Mechanical Engineering
By

Connor Bolton

Maria Cantos

Pengfei Tang

Shanghai University Advisors:

Professor Zhang Jinsong

Zhijian Fan

WPI Project Advisor:

Prof. Kevin Rong

Abstract

With technology advancing at an astronomical rate, we are witnessing events today that a few years ago were only conceptualized in movies. From humanoid robots to the famous Mars Rover, the possibilities are endless. Leap Motion is an example of ground-breaking technology that has the potential to change the way we control machines and therefore, how we control our world! Leap Motion is a sensor that is currently used to navigate through a personal computer with just hand gestures. For our MQP, the team has used this technology to control a physical robot arm.

For the second part of this project, the team used SolidWorks to design a six-degree-of–freedom robot arm with five human-like fingers. The robot was designed to be controlled by Leap Motion, with human hand gestures as the input. It utilizes all of the sensor’s features, including the simultaneous control of all five fingers. The robot could be used for virtually any application, including research or service in the medical or military fields.

Acknowledgements

We would like to thank our Worcester Polytechnic Institute advisor Kevin Rong for making our project and stay in China possible. Also, for his guidance and support throughout the duration of the Major Qualifying project.

We are grateful to Professor Zhang Jinsong for providing us with a lab, and the necessary resources to successfully finish our project.

A special thanks to our project mentor Chris, for his kindness and constant support throughout our design process and programming.

Our WPI team is grateful to our Chinese partners: Rico, Taylor, and Jane for welcoming us into their culture and making sure we were always comfortable. Thank you for working with us side by side on this project.

Table of Contents

1. Introduction

1.1 Project Statement

Our group from Worcester Polytechnic Institute (WPI), teamed up with students from Shanghai University to complete our Major Qualifying Project (MQP). The project is split up into two sections, programming and design. The programming task was to control a six degree-of-freedom robotic arm with Leap Motion (LM), a real time motion-tracking sensor. The second part was to utilize SolidWorks to design a six-degree-of-freedom robot arm with five fingers.

The creators of Leap Motion designed this technology with the intension of developing an entirely new way of navigating computers and playing games using hand gesture control. Our goal as engineering students is to apply this new technology to control an existing robotic arm and further understand the value and potential uses that may result from remote controlled technology in the real world. Figure 1.1 depicts the tools and steps the team used to control the robotic arm, represented in order from left to right.

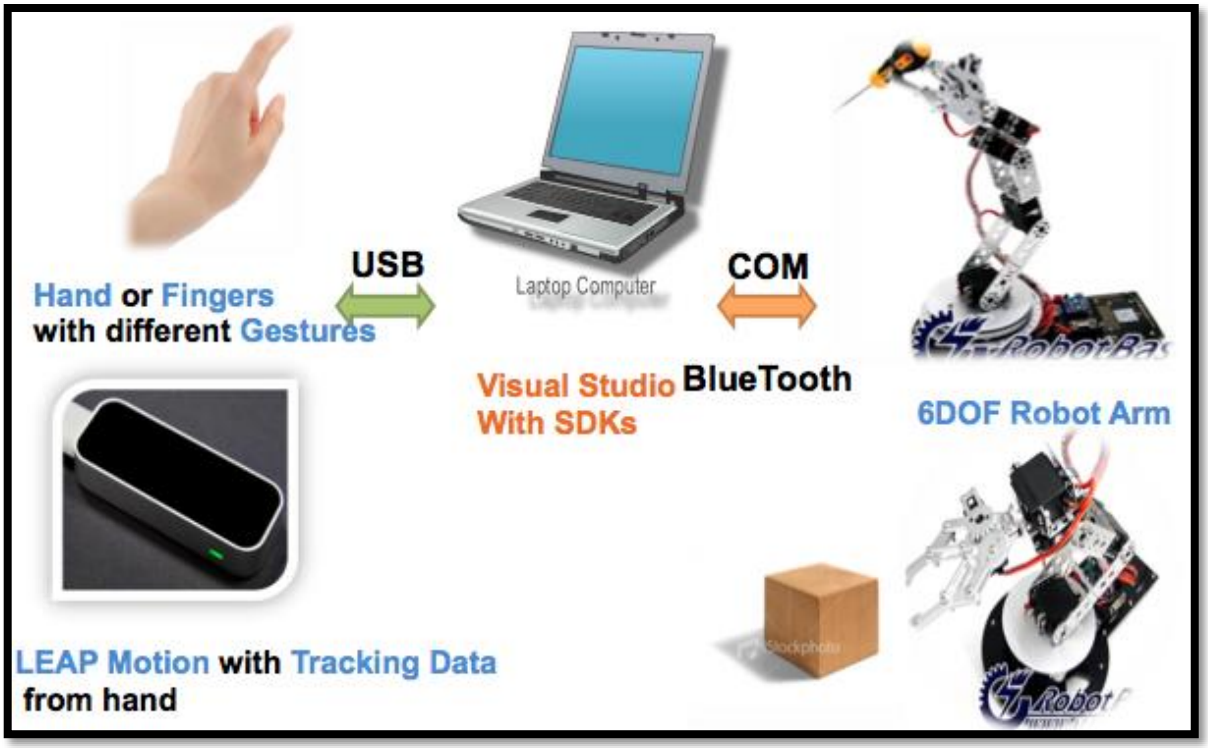


Figure 1.1 Representation of steps taken to complete the project

1.2 Motive

Many world accomplishments have been made possible through the use of remote controlled technology. These robotic jobs are sometimes tasks that are too dangerous for humans. An example is a newer branch of robotics that allows human control from a distance ranging from a few feet to hundreds of miles away. Scientists have been able to successfully explore the Earth's moon and even Mars! [1] Doctors have the ability to operate medically on humans with greater precision [2]. Also, military geared scientists are working on a new type of infantry designed to help save soldier's lives [3]. The research and the robotic possibilities are endless.

One of the greatest accomplishments in this era was NASA's space excursion to Mars. NASA sent its Mars explorer Curiosity, shown in figure 1.2, to do research and to explore the regions of space never before traveled by man. Launched from Florida's Cape Canaveral site on November 26, 2011, Curiosity landed on Mars nearly eight months later. As a testament to today's robotic technology, NASA's Curiosity is large in size, about the size of small utility vehicle (SUV) and weights just about nine hundred kilograms. Curiosity is equipped with a seven-foot long arm designed to drill into Mar's surface, pick up the materials, analyze, and take pictures of objects with its HD camera. The technology within Curiosity enabled the robot to break down minerals and analyze the element composition. The information was sent back to NASA where scientists would analyze the data to understand this new territory. The mission has been so successful that Curiosity continues to explore Mars today, and is expected to function until 2016 [1].

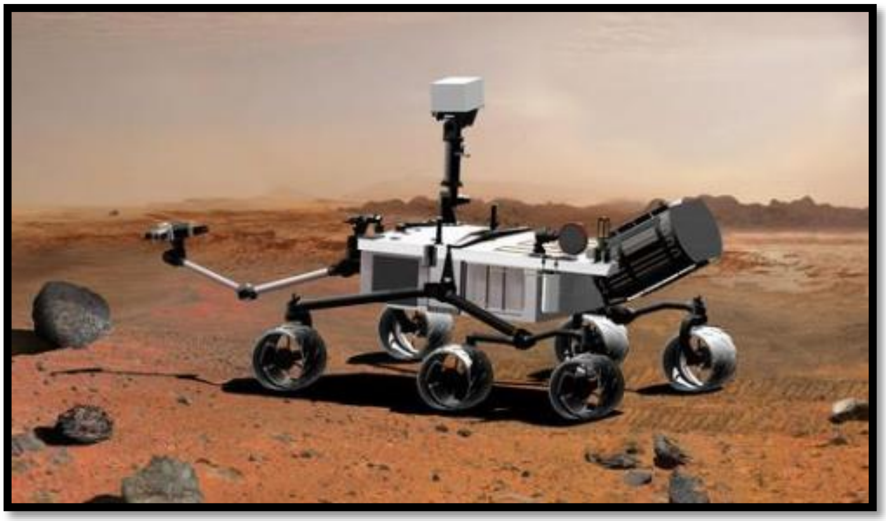


Figure 1.2 Mars Rover

A new and widely used applications in the medical field is the ‘da Vinci Surgical System.’ The surgical device is a robot with multiple arms, which allows doctors to operate on a patients from distance. The doctor uses joysticks to move each arms to precise locations not possible even the steadiest human hands. Each arms is equipped with different tools including: cameras that allow the operator to see a three-dimensional view of the patient with ten times the zoom, surgical knives, and surgical tools. The ‘da Vinci’ allows for a faster and more precise surgery. The device also allows for minimal invasiveness, which dramatically lowers risk of infection [3].

Figure 1.3 da Vinci Surgical System

The United States Army is taking further steps to protect their soldiers by taking advantage of this new technology and further advancing its capabilities. They are using robotic arms and robotic vehicles to help aid in combat and national security by developing remote controlled robot infantry, shown in 1.4. These robots are not only weapons but part of the squad, controlled with computers embedded within the soldier's vests [3].

Northrop Grumman is an American defense technology company [4]. One of its directors, Phil Coker states that the CaMEL (for Carry-all Mechanized Equipment Landrover), can run for 24 hours on three-and-a-half gallons of fuel, and can be equipped with a grenade launcher, an automatic weapon and anti-tank missiles. I can protect soldiers by eliminate targets up to 3.5 kilometers away using daylight telescopes and thermal imaging. This is an example of one of the robots that are estimated to join the field in the next five years [3].

Figure 1.4 Military Robot

2. Background

2.1 History of robotics

A robot can be defined as "a programmable, self-controlled device consisting of electronic, electrical, or mechanical units. More generally, it is a machine that functions in place of a living agent [5]." For centuries, humans have been fascinated by the idea of robotics and automation. Today, universities, private companies, and military powerhouses work tirelessly to create human-like machines to do every day human-like tasks for us, as well as provide unique entertainment.

Archytas of Tarentum, a philosopher and mathematician, is credited for building the world's first robot in 400 B.C. He built a wooden dove that could flap its wings and fly approximately 200 meters, fueled only by steam [6]. Today, we have advanced our technologies to more evolved robots capable of building, rescuing, flying, and swimming underwater. An example of one of the most advanced autonomous, humanoid robots is Atlas, shown in figure

2.1. Atlas is designed to maneuver through rough terrain as well as move away any obstacle that stands in its way. He has twenty-eight degrees of freedom solely actuated through hydraulics that allow it to lift, manipulate, carry objects, and climb stairs with its hands and feet. Atlas' head is equipped with stereo cameras and laser range finders that allow it to navigate through its environment. This is truly one of the highest pieces of robotic technology out there today [7].

Figure 2.1 Atlas Robot

2.2 What is 6-DOF?

A solid body has six degrees of freedom when it has no restrictions on where it is allowed to move in space. The six degrees of freedom include transitional movement along the x, y, and z axis, as well as roll, yaw, and pitch about these axis, shown in figure 2.2. The use of the six degrees of freedom in the engineering world is very important because it allows for more complex and useful designs. Robotic machines with six degrees of freedom are more agile and have the ability to be more precise in job specific movements such as spot welding, material placement, or surgical cutting.

In the robotics field, there are multiple designs of robots that facilitate six degrees of freedom. Each robotic design is usually job specific to be as efficient as possible in the environment that it is placed in. Simplicities of design and structure have been considered in the design of robotic arms so proper background research is not needed. The simplicities have been considered in manufacturing and control and have led to designs of robots with only "volute or prismatic joints and orthogonal, parallel, and for intersecting joint axes [8]." All systems are typically designed to focus an end-effector using all six degrees of freedom if necessary. This will provide a direct relationship between the actuator positions in space to the configuration of the complete body. This relation is defined by the kinematics and inverse kinematics of the robot. A robotic arm is named and described by their allowable degrees of freedom in space and along a coordinate axis. The degrees of freedom suggest the flexibility of the arm and its complete movement. The number of degrees of freedom refers to the "number of single-axis rotational joints in the arm, where higher number indicates an increased flexibility in positioning a tool [8]."

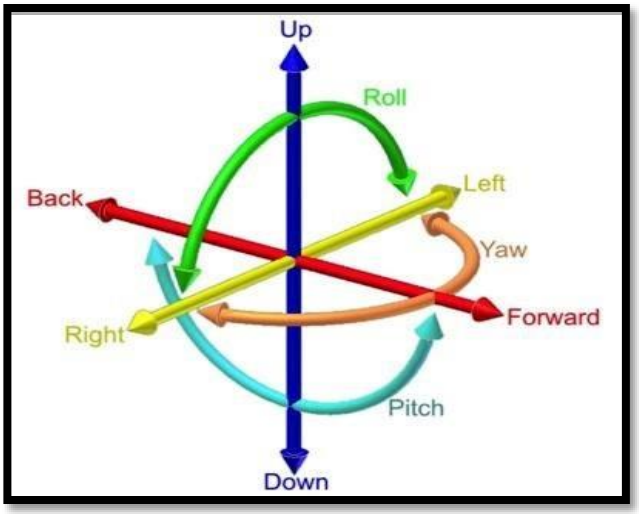


Figure 2.2 Depiction of the six degrees of freedom

One of the most common types of robots are serial manipulator systems, shown in figure 2.3, and parallel manipulator systems. These robots are the most common for use in industrial applications and consist of a series of linkages connected by motor-activated joints that extend from a base to an end effector. The end effector can be as simple as a needle or as defined as a robotic claw or robotic hand. A serial manipulator system takes after an anthropomorphic arm, it consists of a shoulder-simulating design, an elbow-simulating design, and a wrist-simulating design. These three joints allow for a proper six degree of freedom design on the engineering side, but also a proper six degree of freedom movement and accessibility on the kinematics and programming side. Serial designs are perfect for pick and place applications in the manufacturing community. Although very specific and near perfect for some applications, there are disadvantages to having a serial manipulator system design. Firstly, the serial manipulator is not very strong and can easily bend if stressed too much. Because of the lack rigid bodies most of the pressures are put on the actuators causing the serial design to have a low weight tolerance. Secondly, multiple errors in distance and dimension occur because of the linkages and actuator design. All in all, there is a distinctly low effective load that the serial manipulator system can actually manipulate [9].

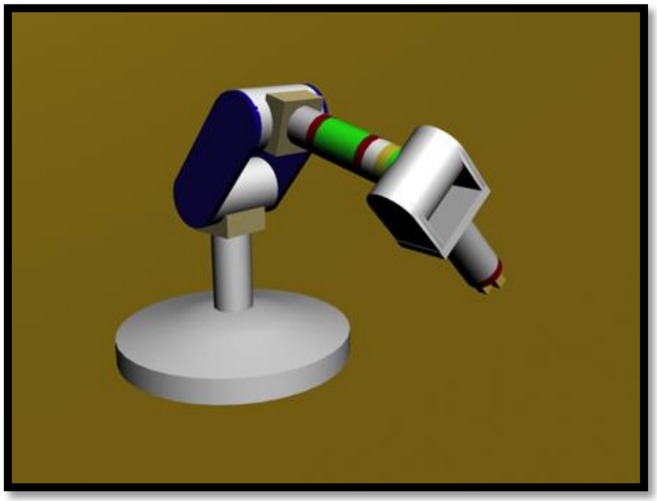


Figure 2.3 Robotic Serial Manipulator

2.3 Sensors

Sensors are used to read the surrounding environment and provide feedback knowledge to humans, robots, or any type of autonomous systems. They are used in everyday life and can range from very simple photoresistors to complicated range sensors. These devices are integrated in heating and cooling systems to ensure perfect temperature for humans. GPS' are positioning devises that allow drivers to navigate to almost any location. Miniature sensors are integrated in smart phones to provide countless hours of games and entertainment to consumers of all ages [10].

Robots rely on sensors to interact with their environment. Stereo cameras allow robots to have depth perception, sound sensors give robots the ability to respond to voice commands, gas and temperature sensors tell them if the environment is safe for humans, etc. The possibilities are still never ending as technology continues to evolve [10].

2.4 About Leap Motion

2.4.1 Introduction

Leap Motion is an external device equipped with infrared sensors that focuses on tracking a human's fingers in the space above the sensor. Leap motion is packed with an extreme amount of power. The sensors located within the sensor are incredibly fast and accurate, allowing the user to navigate his or her own computer with as much freedom as he or she wants. The Leap Motion has the ability to replace the computer mouse completely, replacing all functions and adding its own functionality to the interaction of a virtual machine.

Inventor and founder of Leap Motion technologies, David Holz, became frustrated by the limitations of the keyboard and the limitations of the mouse while studying for his Ph.D. in mathematics at the University of North Carolina. In 2008, he began designing a better and more sophisticated way of interacting with his computer. A few years later, after completing his first leap motion prototype, he paired up with entrepreneur Michael Buckwald in hopes of finding funds to further develop this amazing technology [11].

Although the first prototype was large and extensive to set up, the founders impressed angel investors with their product. The prototype was fast and had little lag time. Today, 12 years later, the company has grown to 80-plus employees and has distributed leap motion devices to several main stores such as Best Buy [11].

Leap Motion's design goal was for the purpose of one day replacing the keyboard and mouse from everyday use. So far, the latest design allows individuals to navigate their computers by using their fingers to flip through photos and music playlists as well as browsing and scrolling through webpages and articles. There is also an extensive application store called Leap Motion App Home, where Leap Motion users can download games and educational activities for kids. The Leap Motion founders allow for an interactive study and co-op development by allowing their users to create their own games and share these games with the community [12].

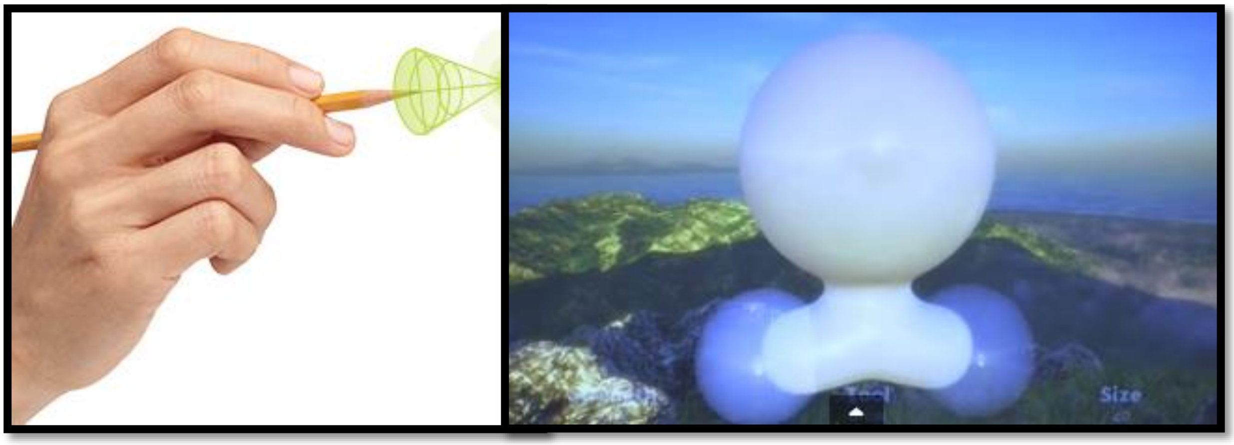


Figure 2.4 Leap Motion Applications

Games range from navigating cars and airplanes to slicing and shooting objects in the air by pointing with a finger. Educational games allow kids to dissect frogs, sculpt 3D objects or paint on virtual canvases using a physical paintbrush in the air. Some games teach the user to virtually play instruments such as the guitar, harp and drums [12].

2.4.2 Software and sensors

The Leap Motion sensor has drastically decreased in size over the years. Leap Motion has reduced to a compact size of 13mm x 13mm x 76mm and only weighs 45 grams. It is designed to sit flat between the user and the computer. The universal USB connection allows the products to be easily used in any workstation. The sensor is based on the right-hand Cartesian coordinate system, with the origin centered at the top center of the device [13].

Instead of using traditional depth sensors, Leap Motion has two Infrared stereo cameras, and three infrared LEDs, which track light with a wavelength of 850nm. The LEDs project a 3D pattern of Infrared light in the positive Y-direction to intersect the user's hand. The lenses on the Leap Motion cameras bend the reflected light rays into the sensor by projecting a large field of view shaped like an eight cubic foot inverted triangle. The height of the interaction space is about 60cm above the Leap Motion. It is limited by the LED intensity, which is restricted by the maximum current that the USB is able to draw from a computer [14].

Leap Motion then records the image into its memory in gray scale at a specific pixel location. It has the capability of tracking all ten fingers to the accuracy of 1/100th of a millimeter at 200 frames per second. It stores the sensor data into its memory and uses the software, called Leap Motion Service, in the user's computer to perform mathematical algorithms that interpret sensor data. To correct lens imperfections, the Leap Motion software uses a calibration map that provides data to calculate the true angle of the original light ray. The software automatically ignores extra objects such as the user's head, and compensates for uneven lighting. It also had the capability of filtering images to ensure a smooth transition of data. The images read by Leap Motion are converted to snapshots, or frames, before being transported to the Leap Motion control panel, which can be used by the programmer [14].

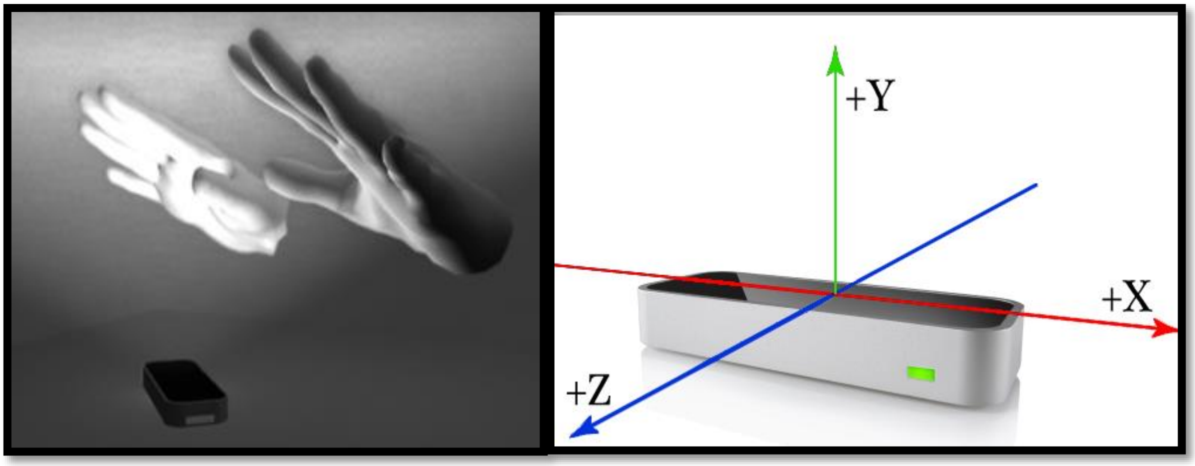


Figure 2.5 Leap Motion Controller

4.3 Leap Motion Capabilities and Limitations

Leap motion has many capabilities, not just including the original function of the sensor that the founders originally planned for. We know that Leap motion can easily replace a computer mouse because of its hover and touch zone feature. The user can aim within the screen by keeping his hand within the hover zone, but as soon as he reaches into the touch zone shown in figure 2.7, the user clicks or selects an object. The user can double-click on an object by quickly reaching into the touch zone twice in the same place [15].

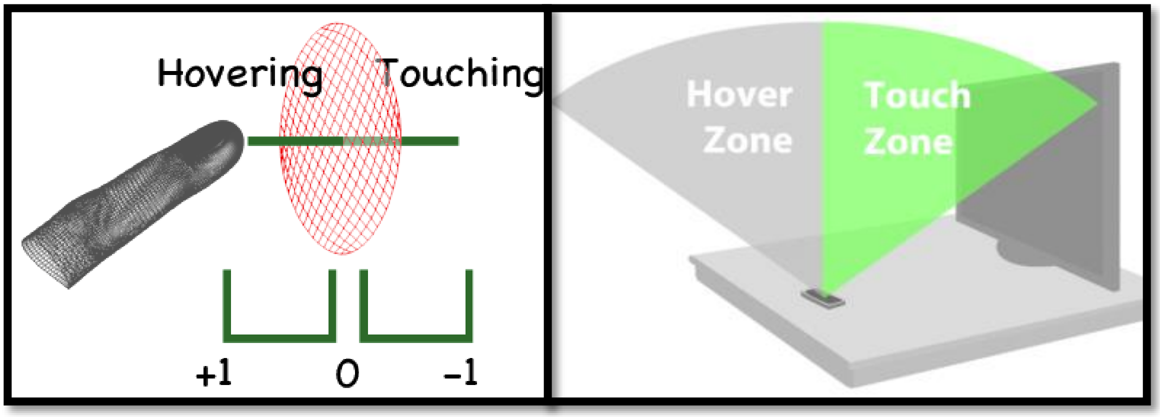


Figure 2.6 Hover Zones

The Leap Motion sensor's field of view is an inverted pyramid, limiting the range of motion more as the hand approaches the sensor. As the hand rises, the wider the range of motion becomes. The field view is of 150 degrees and can sense from 25 mm to 600mm directly above the device. The red box shown in figure 2.8, is called the interaction box. As long as the hand remains within that box, Leap Motion will be able to accurately read detect its position. The user can also scale the coordinate system to make the device more or less sensitive, depending on the chosen application [15].

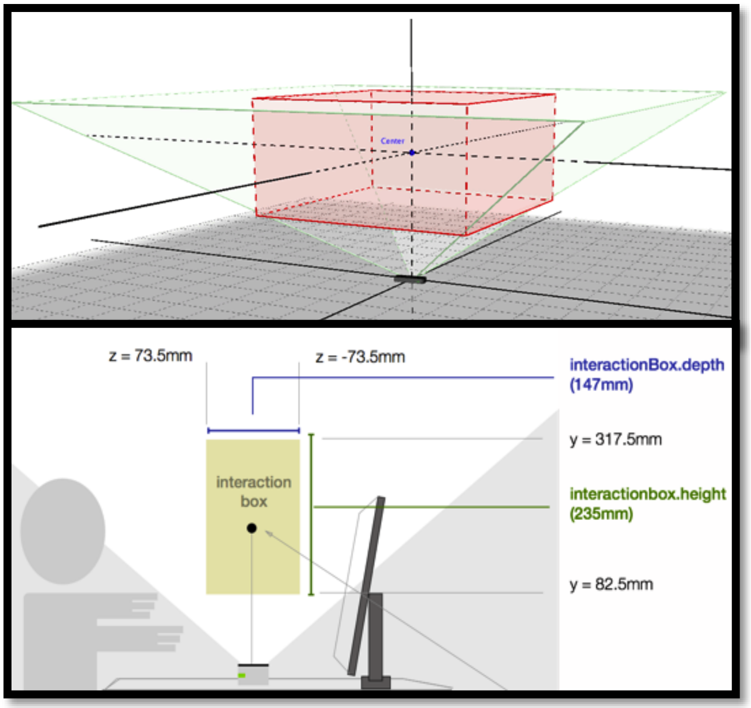


Figure 2.7

Field of View

2.4.4 Future plans and aspirations

Apart from continuously improving the original leap motion device, the company started a new addition called Leap Motion Virtual Reality. The Leap Motion sensor works with the Virtual Reality goggles to enhance the user's 3D experience through goggles. The user can feel like he is inside the computer, while he manipulates objects. The goggles increase the field of view 35 degrees more that the Oculus Rift's, a competitor company stereo displays. It also has night-vision capabilities through the use of infrared sensors. The possibilities are endless.

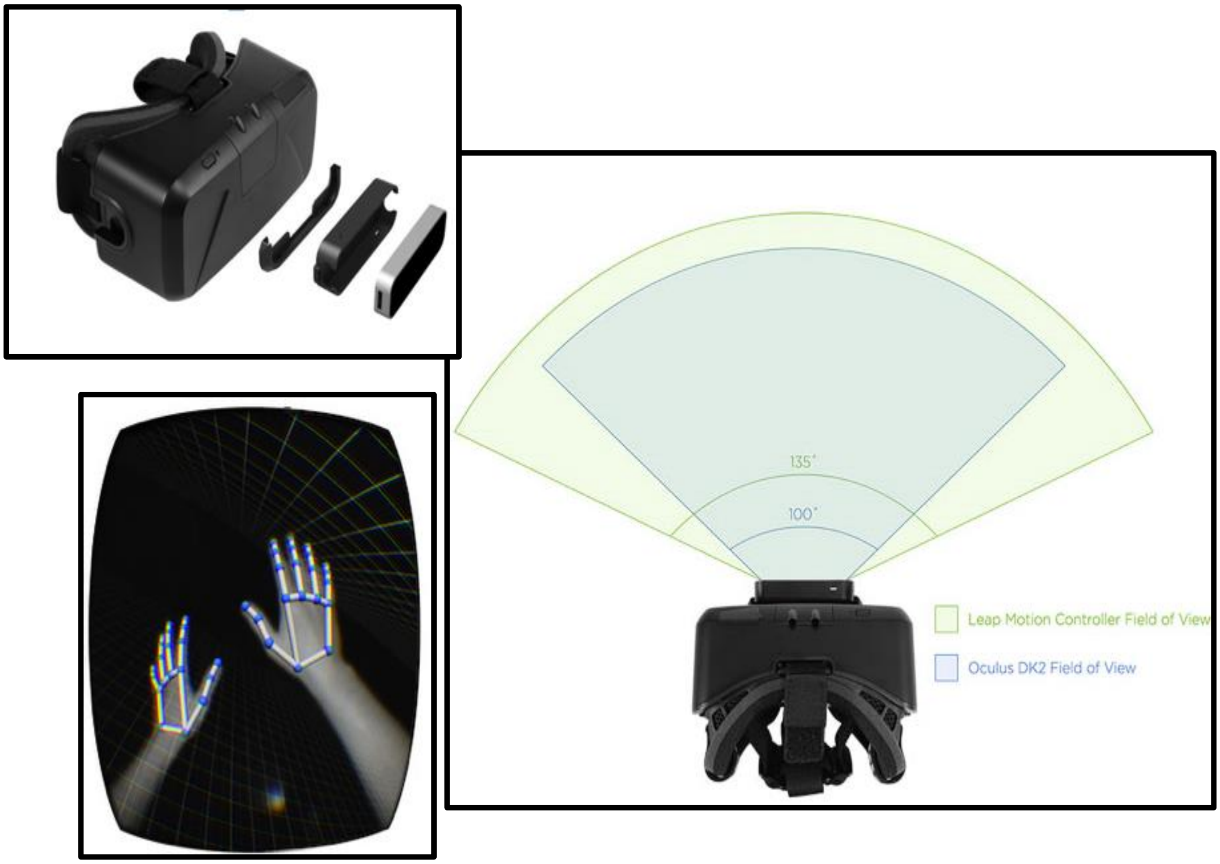


Figure 2. 8 Virtual Reality Equipment

2.5 6-DOF Robotic Arm

Our team assembled the 6-degree of freedom robotic arm shown in figure 2 [17]. Its 6 servomotors allows for 6-degrees of freedom, as pointed out by each arrow. Motor 1 rotates the entire robotic arm and is controlled by a panning hand motion in the x-direction. Motor 2 lifts and lowers and the entire arm when the operator moves their arm along the y-axis, directly above the sensor. Motor 3 also raises and lowers the arm for more precise movement when the operator moves its hand in the Z direction. Lastly, motor 6 opens and closes the robotic gripper when the operator forms a pinching gesture with his/her thumb and index finger. Motors 4 and 5 were not used.

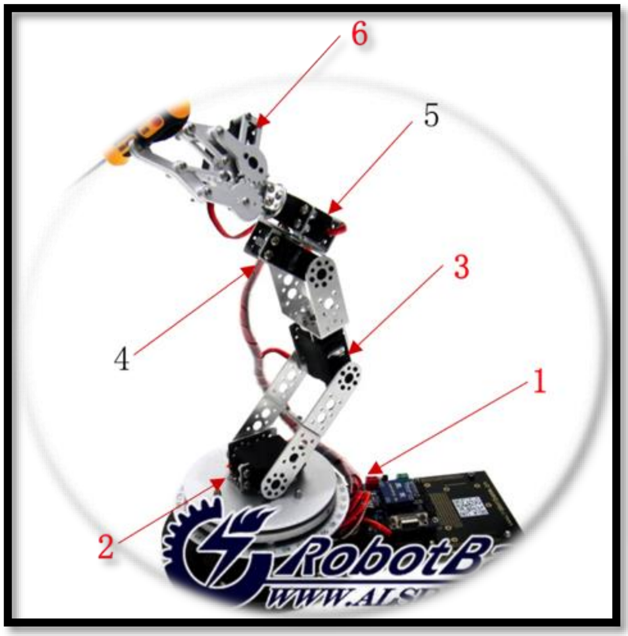


Figure 2.9 Six-Degree of Freedom Robot Arm

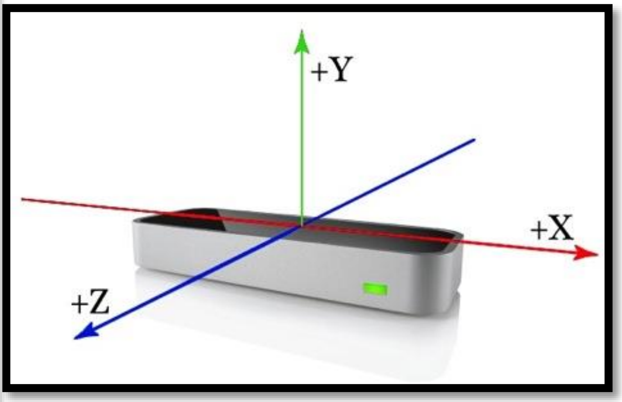


Figure 2.10 Leap Motion Sensor

Each motor is controlled by the microcontroller through PWM wires. They receive commands from the computer program via a Bluetooth device, shown in Figure? The Bluetooth is powered by 5 volts, taken from the microcontroller and receives wireless commands that manipulate the physical arm. Alternatively, the black serial cable shown in figure 2 can also be used to establish communication between the computer and the board. Lastly, the microcontroller is powered by an external power supply, which provides both 12 and 5 volts to the terminal block on the board. The power supply receives 220volts from the wall outlet.

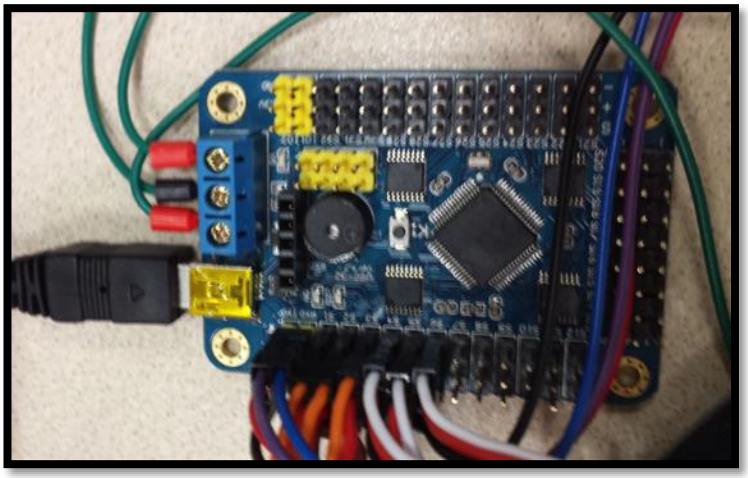


Figure 2.11 Microcontroller

Figure 2.12 Bluetooth

Figure 2.13 Power Supply

2.6 Biomechanics of a Human Hand

There are several different actuating options to choose from to actuate a hand model. We discarded the use of pneumatics because it could weigh down the system and add problems such as air leakages. If we used motors to actuate every joint, we would also add weight and bulkiness. During research, we came across tendon-driven designs being developed all over the world and decided to implement this idea on our hand. Although it seemed complex at first, we quickly realized that is was an efficient method to making our design as human-like as possible. In order to understand this approach, it was worthwhile to research further into the biomechanics of a human hand.

The typical human hand is equipped with five fingers. They aid in everyday functions, such as brushing your teeth, grasping your coffee cup, using utensils at meal times, and typing on a computer. The hands are a human necessity that allowed the world that we live in to be built. Human hands build the great Pyramids of Egypt, assembled the Great Wall of China, and carved ancient Rome. Today, we use our hands to build the great world that we live in. We continuously build the rising cities around the world, we aid in the development of new technologies and implementation of new technologies that are save energy, save lives, and save time.

Biomechanics refers to the study of the mechanical principles of all living organisms. Biomechanics is directly related to the organic movements and the organic structure of any organism, as simplistic as a snake to as intricate as a human hand. Biomechanics uses orthodox engineering sciences and engineering mechanics to analyze the organic movements. When speaking about human body movements, biomechanics refers to the internal bone structure and the connections that bones and joints have along with the relation of bone movements. In order to have a fully functional and efficient humanoid robotic hand and arm, it is imperative to fully research the biological design and functionality of the human hand [17]. Studying the fingers, the palm, and the immediate relation of arches due to the movements show the functionality and the usefulness of the hand.

2.6.1 Bones

A human forearm has two main bones called the Radius and the Ulna, shown in figure 2.14. They supply support for the rest of the hand through its skeletal frame consisting of twenty-seven bones. The wrist contains eight rounded carpal bones, which act as a tunnel feeding nerves, tendons, and ligaments through the palm to the fingertips. Five metacarpal bones extend out of the carpus to form the palm. The five metacarpal bones are numbered I to V, starting from the thumb, being Metacarpal 1, and ending at the pinky finger, dubbed as Metacarpal 5. Each finger contains three long, thin bones called phalanges, but the thumb contains only 2. The first set closest to the palm from your knuckle moving upward, is called the Proximal Phalanges, next are the Intermediate Phalanges, and the finger tips, which are called the Distal Phalanges. The respective joints that connect the phalanges are the DIP, PIP, and MCP. These are hinge joints that allow the hand to open and close [19]. See figure 2.16.

2.6.2 Tendons

Human joints are actuated through tendons, which allow the muscles to move, and ligaments that work to stabilize the movement of the joints. Both the tendons and ligaments are comprised of tough, elastic tissue and are attached to both sides of every finger and thumb. The extensor tendons start as muscles and contract to open the hand. They begin at the forearm, travel past the back of the wrist, and end as extender hoods that flatten as the fingers straighten. The tendons that start on the palm side are flexor tendons and contract to bend the fingers [20]. In order to make our design as human-like as possible, our hand needs to extend and contract with the same degrees of motion as a human hand. The table in figure 2.15 shows the average degree of freedom of each joint.

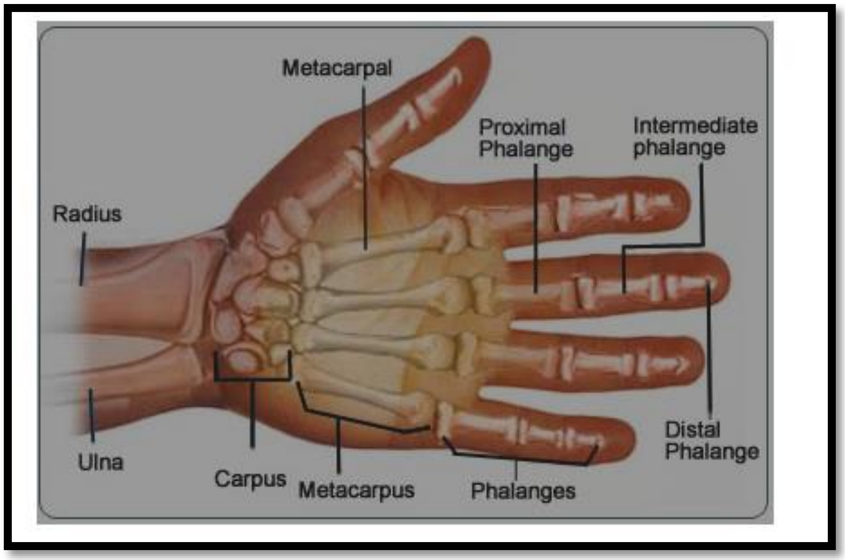
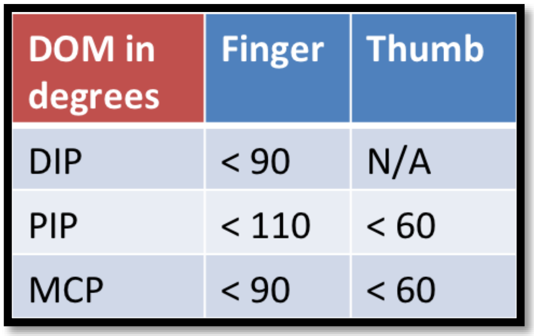


Figure 2.14 Bone Structure of a Human Hand



DOM in degrees	Finger	Thumb
DIP	< 90	N/A
PIP	< 110	< 60
MCP	< 90	< 60

Figure 2.15 Degrees of Motion of the Human Hand

Figure 2.16 Finger Joints and Phalanges

2.6.3 Arches

The body is capable of moving across and along multiple planes. The movement perpendicular to these planes can be described as flexion and extension. In more detail, “flexion is a motion in which the angle of the joint involved decreases [21].” Extension is the movement that will increase the angle. In regards to the hand, flexion would be the movement of closing your hands, and extension would be the movement when the hand opens. More precisely, the fingers of the hand are considered hinge joints and are located at the interphalangeal joints [22].

The human hand is the most extraordinary and adaptive extremity of the human body. It has the ability to grab, grip, and maintain hold of all sized and weighted objects, from a small pencil in the classroom to heavy weighted bars in a gym. The human hand performs many different grips due to the combination and usage of fingers and arches within the palm of the hand. There are three different arches that allow for the precision and functionality of the fingers during everyday tasks of the hands. The three arches, shown in figure 2.17, include the proximal transverse arch, the distal transverse arch, and the longitudinal arch [18].

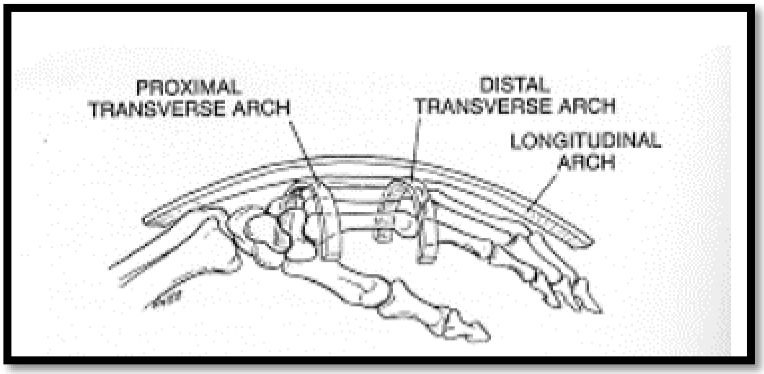


Figure 2.17 Arches of a Human Hand

3 Programming with Leap Motion

3.1 Background

The Leap Motion sensor contains a camera sensor that takes stereographic pictures of the objects in front of it. Leap Motions internal software then scans the images to look for palms, fingers, and pointable objects such as pencils. All these are also combined and recorded as one frame at each position in space. LM scans and sends the PC about 300 frames per second. At this speed, the LM device is able to see and use information quickly and accurately, making for great response times [23].

Once all objects in a frame are identified, they are assigned a unique internal ID that the software uses to keep track of the individual parts of the hand. The leap motion controller preserves an internal model of the human hand and compares it to the current image of the user's hand using the previous ID. This is a method used by Leap Motion to correct any invalid data read by the sensor due to a hand being partially out of range or a finger hidden due to a bad angle [13]. ADD MORE BACKGROUND

3.2 Task and Approach

The scope of the project consisted of using a programming language to use the input read by the Leap Motion sensor to control a physical robot arm. Our group used C++ in the visual studios environment to control the 6- Degree of freedom robot arm. The program received input from the user through the LM sensor and generated commands to the robot via the microcontroller and Bluetooth.

The X, Y, and Z coordinates of the tips of the index finger and thumb were assigned unique variable names in order for the program to find their location in space. The coordinates of the human fingers were mapped to the tips of each of the two robot prongs to mimic human movements. The distance formula was used to calculate the distance between both points at the tips of the fingers. This distance tells the robot to open or close the gripper.

To move the arm, we used the coordinates of the center of the user's palm to drive each motor according to each axis. When the palm travels in the X direction, motor 1 turns the entire robot. Palm movements in the Y and Z directions actuate motor 2 and 3 to raise and lower the arm.

3.3 Problems with the code and how it was fixed

The greatest obstacle we had to overcome was smoothing out the movement of the robot arm when controlled by LM. Each movement looked choppy due to WHAT? HOW WAS IT FIXED?

WHAT WAS ANOTHER PROBLEM AND HOW WAS IT FIXED?

Another problem we had to overcome was the communication between Leap motion and the microcontroller. LM is only able to send commands as numbers while the microcontroller can only receive commands as characters. Our code converted numbers into characters before sending them to the microcontroller via Bluetooth.

3.4 Results

The results were satisfactory. The team was able to complete its task of manipulating the robot arm using hand gestures over the LM sensor. WHAT ELSE?

3.5 How the code could be improved

SMOOTHING OUT THE ROBOT ARM MOVEMENTS?

4. SolidWorks Design Process

4.1 Feasibility

A feasibility study is used to justify a project [24]. It addresses the relevant factors needed to create a comprehensive cost vs. benefit picture, allowing companies to assess a project before deciding whether or not to go forward with it. For this project, the team decided that the most relevant topics to consider in a feasibility study are: cost, usefulness to the consumer, future applications, benefits, and prospecting market.

Because this project consists of just design work, there was no cost for materials or manufacturing. Therefore, finances were not an issue to complete this task.

The design would ideally be used to help the user, or product developer test relevant and prospect applications. By having the mechanism manipulate objects locally, the user is able to personally see his work come to life. However, in order for this device to be helpful, it would have to be compact enough to fit within the user's workspace. Therefore, the team has to design the robotic hand to be as compact as possible, as well as stable enough to withstand rigorous testing.

The benefits of the future applications are perhaps the biggest driving factor for creating this design. A prospecting market could be the medical field. A robotic arm using leap motion technology can be used around the globe by performing quick checkups, dealing with disease, and in surgeries. A real word example where a mechanical design of five fingers could have been implemented was in the research and treatment of Ebola patients. Ebola is a decease that spread around the world because of close contact with patients, nurses, and doctors. The use of a five-finger robotic arm to treat patients could have greatly slowed down the spread of this disease.

Another important application for using a five-fingered robot is to disarm bombs. Using these types of robots in the armed forces could save countless lives when they are operated from a safe distance. Darpa, shown in figure 1, has five fingers in each hand that allows it to manipulate all types of objects. They also facilitate complex tasks such as handling and picking through wires. Based on these examples, it is clear that Leap Motion has amazing capabilities, which can only grow to unlimited potential. Research in these areas can be very beneficial and lead to abundant income if the technology is bought and implemented by the medial field or armed forces.

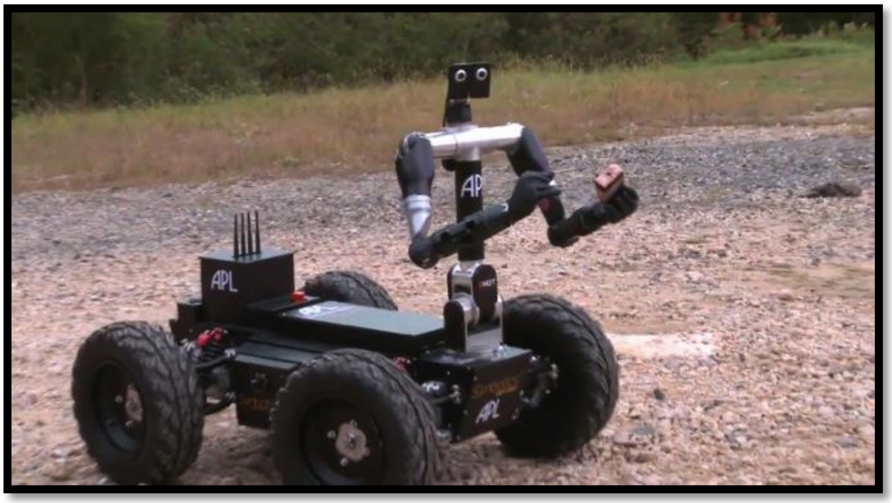


Figure 4.1 Darpa, the Bomb Disarming Robot

The ideal five-finger, six degree-of-freedom robotic would be mostly used by Leap Motion developers and other researchers in the fields mentioned above. The robot is designed to fit in most workspaces and can be used in a wide variety of applications. For example, if a military company purchases this design, it would be able to use it in its lab to practice disarming bombs. Once they tweak the robot according to their needs, the company can use a similar mechanism out in their field.

4.2 Design Requirements

The desired outcome of this project was to design a six-degree of freedom arm with a five-finger hand that would aid in the development of LM. The goal is also to be able to fully utilize all of LM’s features including tracking 5 fingers at a time. In order for the robot to move with the most freedom, the ideal design would have to be able to move with all six degrees of freedom in space. A list of specific requirements are listed below.

Function and Performance

Safety

☐ The mounting system should not tip over

☐ There should be no room in the slides for fingers or clothes to get caught

Operating Characteristics

- Hand should have 5 movable fingers
- Entire system must have at least 6 degrees of freedom

Manufacturability

- Must be cost effective

Assembly

- Must be easy for the buyer to assemble
- Must include manual with easy to instructions
- Must be able to fit in a compact box when disassembled
- Total weigh of package not to exceed 900 lbs.

Durability

- Should be able to pick up at least 15 lbs.
- Should function for at least 20 yrs. in an lab environment
- Must be able to withstand rigorous testing

Cost

- Total manufacturing cost must be less than $1500

4.3 Preliminary and Detailed Design

The preliminary design establishes parameters to work with. This work includes detailed sketches and diagrams of the layout of the robotic arm. The robotic arm parameters were tweaked and reestablished to have a more defined and more optimum framework. Drawing out each individual piece was a good way to visualize the actual size of the robotic arm [25].

After the preliminary design portion. The detailed design is established using a computer aided design program. It includes specifications on materials, dimensions, design life, packaging and more [25].

There were different concept designs that were laid out on paper. The arm drawing, figure 4.1, was a serial arm much like the mechanical arm we programmed with in figure 1. Although this would have been a feasible design, the team decided to develop a translating system instead. This type of system, figure 5.1, better represents how leap motion sees human hand movements.

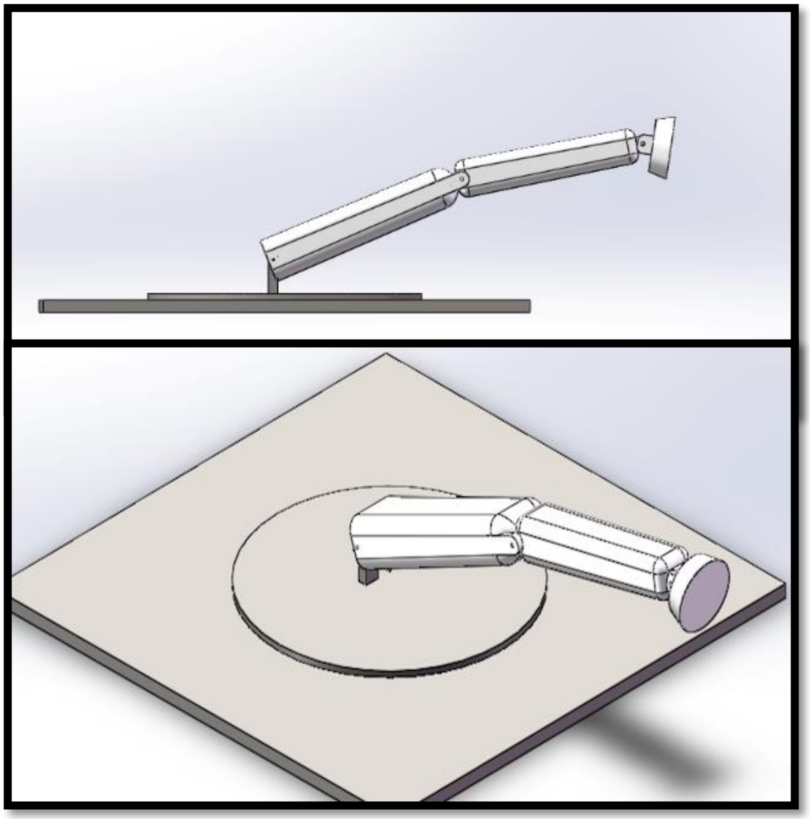


Figure 4.1 Initial Incomplete Robot Arm Design

4.3.1 Why Five Fingers?

Five-fingers were chose for the hand design because they are able to fully utilize the features of LM. A five finger design is much more stable and precise for picking up and gripping objects than the traditional two prong industrial arm. Also, in the case of search and rescue, a human would more readily accept a five finger robot due to the human-like appearance.

4.3.2 Framework

The new design, figure 5.1, is comprised of two motors on either side of the conveyer that uses a rack and pinion concept. Rack and pinion is hassle-free system that eliminates the need for conveyors. It is also deemed to be more precise. Along the conveyer, a central block would house another motor, connected to a rack and pinion concept to move along the x-axis and across the conveyer. The block has a piece which will allow the arm to have yaw motion. This motion will help the motion of the arm over the Leap Motion on a certain pivot point.

The extension into the z-axis is made possible by a ‘telescoping’ system that extends roughly one meter. This too uses rack and pinion to extend. The extension is comprised of six pieces, each has just enough space for one small motor. Connected to the end is a cap that allows the hand to connect, just like a wrist. The wrist area is connected to a palm that is designed to have two motors inside to move the wrist in a tilting motion, along with a roll motion. The design of the hand was tricky. At first we thought the best way to move the fingers was through motors on each joint. However, after doing further research it was determined that the fingers should use a tendon like system using a series of pulleys, shown in figure 5.12. This tendon like system allows for precise movements and enough strength to hold all different sized objects.

5. Final Model Specs and Images

5.1 Frame

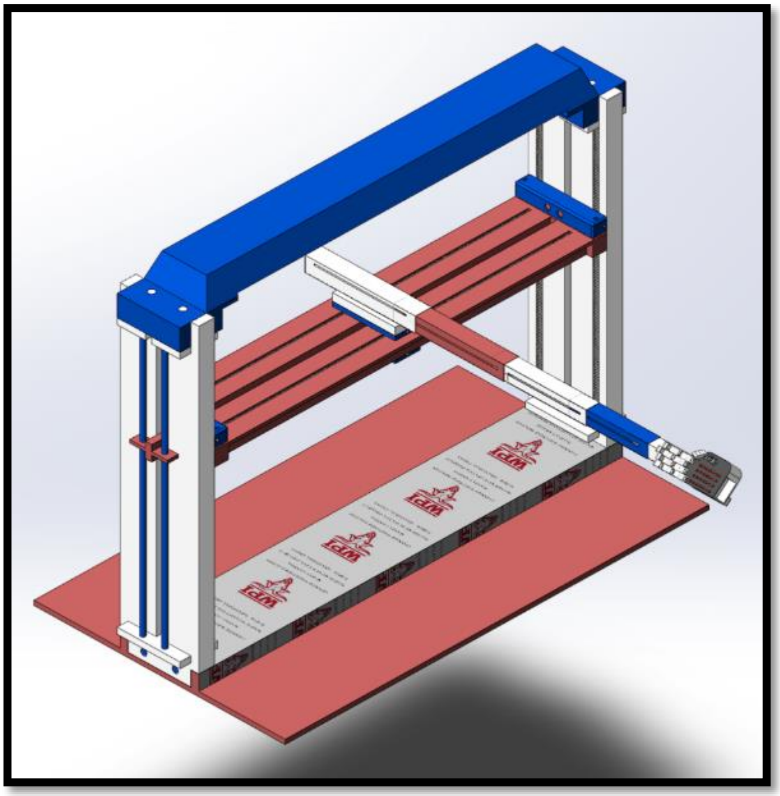


Figure 5.1 Final Design

Total Dimensions (m):
1.66 L x 1.13 W x 114 H
Reach: 1.18 meters
Vertical: 1.07 meters
Horizontal: 1.16 meters

5.2 Rack and Pinion

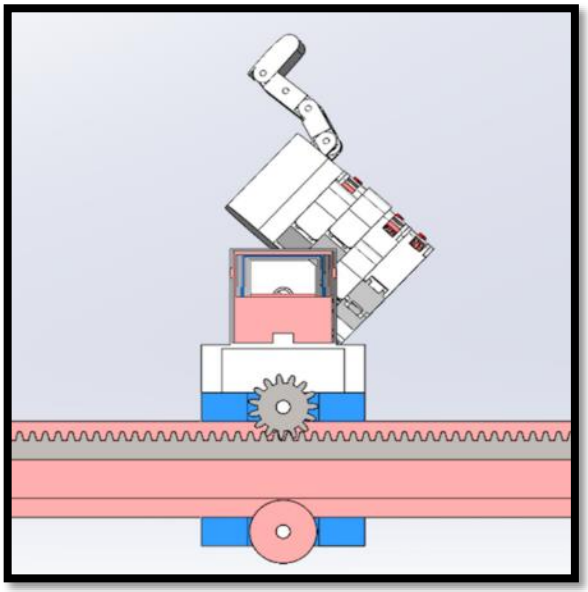


Figure 5.2 Vertical Movement

A motor moves the arm mount, along the conveyer and the slide in the Y-direction.

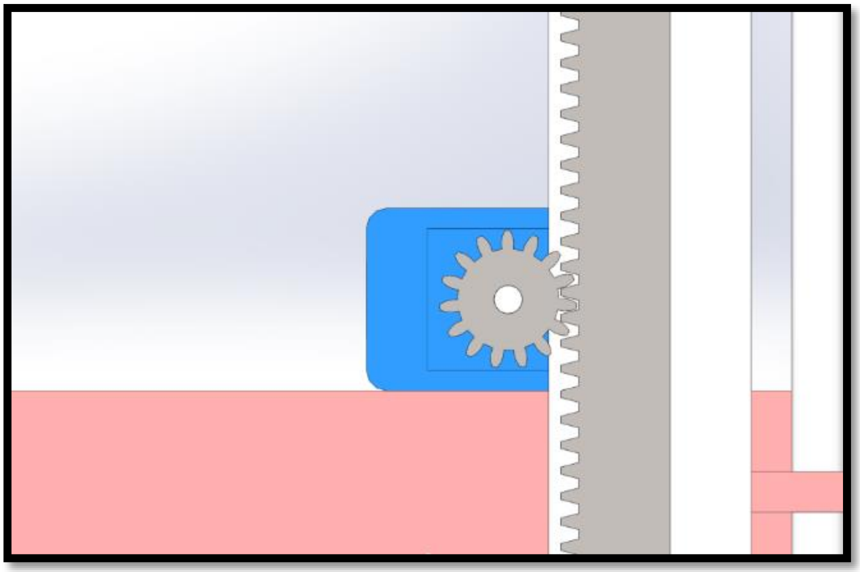


Figure 5.3 Rack and Pinion along the Y-direction

5.3 Slides

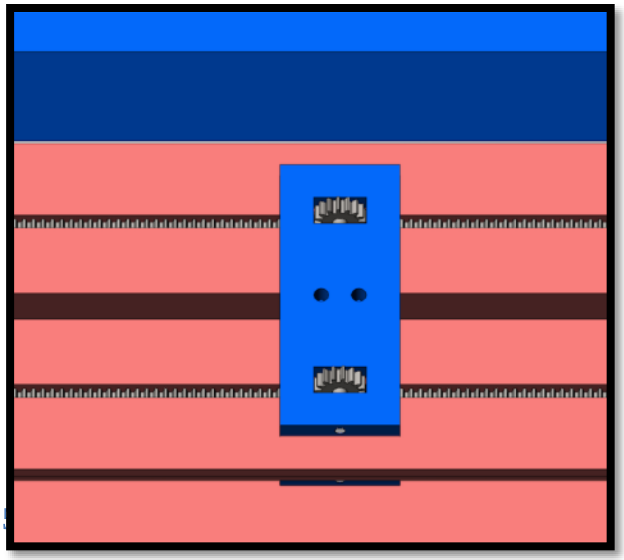


Figure 5.4 Slide Top View

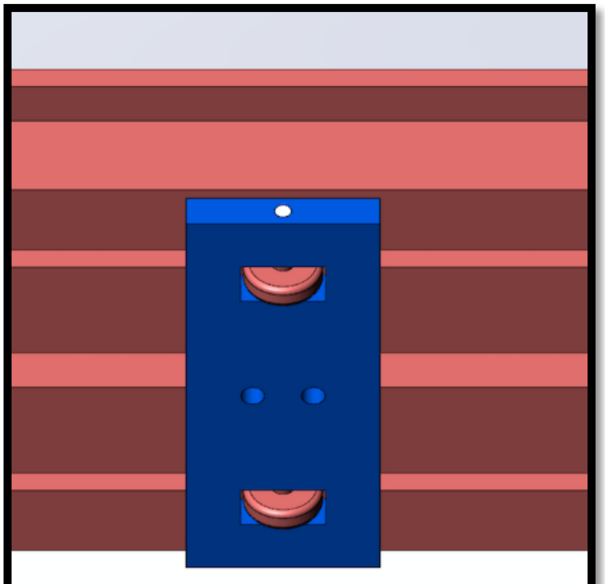


Figure 5.5 Slide bottom View

The rack and the wheels are each fastened by a pin

5.4 Base and Side Walls

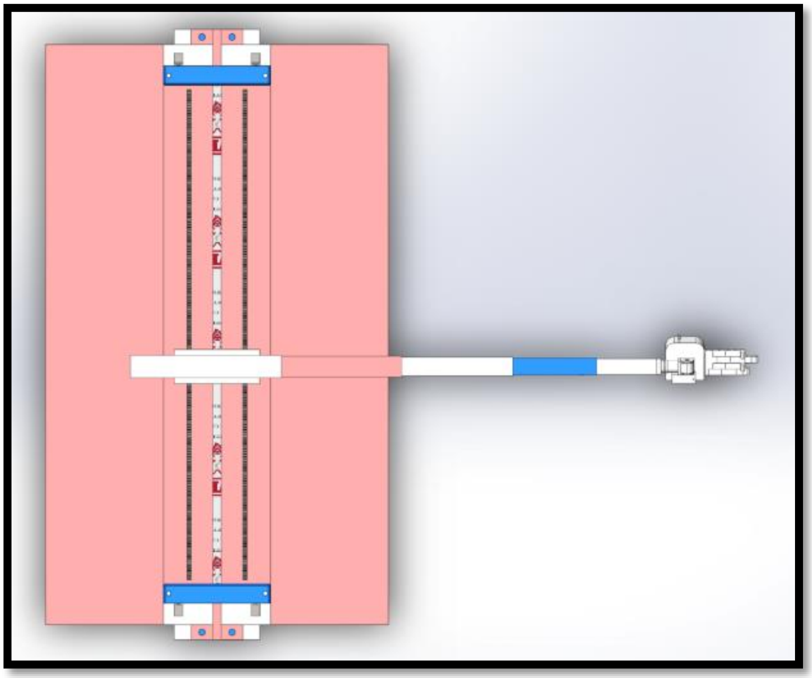
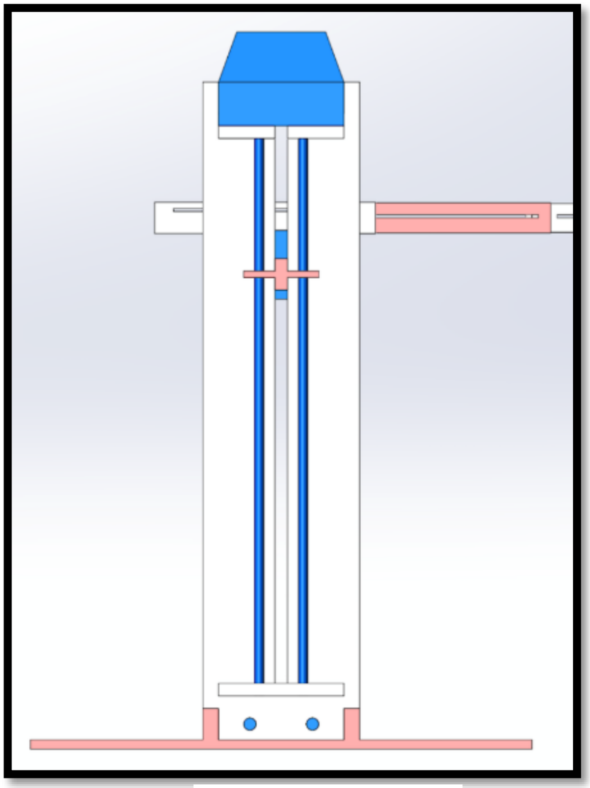


Figure 5.6 Bottom-up View



Designed with two lanes for rack design

Outside struts offer guidance and support

Figure 5.7 Side View

5.5 Hand

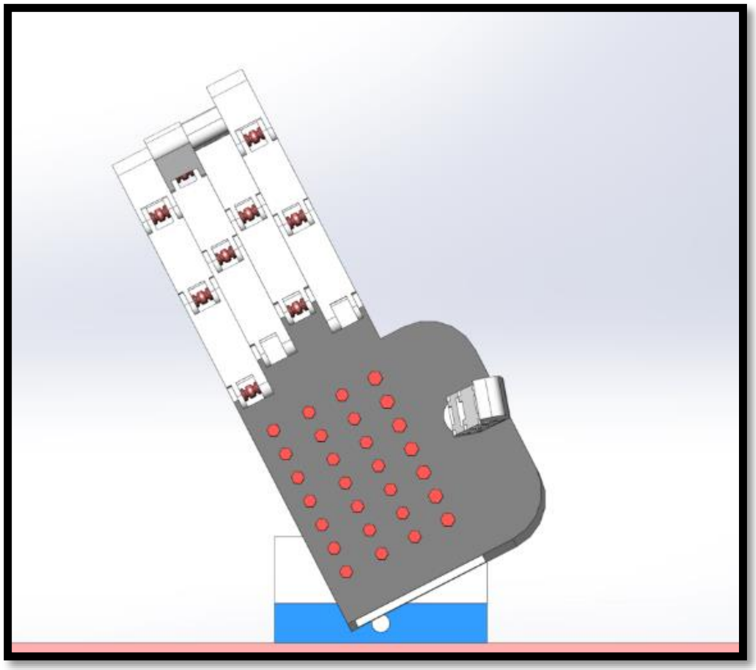


Figure 5.8 Palm View

Parts:
16 total phalanges parts (Proximal, Intermediate, and Distal Phalanges

27 total pins throughout hand for pulley system

Figure 5.9 Motor Slots inside the Hand

A pulley system is intended to be implemented within the hand to actuate the fingers.

There will be five motors to control each fingers

5.6 Fingers and Wrist

Two Degrees of Freedom

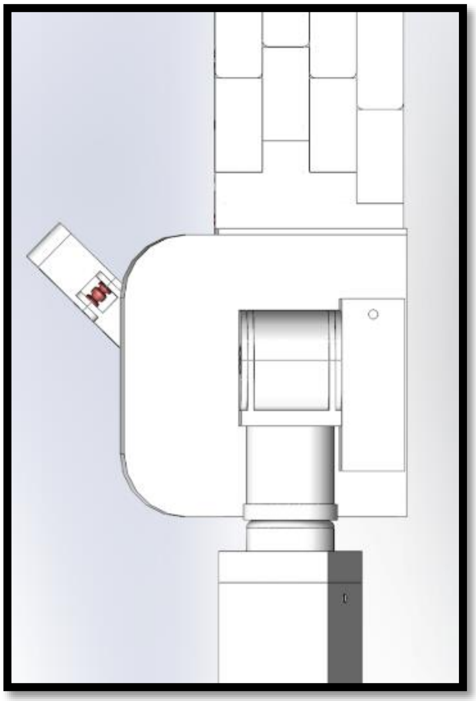


Figure 5.10 Top Hand View

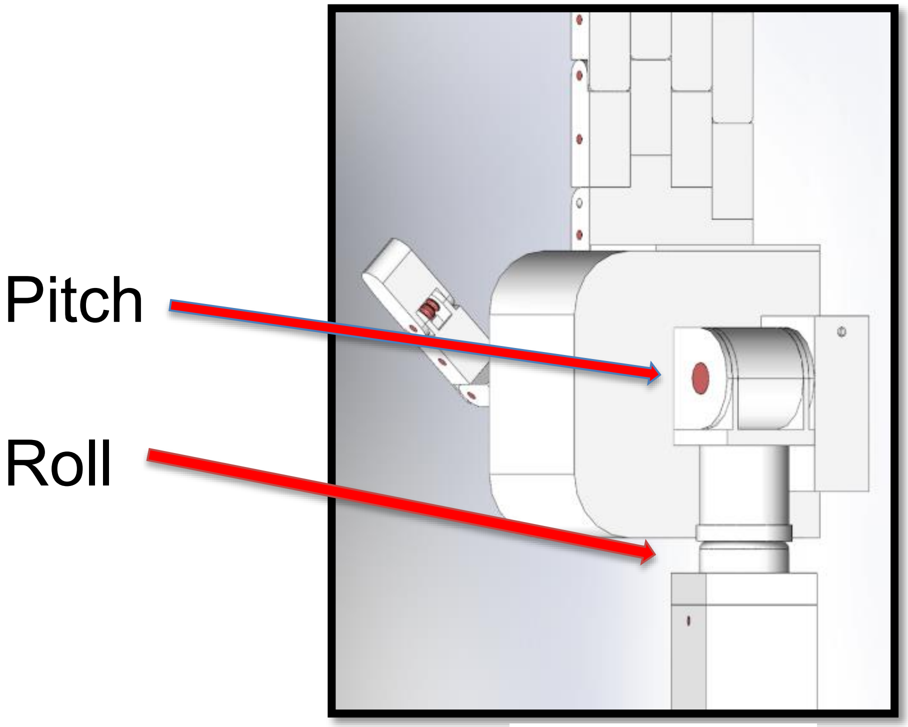


Figure 5.11 Pitch and Roll

5.6 Static Grip Analysis

It is important to find the grip force of a manipulator when designing robots that pick up objects. This would allow the creator to know whether his design is appropriate for the application. For this project, we began the analysis by finding the static equilibrium equations, in figure 5.13, to find the unknown forces. The equations are then written in matrix form as shown in figure 5.15. The final steps would be calculated in future work. Figure 5. 13, shows the final design of this project holding a cylinder with two fingers. Figure 5.14 shows a hand sketched free body diagram of each link of a finger.

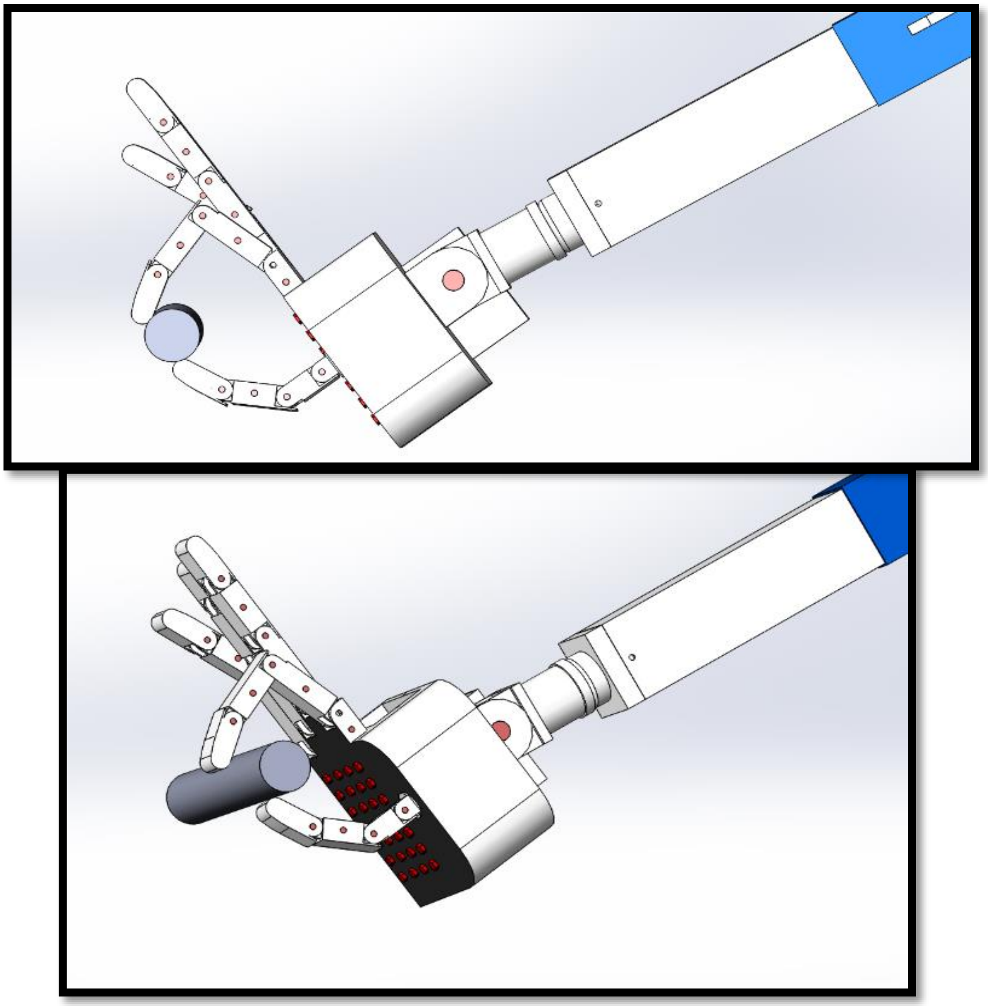
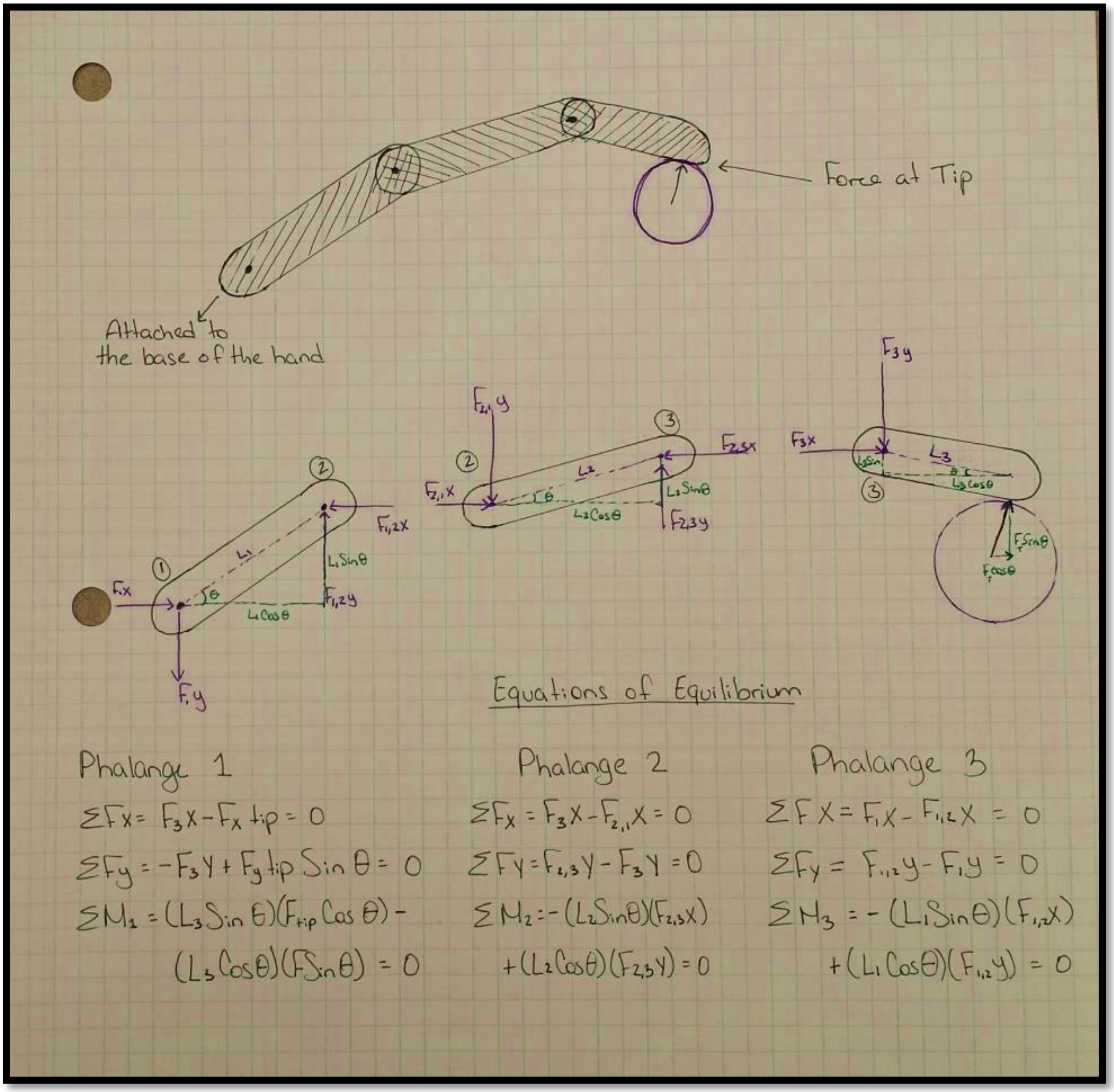


Figure 5.13 Robot Hand Gripping a Cylinder



Equations of Equilibrium

Phalange 1

$$\Sigma F x = F_3 X - F_x \, tip = 0$$

$$\Sigma F_y = -F_3 Y + F_y \, tip \sin\theta = 0$$

$$\Sigma M_1 = (L_3 \sin\theta)(F_{tip}\cos\theta) - (L_3\cos\theta)(F\sin\theta) = 0$$

Phalange 2

$$\Sigma F_x = F_3 X - F_{2,1} X = 0$$

$$\Sigma F Y = F_{2,3} Y - F_3 Y = 0$$

$$\Sigma M_2 = -(L_2 \sin\theta)(F_{2,3} X) + (L_2 \cos\theta)(F_{2,3} Y) = 0$$

Phalange 3

$$\Sigma F X = F_1 X - F_{1,2} X = 0$$

$$\Sigma F_y = F_{1,2} y - F_1 y = 0$$

$$\Sigma M_3 = -(L_1 \sin\theta)(F_{1,2} X) + (L_1 \cos\theta)(F_{1,2} y) = 0$$

Figure 5.14 Hand Sketch of forces on each link

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} F_3 X \\ F_3 Y \\ F_2 X \\ F_2 Y \\ F_1 X \\ F_1 Y \end{bmatrix} = \begin{bmatrix} F_X tip \\ F_y tip \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figure 5.15 Matrix Representation of Equations

6. Cost Analysis of the Mechanical Design

6.1 Frame

Because the frame is about the size of a desk, the lightest possible material was chosen. After some research, magnesium alloy seemed to be the best fit because it is light, strong, and low cost. Magnesium alloy is the lightest industrial metal in the world, about one-third lighter than aluminum. It is also shock absorbing and dampens vibrations [26].

Magnesium is also one of the most common metals used for sheet metal fabrication [27]. The frame and the telescoping arm parts of the robot could be manufactured easiest through sheet metal fabrication because of its shape. The tables below are a rough estimate of the cost of the material needed per robot. The prices were estimated based on the size of each part, then compared to the cost of the closest sheet metal size sold online [28].

	Cost per sheet Size (in)	Sheets used per part	Surface Area (m2)	Weight (lb)
Base (Magnesium)	180$ -.032x36x48 45$-0.032x12x48	2	2.42	120
Sides (Magnesium)	180$- 1/4x12x48	2	0.87	41x2=82
Top (Magnesium)	60$- 1/4x6x48 60$-1/4x6x12	2	0.82	70
Vertical Slide (Delrin)	410$- 1.75x12x48	1	1	27
Total	935$	7	5.11	299

	Cost per sheet Size (in)	Sheets per part	Surface Area (in2)	Weight (lb)
Telescoping Slide 1	50$ 0.04x 12x 24	1	280	0.9
Slide 2	50$ 0.04x 12x 24	1	215	0.3
Slide 3	50$ 0.04x 12x 24	1	178	0.24
Slide 4	50$ 0.04x 12x 24	1	138	0.2
Slide 5	50$ 0.04x 6x 12	1	107	0.16
Total	250$	5	918	1.8

6.2 Hand

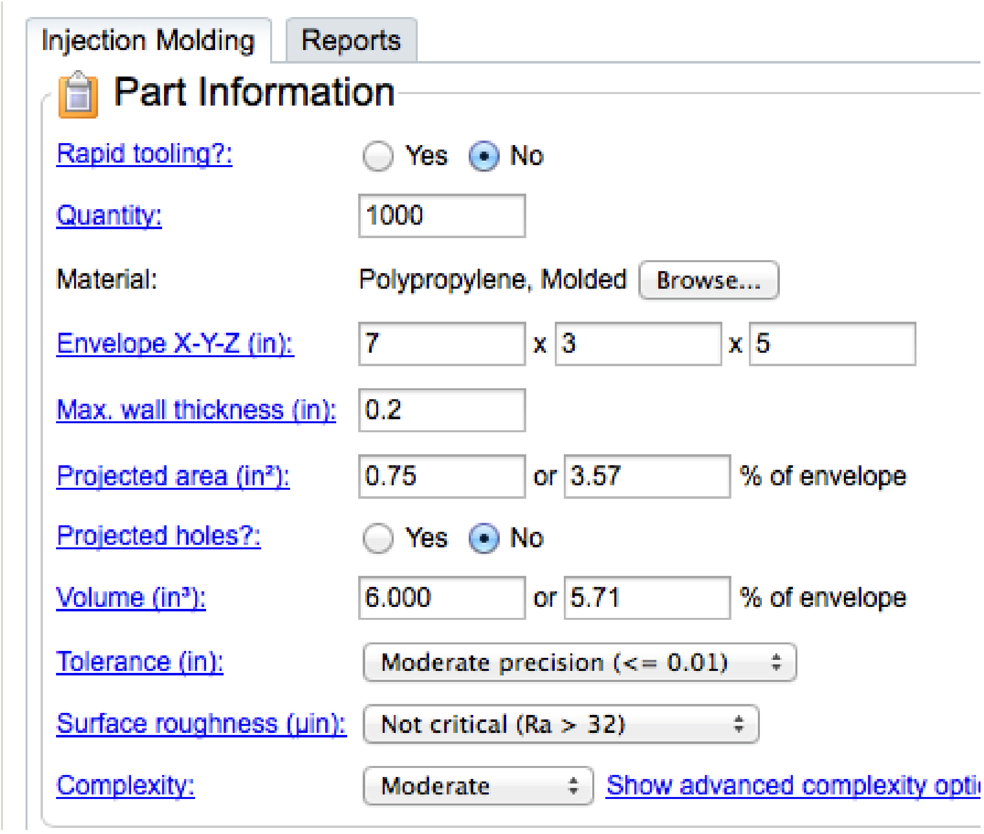
The hand of the robot has to be manufactured with a more malleable material than sheet metal. The palm and fingers have unique shapes and angles that require higher precision manufacturing, such as injection molding. Although injection molding has a higher initial setup cost, it lowers dramatically when used in mass production [29].

While looking through the most popular materials used for injection molding, it was clear that Polypropylene would be the best material. Most manufacturers chose polyethylene as the polymer of choice for the plastic molding process because of its availability, price, ease of use, and strength properties. Usually 80% - 90% of all polymers used in the plastic molding industry are polyethylene compounds [28].

Polypropylene Specs [30]:

- Excellent resistance to concentrated acids, alcohols, bases and mineral oils.
- High resistance to heat, with a melting point of 170 °C.
- Tensile Strength: 4,500 psi
- Estimated cost of 1 USD per pound
- Density: 0.0339 lb/in³

In order to estimate a cost of manufacturing for the hand, we used an online cost calculator from [31]. We filled in the information according to approximate size of the hand and its parts, material, and estimated cost of labor.



Part Information

Rapid tooling?: ○ Yes ● No

Quantity: 1000

Material: Polypropylene, Molded

Envelope X-Y-Z (in): 7 x 3 x 5

Max. wall thickness (in): 0.2

Projected area (in²): 0.75 or 3.57 % of envelope

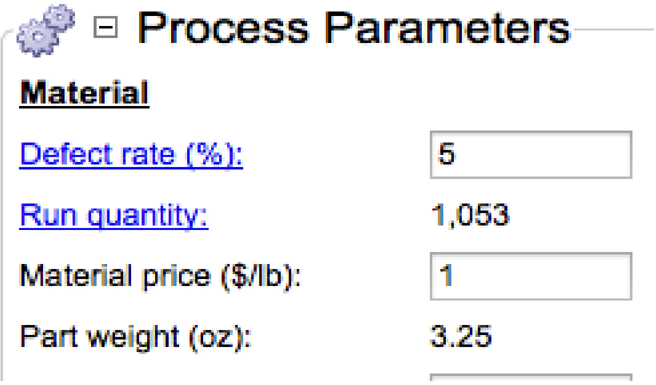
Projected holes?: ○ Yes ● No

Volume (in³): 6.000 or 5.71 % of envelope

Tolerance (in): Moderate precision (<= 0.01)

Surface roughness (μin): Not critical (Ra > 32)

Complexity: Moderate



Process Parameters

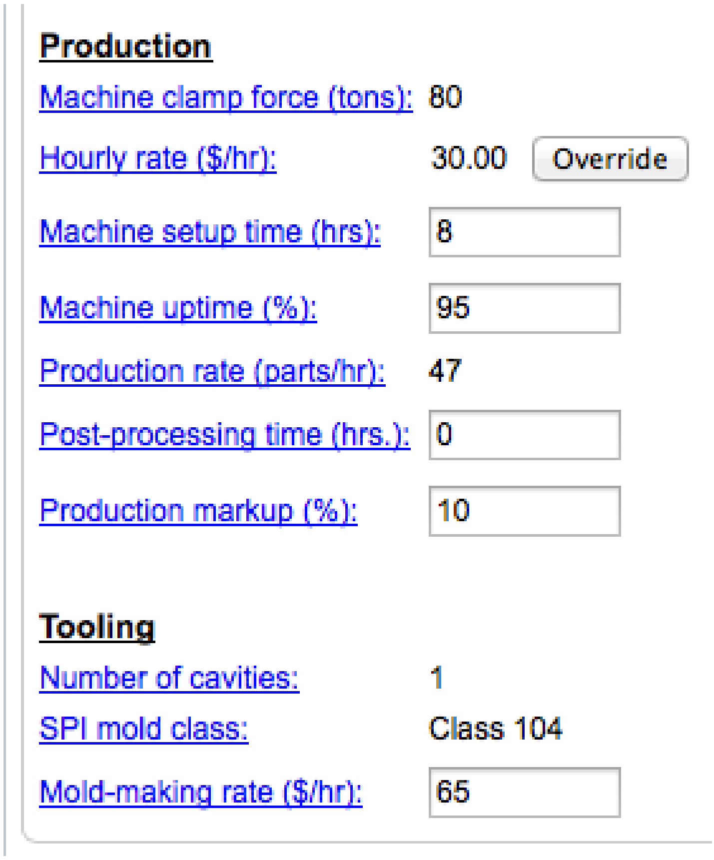
Material

Defect rate (%): 5

Run quantity: 1,053

Material price ($/lb): 1

Part weight (oz): 3.25



Production

Machine clamp force (tons): 80

Hourly rate ($/hr): 30.00

Machine setup time (hrs): 8

Machine uptime (%): 95

Production rate (parts/hr): 47

Post-processing time (hrs.): 0

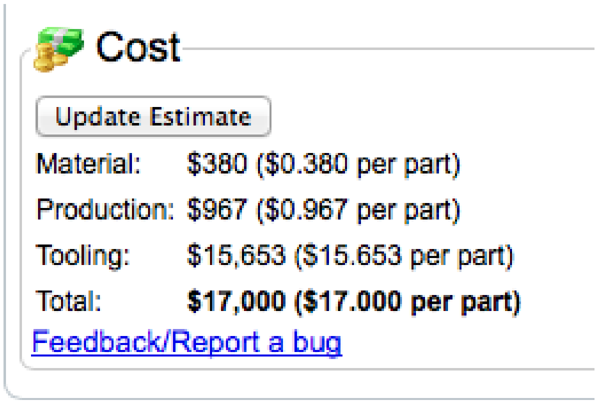
Production markup (%): 10

Tooling

Number of cavities: 1

SPI mold class: Class 104

Mold-making rate ($/hr): 65



Cost

Material: $380 ($0.380 per part)

Production: $967 ($0.967 per part)

Tooling: $15,653 ($15.653 per part)

Total: **$17,000 ($17.000 per part)**

Feedback/Report a bug

According to the cost calculator, each part that composes the hand would cost about 17$. This price can change depending on the size of the part, labor costs, or number of cavities per envelope.

7. Results

The real time control of the robotic arm was a success along with the design of the six-degree of freedom robotic arm. There were a few initial concepts that were designed but were set aside. These concepts included a serial design much like the one physically build for the programming of the Leap Motion. The serial design was too similar and was not under the design parameters that our sponsor wanted. Our sponsor asked for a design that would allow for specific and precise movements along the x-axis, y-axis, and z-axis. To facilitate these needs the model was designed to have a one meter by one meter by one meter in able to have a reach to gather items from any location within the reachable area and move it to a new location.

The most prominent piece of the design is the telescopic arm. This design was decided upon because of the reach that it could have and the compactness that it obtains. The design was originally supplied with a pulley system that would retract the arm however, a rack and pinion design was necessary and much more useful. The arm pieces are interlocked and drive on the rack and pinion with a small motor. With more time, mechanical testing, and further development the robotic arm could be reworked and developed. The robotic arm is designed towards the development of the Leap Motion technology. The Leap Motion technology will develop into a great tool that can be used in the mechanical, medical, and military fields of work.

Working in China, side by side with students of a different culture was enriching but hard for many reasons. One of the biggest and more obvious reasons is the language barrier and the day to day life culture. It is powerful to learn and be exposed to such a different culture. It allows one to build and grow as a person. International relations are grown and techniques of cultural exposure can be used in day to day life situations. Growth through adapting to this culture was very important. Knowing and learning about one another allowed for the smooth transition between cultures and the proper teamwork to have project development.

References

[1] Howell, E., 2014, "Mars Curiosity: Facts and Information," from http://www.space.com/17963-mars-curiosity.html

[2] Stonington, J., 2010, "World's Most Advanced Robots," from http://images.businessweek.com/slideshows/20101101/world-s-most-advanced-robots#slide7

[3] Gaudin, S., 2013, "Machine gun-toting robots may soon back up U.S. soldiers," from http://www.computerworld.com/article/2486011/emerging-technology/machine-gun-toting-robots-may-soon-back-up-u-s--soldiers.html

[4] n.d, About Us, from http://www.northropgrumman.com/AboutUs/Pages/default.aspx

[5] Bellis, M., n.d, "The Definition of a Robot," from http://inventors.about.com/od/roboticsrobots/a/RobotDefinition.htm

[6] Stamp, J., 2013, "A Brief History of Robot Birds," from http://www.smithsonianmag.com/arts-culture/a-brief-history-of-robot-birds-77235415/?no-ist

[7] Unpublished, A., n.a., "Atlas - The Agile Anthropomorphic Robot," from http://www.bostondynamics.com/robot_Atlas.html

[8] Unpublished, A., 2014, "Six Degrees of Freedom," from http://en.wikipedia.org/wiki/Six_degrees_of_freedom

[9] Unpublished, A., 2014, "Serial Manipulator," from http://en.wikipedia.org/wiki/Serial_manipulator

[10] Unpublished, A., 2014, "Types of robot sensors," from http://www.robotplatform.com/knowledge/sensors/types_of_robot_sensors.html

[11] Richardson, N., 2013, "One giant leap for mankind," from http://www.inc.com/30under30/nicole-marie-richardson/leap-motion-david-holz-michael-buckwald-2013.html

[12] Colgan, A., 2014, "Leap Motion," https://www.leapmotion.com/product

[13] Colgan, A., 2014, “Finger,” from https://developer.leapmotion.com/documentation/java/api/Leap.Finger.html

[14] Colgan, A., 2014 “Colgan, A., 2014, “How does the Leap Motion Controller Work?” from http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/

[15] Unpublished, A., 2013, "Interaction Zones and the Leap Motion," from http://www.benfarrell.com/2013/08/07/interaction-zones-and-the-leap-motion/

[16] Colgan, A., 2014, "Leap Motion for Virtual Reality," from https://www.leapmotion.com/product/vr

[17] Unpublished, A., 2015, "Dagu 6 Degrees of Freedom Robotic Arm (Assembled)," from http://www.robotshop.com/en/dagu-6-degrees-of-freedom-robotic-arm.html

[18] Thompson, D., 2001, "Bionics of the Hand, from "http://moon.ouhsc.edu/dthompso/namics/hand.htm

[19] Taylor, T., n.d., "Hand and Wrist", http://www.innerbody.com/image/skel13.html

[20] n.d., "Hand Anatomy,"http://www.eorthopod.com/hand-anatomy/topic/157

[21] Miller, S., 2014, "What is the difference between flexion and extension?" From http://www.wisegeek.com/what-is-the-difference-between-flexion-and-extension.htm

[22] Unpublished, A., 2015, "Interphalangeal articulations of hand," from http://en.wikipedia.org/wiki/Interphalangeal_articulations_of_hand

[23] Canneyt, M., 2013, "Programming with Leap Motion," from http://www.freepascal.org/~michael/articles/leap1/leap1.pdf

[24] Pergl, R., n.d., "Feasibility Study Inputs based on Requirements Engineering," from http://ceur-ws.org/Vol-601/EOMAS10_paper11.pdf

[25] Unpublished, A., 2015, "Engineering Design Process," from http://en.wikipedia.org/wiki/Engineering_design_process

Appendix A

```
#include "stdafx.h"
#include "Interface.h"
#include "leap.h"
#include <iostream>
#include <string.h>

int _tmain(int argc, _TCHAR* argv[])
{
    Interface IF;
    char ch;
    UCHAR
writeBuff[34]={'#','1','P','1','5','0','0','#','2','P','1','5','0','0','#','3','P','1','5','0','0','#','6','P','1','5','0','0','T'
,'4','0','0',0x0D,0x0A};
    IF.Open(L"COM9");

    Leap::Controller controller;
    Leap::Frame frame;
    ch=getchar();
        if (ch=='s'){
   frame = controller.frame();
    Leap::HandList hands=frame.hands();
    Leap::Hand hand0=hands[0];
    float xPos=hand0.palmPosition().x;
    float yPos=hand0.palmPosition().y;
    float zPos=hand0.palmPosition().z;
    std::cout<< "hands: " << frame.hands().count()
     << ",fingers: " << frame.fingers().count()
          <<" xPos:"<<xPos
          <<" yPos:"<<yPos
            <<" zPos:"<<zPos<< std::endl;

    while(1)
    {
        frame = controller.frame();
```

```
    hands=frame.hands();
hand0=hands[0];

        //1 号电机
        float currentx=hand0.palmPosition().x;
        float distx=currentx-xPos;
        int   P1=1500-(int)distx*5;
        if (P1<500)
          P1=500;
        if (P1>2500)
          P1=2500;
        int   a1=P1%10;
        int   b1=(P1/10)%10;
        int   c1=(P1/100)%10;
        int   d1=P1/1000;
        char  A1=a1+'0';
        char  B1=b1+'0';
        char  C1=c1+'0';
        char  D1=d1+'0';
        printf("#1P%c",D1);
        printf("%c",C1);
        printf("%c",B1);
        printf("%c     ",A1);
        writeBuff[6]=A1;
        writeBuff[5]=B1;
        writeBuff[4]=C1;
        writeBuff[3]=D1;


        //2 号电机
        float currentz=hand0.palmPosition().z;
        float distz=currentz-zPos;
        int   P2=1500+(int)distz*5;
        if (P2<500)
          P2=500;
        if (P2>2500)
          P2=2500;
        int   a2=P2%10;
        int   b2=(P2/10)%10;
        int   c2=(P2/100)%10;
        int   d2=P2/1000;
        char  A2=a2+'0';
        char  B2=b2+'0';
        char  C2=c2+'0';
        char  D2=d2+'0';
```

```
printf("#2P%c",D2);
printf("%c",C2);
printf("%c",B2);
printf("%c    ",A2);
writeBuff[13]=A2;
writeBuff[12]=B2;
writeBuff[11]=C2;
writeBuff[10]=D2;


//3 号电机
float currenty=hand0.palmPosition().y;
float disty=currenty-yPos;
int   P3=1500-(int)disty*10;
if (P3<500)
   P3=500;
if (P3>2500)
   P3=2500;
int   a3=P3%10;
int   b3=(P3/10)%10;
int   c3=(P3/100)%10;
int   d3=P3/1000;
char  A3=a3+'0';
char  B3=b3+'0';
char  C3=c3+'0';
char  D3=d3+'0';
printf("#3P%c",D3);
printf("%c",C3);
printf("%c",B3);
printf("%c    ",A3);
writeBuff[20]=A3;
writeBuff[19]=B3;
writeBuff[18]=C3;
writeBuff[17]=D3;

//4 号电机
Leap::Finger firstFingerInList = frame.fingers()[0];
Leap::Finger secondFingerInList = frame.fingers()[1];

float x0 = firstFingerInList.tipPosition().x;
float y0 = firstFingerInList.tipPosition().y;
float z0 = firstFingerInList.tipPosition().z;

float x1 = secondFingerInList.tipPosition().x;
float y1 = secondFingerInList.tipPosition().y;
```

```
                float z1 = secondFingerInList.tipPosition().z;

                float distance=sqrt((x0-x1)*(x0-x1)+(y0-y1)*(y0-y1)+(z0-z1)*(z0-z1));

                //PlanA
                int   P4=1800-(int)distance*5.5;
                if (P4<500)
                   P4=500;
                if (P4>1800)
                   P4=1800;

                //PlanB
                /*int P4;
                if (distance<30)
                        P4=1600;//闭合
                else
                        P4=1200;*/ //张开

                int   a4=P4%10;
                int   b4=(P4/10)%10;
                int   c4=(P4/100)%10;
                int   d4=P4/1000;
                char  A4=a4+'0';
                char  B4=b4+'0';
                char  C4=c4+'0';
                char  D4=d4+'0';
                printf("#4P%c",D4);
                printf("%c",C4);
                printf("%c",B4);
                printf("%c\n",A4);
                writeBuff[27]=A4;
                writeBuff[26]=B4;
                writeBuff[25]=C4;
                writeBuff[24]=D4;

                //send command
                IF.SendCommand(writeBuff);

                Sleep(50);

        }
 }
        Sleep(100);
        return 0;
}
```

```
#include "stdafx.h"
#include "Interface.h"
#include <iostream>
#include <string.h>
#include "leap.h"

//-------------------------------------------------------------------------------------
Interface::Interface(void)
{
	m_bComRunState=false;
}
//-------------------------------------------------------------------------------------
Interface::~Interface(void)
{
	Close();
}
//-------------------------------------------------------------------------------------
bool Interface::Open(wchar_t *ComNo)
{
	m_hCom=CreateFile(ComNo,//COM 口
		GENERIC_READ|GENERIC_WRITE , //允许读和写
		0, //独占方式
		NULL,
		OPEN_EXISTING, //打开而不是创建
		0, //同步方式
		NULL);

	if(m_hCom==(HANDLE)-1)
	{
		printf("打开串口错误\n");
		return false;
	}
	m_bComRunState=true;
	return true;
}

//-------------------------------------------------------------------------------------
void Interface::Close()
{
	if (m_hCom)
	{
		CloseHandle(m_hCom);
		m_bComRunState=false;
```

```
	}
}

//-------------------------------------------------------------------------------------
DCB Interface::getComState()
{
	GetCommState(m_hCom,&m_dcb);
	return m_dcb;
}

//-------------------------------------------------------------------------------------
//19200 配合蓝牙波特率
void Interface::setComState()
{
	SetupComm(m_hCom,100,100); //输入缓冲区和输出缓冲区的大小

	COMMTIMEOUTS TimeOuts;
	//设定读超时
	TimeOuts.ReadIntervalTimeout=MAXDWORD;
	TimeOuts.ReadTotalTimeoutMultiplier=0;
	TimeOuts.ReadTotalTimeoutConstant=0;
	//在读一次输入缓冲区的内容后读操作就立即返回,
	//而不管是否读入了要求的字符。

	SetCommTimeouts(m_hCom,&TimeOuts); //设置超时

	GetCommState(m_hCom,&m_dcb);
	m_dcb.BaudRate=9600; //波特率为 9600
	m_dcb.ByteSize=8; //每个字节有 8 位
	m_dcb.Parity=NOPARITY; //无奇偶校验位
	m_dcb.StopBits=ONESTOPBIT; //一个停止位
	SetCommState(m_hCom,&m_dcb);

	PurgeComm(m_hCom,PURGE_TXCLEAR|PURGE_RXCLEAR);  //清除读写缓冲区
}

//-------------------------------------------------------------------------------------
//启动采样功能
void Interface::SendCommand(UCHAR *writeBuff)
{
	BOOL bWriteStat;
```

```
	DWORD bytesWritten;
	bWriteStat=WriteFile(m_hCom,writeBuff,34,&bytesWritten,NULL);
	PurgeComm(m_hCom, PURGE_TXABORT|
		PURGE_RXABORT);
	return;
}


//--------------------------------------------------------------------------------------
bool Interface::ComIsOpen()
{
	return m_bComRunState;
}
```