Design of a Teleoperated Rock Sampling System

Shajan A. Thomas

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Kevin Kochersberger, Chair
Dennis Hong
Christopher Williams

July 11, 2011
Blacksburg, Virginia

Keywords: Robot Arm, Design, Autonomous Systems

Design of a Teleoperated Chunk Sampling System

Shajan A. Thomas

(ABSTRACT)

Telemanipulators allow a user to interact with a potentially dangerous environment remotely. Deploying a robot arm from a UAV would allow an operator to reach farther and quicker than he or she would with a ground robotics system alone. This thesis will discuss the design and fabrication of a compact, light, 3 degree of freedom robot arm using common off the shelf products and machined components that in combination can pick up half pound samples and has a reach of 260 mm.

Also addressed is making the telemanipulator interface easier to use. One of the challenges in using a robot arm with a single camera in a beyond line-of-sight scenario is the difficulty of interacting with the environment because of a loss of depth information. This lack of information can be remedied with additional sensors. Once depth to an object of interest is known, the sampler can automatically pick up objects of interest.

The manipulator arm will be used in conjunction with systems developed by the Unmanned Systems Laboratory at Virginia Tech. This group is developing a unmanned ground vehicle to be carried in the payload pod of a unmanned aerial vehicle. The robot's ultimate objective is to collect shrapnel and bomb material from potentially dangerous environments.

Contents

1 Introduction **1**

1.1 Background 2

1.2 Ground Sampling Robot 2

1.3 Products on the Market 7

1.3.1 Sensors Options 9

1.3.2 Actuation Options 9

1.4 Nomenclature 11

1.5 Thesis Organization 11

2 Literature Review **13**

2.1 Telemanipulation 13

2.2 Computer Assisted Mobile Manipulation 14

3 Mechanical Design **16**

3.1 Requirements for Teleoperated Chunk Sampler 16

3.2 Condensing the Design Space 18

3.3 Motor Performance Requirements 22

3.3.1 Static Torque Requirements Estimation 22

3.3.2 Dynamic Torque Requirements Simulation 23

3.3.3 Motors Selected 25

3.4 Link Members 28

3.5 Shoulder and Elbow Design 28

3.6 End Effector 30
3.7 Arm Base Design 32
3.7.1 Drive Solution 33
3.8 Sample Collection Vessel 35
3.9 Material Costs 35
3.10 Command Modes 36
3.11 Proposed Operations 37
3.12 Design Verification 38

4 Display Concepts **41**
4.1 3D Model of Robot Pose 42
4.2 Infrared Sensor Depth to Objective 43
4.3 Rock Detection Using a Single Camera 44

5 Computer Assisted Sampling **47**
5.1 Supervisory Control 47
5.2 The Kinect™ 48
5.3 Sample Centroid Determination 49
5.4 Robot End Effector Positioning for Grasp 51
5.5 Servo Position Analysis 53
5.6 Centroid Detection Calibration 54
5.6.1 Centroid Detection Test Setup 55
5.6.2 Centroid Detection Test Procedure 57
5.6.3 Centroid Detection Test Results 58
5.7 Arm Movement Precision Testing 60
5.7.1 Arm Movement Test Procedure 63
5.7.2 Arm Movement Test Results 64
5.8 Combined Error 65
5.9 Gripper Test 67

5.9.1 Setup and Procedure 68
5.9.2 Test results 68

6 Conclusion and Improvements **69**
6.1 Summary of Work 69
6.2 Proposed Improvements 70

Bibliography **73**

A Denavit-Hartenberg Representation of the Arm **76**

B Stress Calculations: Arm Member Stresses **78**

C Application of Lewis Equation for Gearing **81**

D End Effector Link Loading **82**

E Power and Command Circuit **85**

F Autodesk Timing Belt Strength Test **86**

G Bill of Materials **87**

H Source Code **90**
H.1 Computer Assisted Mode 90
H.2 Pick Up Rock 96
H.3 Motor Command Interpretting 97
H.4 LabVIEW Joystick Command 102

List of Figures

1.1 Illustration of how the GSR will be deployed from the RMAX. 3
1.2 The Yamaha RMAX retrofitted as a UAV at the Unmanned Systems Lab. . 3
1.3 GSR with pan tilt camera. 4
1.4 See through rendering of the GSR. 5
1.5 Communications and power schematic. 6
1.6 The 1 DOF chunk sampler developed by Matthew Torok at the USL. . . . 7
1.7 Final form of the robot. 8
1.8 Robot arm with components labeled. 11

3.1 Potential low DOF mechanism choices 19
3.2 Angular grippers versus parallel grippers. 21
3.3 Free body diagram for 3 DOF robot arm. 23
3.4 SimMechanics block for robot arm. 25
3.5 Plot of motors considered restricted to torque per mass and price 27
3.6 Shoulder exploded view. 30
3.7 Elbow exploded view. 31
3.8 End effector or gripper closed and opened. 32
3.9 Alternative end effector gripper configuration versus four-bar linkage. . . . 32
3.10 View obstruction reduced with parallelogram mechanism. 33
3.11 Profile view of the end effector. 34

4.1 Labview generated 3D representation of the robot arm. 42

4.2 The 3D representation of the robot arm with the IR sensor data included. . . 43
4.3 Rockfinder demonstration on an image. 45

5.1 Point cloud of a D battery. 49
5.2 Depth map to Centroid 50
5.3 Homogeneous Transformation Schematic 51
5.4 Restricted 2 linkage schematic. 52
5.5 Normalized error over robot motion. 54
5.6 Bolt washer assembly used for Kinect™ calibration. 55
5.7 Optical table hole pattern. 56
5.8 Measurements used to calibrate the output of the centroid detection algorithm in the Z direction. 57
5.9 Estimated maximum gripper error. 58
5.10 Matching the Kinect™ coordinate frame to the table's coordinate frame. . . 59
5.11 Calibration test points on optical table. 59
5.12 Kinect™ centroid plot versus expected values. 61
5.13 Kinect™ centroid X values versus error observed. 62
5.14 Kinect™ centroid Y values versus error observed 62
5.15 End effector grabbing a block 200 mm away from front lip of the sampling tray. 63
5.16 End effector grabbing a block 50 mm away from front lip of the sampling tray. 64
5.17 Set of movements used to test vertical movement precision. 64
5.18 Set of movements used to test horizontal movement precision. 65
5.19 Y placement error at Y=0 mm. 66
5.20 Y placement error at Y=-50 mm. 66

A.1 Comparison of schematic with physical robot. 77

B.1 Link Free Body Diagram 78
B.2 Distance between centers of the links 79

D.1 FBD of end effector resisting forces involved in gripping 83

D.2 FBD of end effector link resisting sample loading 84

E.1 Power and motor command circuit. 85

List of Tables

1.1 Part names used in rest of the document. 12

3.1 Motors picked for final arm. 26
3.2 Potential robot arm material. 28
3.3 Truncated Sales Bill of Materials 35
3.4 Bearing cost estimate if purchased. 36

5.1 Precisions and biases referred to in equations 5.13 and 5.14. 67

A.1 DH parameters for the full robot arm. 76

F.1 Timing belt strength confirmation. 86

G.1 Sales Bill of Material:Base 87
G.2 Sales Bill of Material:Shoulder 88
G.3 Sales Bill of Material:Elbow 88
G.4 Sales Bill of Material:End Effector 89

Chapter 1

Introduction

The Unmanned Systems Lab (USL) at Virginia Tech has been tasked with prototyping a suite of tools that will aid in remote sample collection from a UAV. In this thesis, the design of the chunk sampler in the form of a robot arm or manipulator is discussed. The sampler will be used to collect material from a Radiological Dispersal Device (RDD) blast. The analysis of these samples will aid in determining the safety of an area for rescue workers and civilians.

A device along with software that supports ease of use, was designed and built and will be described in this thesis. The device is a 3 degree of freedom (DOF) robot manipulator that can pick up 250 g (0.55 lbm) samples with a maximum diameter of 44 mm (1.75"). The device fits and collapses onto a sampling tray which fits onto a ground sampling robot (GSR) or unmanned ground vehicle (UGV). The device must also be lightweight at 1.5 kg (3.3 lbm) as it is to be deployed from a rotocraft. As far as ease of use is concerned, the arm is operational under a single click mode of operation.

The final version of the robot arm will have two modes of operation. The first is manual control where each joint of the arm is controllable by the operator. The second is a semi-autonomous mode of operation which allows a user to click a sample on the screen and the robot arm will retrieve it. The ground sampling robot is currently equipped with a single camera which can limit a user's depth perception in an environment. This in turn, makes interacting with objects in an environment challenging. The lack of depth perception becomes a noticeable hurdle with telemanipulation when using a single camera in conjunction with a robot arm in beyond line of site operation [1]. The use of a 3D sensor is proposed

to allow for semi-autonomous operation. The advantage of a semi-autonomous operation is that it allows less training and concentration while collecting samples.

1.1 Background

The systems being developed at the USL will be deployed in response to a nuclear event resulting from a natural disaster or explosion. The collection of information about the disaster site and samples from ground zero will be facilitated by devices designed and prototyped at the USL. These devices revolve around the use of an unmanned air vehicle (UAV) with vertical take off and landing (VTOL) capabilities. This aircraft can map terrain and provide video information back to the ground station. At the lab, a Yamaha RMAX Autonomous helicopter has been outfitted with a radiation detector which can be used to localize the radiation source. In addition to surveying an area, the VTOL UAV can deploy a ground sampling robot to gain a closer look and collect particulate and chunk samples.

The GSR will be deployed from the unmanned VTOL aircraft via a tether and winch system. This system is shown in Figure 1.1. The USL RMAX, has a maximum payload of 28 kg (61 lbm), restricting the size of the sampling device to approximately 2.27 kg (5 lbm). The rest of the weight budget is taken up by other devices such as the ground robot, sensing equipment, on board computers, and localization equipment. The Yamaha RMAX is pictured in Figure 1.2. The GSR is shown in 1.3.

The chunk sampler is a device that allows for the collection of bomb material specifically in the event of the detonation of a Radiological Dispersive Device (RDD), or dirty bomb. An RDD is radioactive material packed around a conventional explosive that can be disruptive to a small area. The chunk sampler is designed to collect samples from the bomb blast area.

1.2 Ground Sampling Robot

The GSR is a lightweight treaded robotic platform that has been designed to be deployed by an unmanned aerial vehicle. The GSR is an unmanned ground vehicle (UGV) with sampling capabilities. It weighs 6 kg with the chunk sampler in place. The lightness is due to its construction from low density materials: carbon fiber tubing, balsa wood, and aluminum.

Figure 1.1: Illustration of how the GSR will be deployed from the RMAX.

Figure 1.2: The Yamaha RMAX retrofitted as a UAV at the Unmanned Systems Lab.

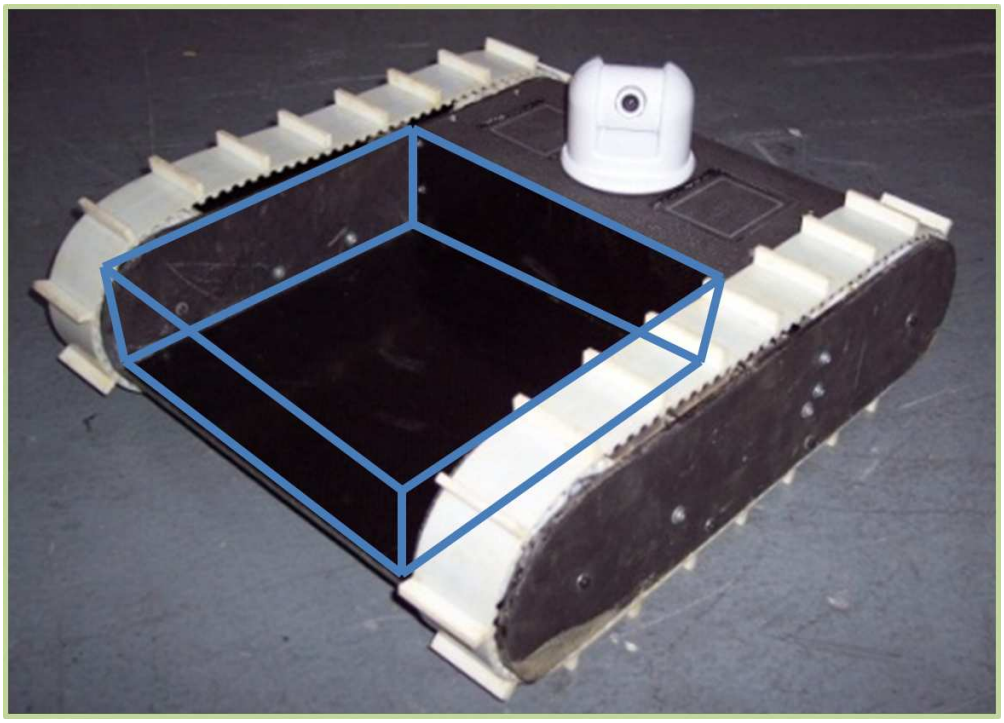


Figure 1.3: GSR with pan tilt camera. The blue box represents the potential volume for a chunk sampling device.

Dimensionally, the robot is 460 mm(18") x 390 mm(15.5") x 140 mm(5.5") and fits into the underside of the Yamaha RMAX.

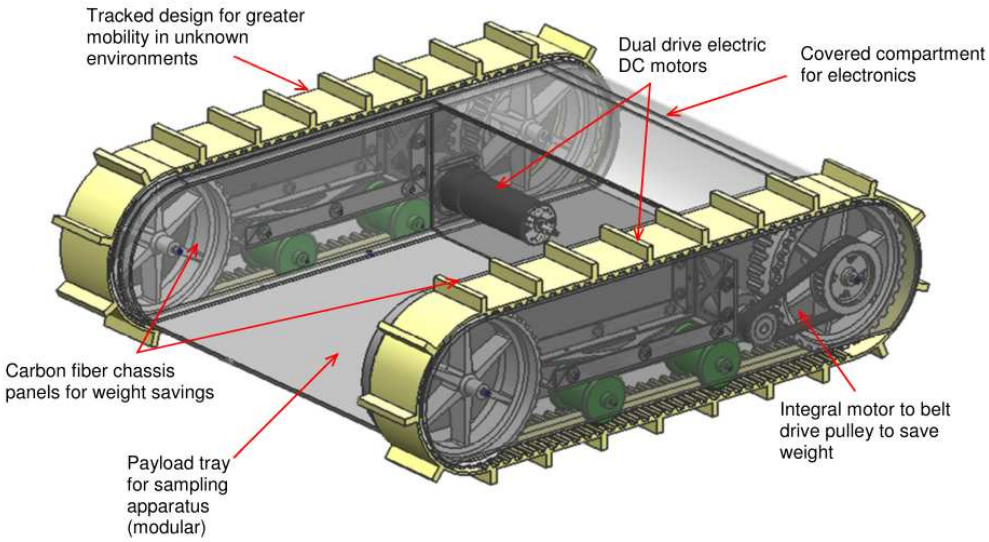


Figure 1.4: See through rendering of the GSR. Key components are labeled.

A schematic of the ground sampling robot is seen in Figure 1.4. The location of the sample tray is illustrated as located in the front end of the robot. The treads of the GSR are driven by two brushed DC motors that give the necessary torque to allow the robot to negotiate uneven terrain. Two sets of rapid prototyped polycarbonate pulleys are used to transfer force from the motors to the treads. The treads used are polyurethane timing belts typically used in a factory settings but repurposed for the ground robot.

The GSR allows a human operator the ability to explore a remote area. The operator interacts with a LabVIEW user interface which interprets the operator's commands to the GSR. To obtain control of the GSR, communications are sent over an Xbee and a WiFi link. An overall schematic of the power distribution and communication can be seen in Figure 1.5. Commands to the robot are sent and received over a Xbee radio link and are interpreted by an Arduino Mega. The Arduino then sends PWM signals to a Sabertooth motor controller which amplifies the signal and drives the treads. A pan-tilt webcam is mounted to the back portion of the robot which provides an eye for the teleoperator on a remote scene. The information collected by the camera is sent back via a standard WiFi^{TM} link, provided by a router mounted within the GSR. This information is then displayed on the user's LabVIEW user interface.

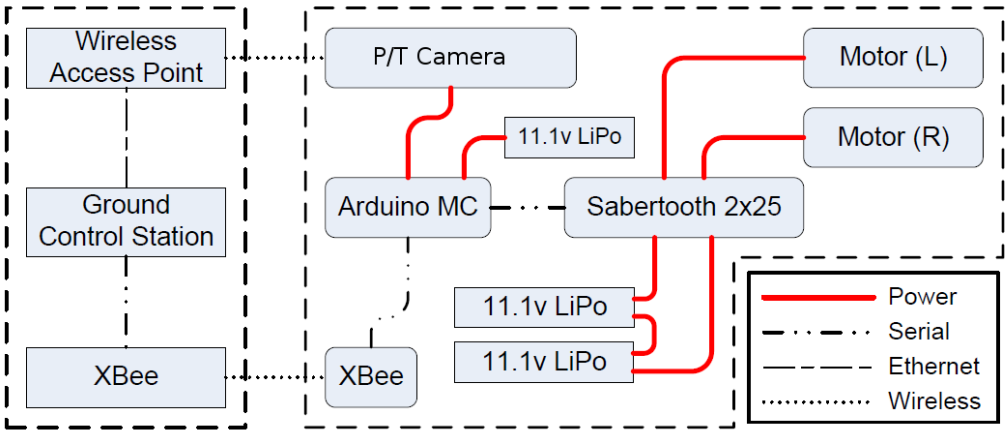


Figure 1.5: Communications and power schematic.

The sampler described in this thesis will be attached to the GSR. The GSR has a modular sampling tray that can accommodate an array of potential samplers that adhere to a custom form factor. Currently, there are two possible sampler ideas that have been developed. The first is a particulate sampler used to sample dust that arises from a nuclear detonation [2]. That particulate sampler is a low wattage and low mass solution for collecting fine grained powders. The device uses stored pneumatic energy to collect the sample material as opposed to a vacuum based system. As a result of the sampling system's low weight requirements, the mass investment into the impellers, a permanent motor, and battery pack of traditional vacuum systems were seen as wasteful. Once a sample is collected, this mass is of no use for the rest of the mission. The effectiveness of this design is currently being investigated by researchers at the Unmanned Systems Lab.

The second type of sampler is a the chunk sampler described in this thesis. The USL has two devices that fit into this category. One is a 1 DOF chunk sampler that uses a bucket claw construction seen in Figure 1.6. The other is the 3 DOF telemanipulator that was developed by the author shown in Figure 1.7. The articulated arm is designed to pick up 0.5 lbf samples and deposit the retrieved items into the SCV. Once the samples are collected, the arm will close over the SCV and prevent the samples from leaving the SCV during transit.

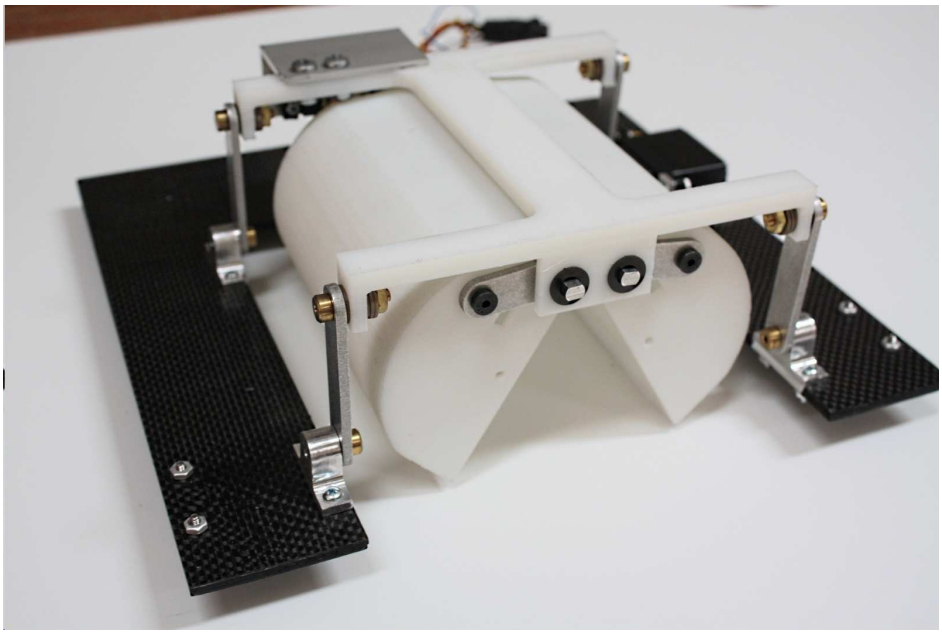


Figure 1.6: The 1 DOF chunk sampler developed by Matthew Torok at the USL.

1.3 Products on the Market

There are many manufacturers of robot arms that are in the market from hobby products, industrial, academic and military products. Most of them do not fit our particular application in being too large, or not compact enough to stow away inside our ground robot pod while the unmanned aerial vehicle is in transit. The following section will take a look at products sold on the market with information gathered from the manufacturers' websites.

On the hobby end of the market, the first robot arm that was considered was the AX-12 Smart Arm produced by CrustCrawler robotics. The smart arm uses 7 Robotis Dynamixel AX-12 robot servos. These robot servos are an improved version of traditional hobby servo motors. First off, these motors can be daisy chained and share the same communication lines. This reduces the amount of wiring necessary. The communication protocol for the AX-12 is serial communication instead of a constant pulse width modulation (PWM) signal that would be needed in the case of hobby servos. The AX-12 will not work for our application as it does not fold compactly into the volume allotted on the Yamaha RMAX.

The Barrett Whole Arm Manipulator (WAM) is so named because the whole robot arm can be used to interact with its environment. It weighs 25 kg, has a payload capacity at 4 kg,

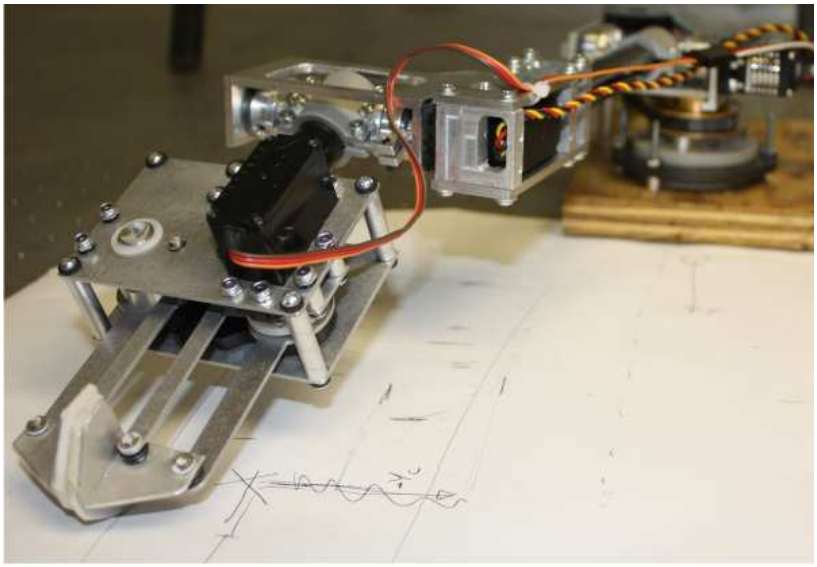


Figure 1.7: Final view of the robot.

and a reach of 1 m. The arm uses a cable transmission to transfer power from the base of the robot arm out to the farther links [3]. The WAM arm also has a set of axially stepped pulleys which aid in mechanical power transmission to the rest of the robot. A cable drive was considered as it would reduce the amount of mass away from the base of the robot arm, potentially increasing its payload capacity but comes at a cost of a longer design cycle.

The closest robot arm found that loosely fit the application was the RE^2 DS1-MA series manipulator arm. This single DOF manipulator has a lifting capacity of 5 lbf, a reach of 7-9 inches, and can supply a grip force of 25 pounds all while maintaining a weight under 3 pounds. This solution would require precise manipulation of the UGV in potentially uncooperative terrain and constrains the location of the sampler system on the sample collection tray.

The sampler designed in this paper satisfies the requirements for our particular application. Ideas from hobby markets and military markets were combined to create the articulated robot arm implemented.

1.3.1 Sensors Options

In order for a robot to be able to interact with their surroundings, it must have the capability to measure the arm's position relative to objects of interest. The set of sensor that can be used in robotic systems can be broken down into two categories: exteroceptor, environmental sensor, or proprioceptors, self sensors [4]. The following is a selection of sensors that can be used to locate a robot or locate an object of interest.

Joint positional sensors and force sensors make up the majority of the proprioceptor category. The German Aerospace Center (DLR) uses strain gages on the joints of their Light Weight Robot (LWR) III arm to determine the torque being exerted at the joint. This in conjunction with custom potentiometers help gives the robot necessary feedback [5]. Strain gages are used again to determine force exerted in their end effector, Hand II. This information is necessary when manipulating delicate objects, as gripping an object too hard may destroy it. Proprioceptors are usually integrated into the body of the robot arm.

Rotational position sensors for robot arms are important for accurate end effector placement. Optical encoders are a type of sensor that is commonly used in robotics. These work by counting the number of stripes marked into a circular pattern that pass by a light sensor. Potentiometers works with a sliding internal contact that changes the resistance according to a shaft displacement.

Environmental sensor or exteroceptors help a robot sense the environment around it. Whole Arm Proximity Sensors (WHAP) make use of capacitance to determine if obstacles are approaching the robot arm in Novak [6]. Novak goes on to describe how these sensors were implemented to prevent a user from colliding into obstacles by only allowing motion away from the obstacles while constraining movement towards the obstacle.

1.3.2 Actuation Options

Actuators are necessary for a robot arm to move and to do anything of use. Reasonable actuators range from hobby servo motors, DC motors, electrically commutated DC motors, ultrasonic motors, to artificial muscles. The following is a set of actuators considered for this application.

Pneumatic Artificial Muscles (PAM) [7] are favorable as a result of their spring-like proper-

ties. This is desirable in cases where impulses were to occur to the robot arm or where there is no force feed back. The air muscles will simply stretch to accommodate any unintended loading and it would save potential stressing of a delicate frame. PAM actuators are very light weight but require a pneumatic system on the GSR to pump air in and out of the muscles. These motors can control joint angles when run in antagonistic pairs, similar to how our human muscles move the joints.

Ultrasonic motors have favorable torque to weight ratios [8]. They are a form of solid state actuation that involves small scale movements. An added benefit is that they also consume a relatively low amount of power than other solutions. They have been used on JPL robot arms due to their favorable characteristics.

The DLR robot arms, LWR I, II, and III, use a set of custom actuators branded as RoboDrive [5]. These DLR designed motors, have lower masses and lower power losses than motors on the market as Hirzinger claims. Due to the low volume of motors produced, the costs are expected to be high.

Robotis offers a line of advanced robot servo called Dynamixels. There are a large selection of Dynamixel servos that satisfy a range of torque requirements. The servo motors have added functionality over hobby servos in that it they can be daisy chained; saving a robot designer the need to have separate wiring harnesses for each motor going back to the control board. Instead, motor 2 plugs into motor 1 and so on. Commands are sent to the motors using serial commands which can be more desirable than maintaining PWM signal to the motor as would be the case with hobby servos.

Volz offers a line of ruggedized servo motors. The servos meet vibration standard MIL-STD 810F and acceleration standard MIL-STD 810G. These servos may be used in place of the Hitec servos specified for the current version of the arm if future ruggedization is required.

As this system is a conceptual prototype, the actuators purchased were inexpensive and yet delivered the necessary torques to pick up an object of interest. Hitec servos HS-7950TH for the shoulder, HS-5685MH for the elbow joint, and Stepper Motor #1209 Stepper Motor from Pololu, and a JRSport standard servo were chosen as the motors in the robot arm. Section 3.3 describes the process through which these motors were arrived at.

1.4 Nomenclature

In the rest of the document, parts of the chunk sampler will be refered to by names shown in Table 1.1. The chunk sampler system consists mainly of an articulated robot arm and was developed at the Unmanned Systems Lab. Therefore, it will be refered to as the USLArm. Figure 1.8 has a picture of the final chunk sampler with different parts of the system. 'A' refers to the base of the robot arm. It is about the base that the entire mechanism can pivot. 'B' is a 1-DOF shoulder joint, or for the purposes of this paper, simply the shoulder joint. 'C' will be refered to as the shoulder linkage or link. 'D' is the elbow joint. 'E' is the elbow link. 'F' is the end effector. 'G' are refered to as the grippers and are the means through which rocks are picked up. 'H' is the sample collection vessel (SCV). The SCV, in combination with the robot arm, forms a container for samples.

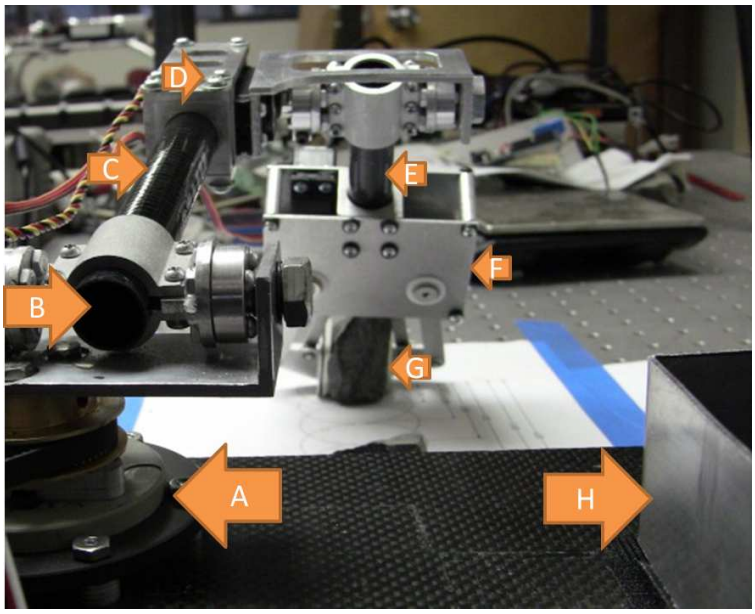


Figure 1.8: Robot arm with components labeled.

1.5 Thesis Organization

The following chapters will contain a literature review related to robot arms. After describing the products and the market and taking a look at what others have done in the field of telemanipulation, the mechanical design process for the manipulator arm will be described.

Table 1.1: Part names used in rest of the document. Key to Figure 1.8.

Identifier	Part Name
A	base
B	shoulder
C	shoulder link
D	elbow
E	elbow link
F	end-effector
G	gripper
H	sample collection vessel (SVC)

From here interface designs will be described and then finally the computer assisted sample collection mode will be shown.

Chapter 2

Literature Review

The following chapter will give an overview of how telemanipulation systems are being used and explore methods in which computers assist in manipulation.

2.1 Telemanipulation

A telemanipulator can be defined as a system that can project a human's actions to dangerous environments [9]. The first telemanipulation systems were developed in the early days of the nuclear age to handle radioactive material from a safe location [10]. A thick wall with viewing ports was used to shield the user from exposure to radiation. Later on these devices were modified into master-slave robots where a human would move a device that was very similar in shape to the arm doing the work on the other end. An operator would move the master system and the slave system would respond with the same movement.

Schenker [11] describes a sample return robot developed at Jet Propulsion Lab (JPL) that was the prototypical Mars Exploration Rover. The SRR-1 uses an all composite manipulator arm to retrieve samples and has a mass of 40 kg. The MarsArmII, has 4 degrees of freedom with a gripping end effector that has a total of a 2.3 m reach. Ultrasonic motors were used in this application as they have a very high torque output to weight ratio [12].

Byler documented early design consideration for a robot arm rock sampling system for extraplanetary exploration [13]. One of the main design considerations described in the article deals with what arm linkage and motor configuration to use on an articulated arm.

The options evaluated are a cable drive method mentioned previously and implemented in the Barret WAM arm. A counter balanced arm was considered as it could reduce the actuator torque requirements for the system but at a potential cost of added weight. A direct-drive configuration was also considered where the motor output shafts are located coaxially on the joint axis. Byler recommends a cable drive system with warnings concerned with the robustness of the cables and the difficulty of assemblage of the system. A cable drive system was avoided as it would require added design time.

A widely used mobile manipulator currently used in Operation Iraqi Freedom is the iRobot® Packbot with the manipulator attachment; the robot has a particular use in explosive ordnance detonation [14]. In another application, a modified end effector was equipped on the Packbot that specialized in opening doors. The rubber padded end effector's design made use of a compliant joint that allowed the robot to open a door in an easier way [15], this reduced the number of manipulator joints that needed to be actuated and shows how a mechanism can be tailored to a particular application. A soft foam was used on the grippers of the end effector with the USLArm to improve the gripping capabilities of the robot.

2.2 Computer Assisted Mobile Manipulation

Computer assisted mobile manipulation would allow a user to pick up an object of interest. This mode of control is a form of supervisory control as the user selects a desirable sample while the robot arm will pick up the sample and places it into a sample collection area. Further discussion of supervisory control can be found in Section 5.1. This solution is proposed to solve depth perception issues with a single camera [1]. The disadvantage is that the method requires advanced sensors to carry out the task. This adds mass and introduces layers of complexity in hardware and software.

Another application of manipulators is saving astronauts the task of remote maintenance tasks during spacewalks. An astronaut's dexterity is severely limited when wearing a space suit; a robot arm would be able to improve it. Research into a telemanipulator has been occurring at the the German Aerospace Center (DLR). In a 1993 test, researchers were successful in the grasping of a free floating object on the Space Shuttle Columbia [16] using the ROTEX manipulator arm. A communication delay that varied between 4 and 7 seconds, made grasping moving objects difficult. The problem would be much easier in the case of

a statically located object. The engineers compensated for the delay in communication by predicting the location of the robot motion in tandem with the floating object's position. A control loop using a Kalman filter in conjunction with computer vision techniques were used to achieve the grab.

The Packbot has the option to become equipped with a manipulator arm. Autonomous Solutions, Inc has developed a user interface where upon a click of a mouse on an object of interest, the software will move a manipulator to the point of interest [17]. This group uses a set of redundant depth sensor and fuses the information. Autonomous Solutions has found that laser ranger finders work well for targets that are directly in front of the sensor only while a 3D camera setup does not work well on homogeneous surfaces. The combination of the two sensors compensate for weaknesses in each individual sensor.

Chapter 3

Mechanical Design

This chapter will focus on the current features of the robot arm. The requirements will be established and then verified at the end of the chapter. The design process used will be described along with the justification of certain choices.

3.1 Requirements for Teleoperated Chunk Sampler

The design challenge is to create a remote control device that picks up blast debris and secures it for transport on the GSR. The device must also have a means for making it's operation easy to use to reduce the amount of attention needed to accomplish tasks like picking up objects of interest. The proposed chunk sampler must merge with the existing GSR. It must fit into a volume dictated by the designer of the GSR and receive power and signal from power distribution and onboard microcontroller. The following lists the desired requirements for the sampling device.

Weight

The device must be lightweight as a result of the payload capacity of the RMAX. The weight budget has been set at 2.26 kg(5 lbm) for any sampling device on the GSR.

Volume

The sampler will be able to fit into a measured volume of 230 mm(9") by 265 mm(10.5") and 90 mm (3.5") space on the GSR. This is a measured volume on the GSR.

Collect Samples

The sampler will be able to pick up a soda can sized sample with a diameter of 65 mm (2.5”). This volume turned to granite would yield a mass of 1000 g.

Multiple Samples

The device will have the option to pick up a single or multiple samples in situations of no cross contamination. The device should be able to collect another sample and still retain previously collected samples.

Inexpensive

The solution must be inexpensive in material costs as the robot is disposable. The cost will be less than 1000 USD in direct material costs.

Simple Interface

The user interface needs to be easy to use allow for an operator to respond to dangers at their present location. The interface for the semi-autonomous mode of operation will have 1 handed operation.

Operate in an Outdoor Environment

The robot will be operating in an outdoor environment and needs to be dirt resistant. The device will use dirt resistant and enclosed components.

Beyond Line of Site (BLOS) Operation

The device will take it’s commands over a wireless link to the GSR via serial command packets.

Development Time

The sampler must be designed and built by a graduate student. The prototype will be developed over the course of 1 year.

Reach

The device will reach well into current camera view on the GSR. A proposed reach of 4 cm to 20 cm in front of the robot is the goal. This range was found by placing objects in front of the GSR and determining what was visible.

Camera View Obstruction

The device will not obstruct the view of the samples being collected.

Fabrication and Assembly

Parts will be cut using a mill and a water jet cutter. Common off the shelf (COTS) products will be used to expedite fabrication. Assembly will be done in-house.

Life Span

As the device is used for a short period of time and destroyed due to contamination, frequency loading is not considered. A minimum safety factor of 1.5 will be used.

These requirements guide the design process. A verification of these specifications can be found at the end of of the chapter in Section 3.12.

3.2 Condensing the Design Space

The possible solutions to the design challenge could be any sort of device that accomplishes sample collection and fulfills the requirements above. Any mechanism or device that fulfills the requirements could be a potential solution. Additional sensors can be added to improve localization of the arm to the GSR. Modifications could be made to any part of the ground sampling robot, just as long as the GSR still fits underneath the RMAX and is under the budgeted weight. The electronics of the GSR could be completely replaced to accomodate new actuation options or sensors on the GSR.

Degrees of Freedom for Sampler

There are two counteracting forces when it comes to determining the degrees of freedom of a potential sampling mechanism. As a result of the system's low weight requirements, a lower DOF is favorable. On the contrary, the operator would prefer an extended workspace, therefore a higher DOF would be necessary to allow a user to park the GSR and interact with a scene. A larger workspace is desirable as, the sampling mechanism can compensate for placement error. A set of concepts were looked at and eliminated before a 3 DOF arm was settled on. Revolute joints were solely examined as prismatic mechanisms tend to have a reduced workspace for the amount of space that they occupy [18].

(a) (b) (c)

Figure 3.1: Shows potential low degree of freedom solutions for a sampling mechanism. Image (a) shows a 2 degree of freedom serial manipulator. Image (b) is a 4 bar mechanism that traces a curve in a plane. Image (c) is 3 DOF manipulator that pivots about the an axis perpendicular to the sampling tray.

Figure 3.1(a) is a revolute-revolute planar mechanisms. With (a), the user would have to line up the sample with a work plane by maneuvering the robot before collection. This process can be aided with a visual overlay on the camera output. The user would position the GSR and then position the end effector over a target. From here, the sample will be collected. The arm would retract and store the sample either somewhere on the sampling tray, or within the gripper itself. With a 4-bar solution shown in (b), the user would have to position the robot so that the rock was positioned in a sampling area and then the user could proceed to retrieve the sample. By making the gripper or claw large, a designer can account for large positioning errors as is the case with the sampler shown in Figure 1.6. A downside to this is that a large claw obstructs a user's view of the collection area. To get around this issue, a secondary camera will be placed on the front of the GSR for the sole purpose of aiding an operator in positioning the GSR for sample collection.

In the end, (a) and (b) were seen as more unwieldy to operate than their reduced actuation was worth. These three methods also limit the locations where the rocks can be placed in a sample collection vessel on the GSR. Fine movements with the robot were also determined to be difficult to do as slipping at the tracks has been observed. In uneven or loose terrain, precise movement of the GSR cannot be obtained, a chunck sampler would need to account for this imprecision. The three DOF solution seen in (c) allows for greatest flexibility in the placement of the sample collection vessel. The operator will require a large workspace that only a multi-DOF robot arm can supply. A 3 DOF arm allows for the collection of multiple samples.

Sensing

The objective of adding a sensor is to allow for one click sample collection. An additional

sensor or set of sensors is necessary for semi-autonomous operation of the sampler. It is simply not possible to consistently locate objects in 3D space using a camera, specifically in unstructured environments. It is necessary to use an entirely different sensor that operates independently, in conjunction, or replaces the single camera. Depth sensors can be classified as point, line, or area sensors.

A point and line depth sensor may work in a full autonomous mode of operation but an operator may not see the correspondence between a point or line in the depth output and the camera view. A user will have difficulties matching what the sensor is outputting with what the image is showing. Knowledge of the volume the sample takes up is unknown, the user would have to guess as to whether or not a target is approachable. The use of an area sensor would be able to remediate many of these deficiencies. A user will be able to identify objects from the camera view in the depth view. Software would be able to determine the volume of an object which can be used to determine whether or not the sampler system would be able to grab it.

Depth information that orients samples relative to the GSR is necessary for mission success. The placement of the 3D sensor is the next consideration that will help with this orientation. One option is to have the 3D sensor mounted onto the RMAX. This would provide an overhead view of a sampling area that an operator can use to collect material. The concern here is that the RMAX will fly a minimum of 40 meters above the target area. This is to avoid disturbing the sample collection area with rotor down wash. Any information from the RMAX will be at too great a distance away from the GSR to be accurate whether it be a camera view or 3D depth information. Additionally at 40 meters away, samples will be difficult to discern from the background. The next option is to mount the 3D sensor directly to the GSR. This will provide the highest fidelity data but requires extra hardware to package up the point cloud data and send it back to the base.

Grasping

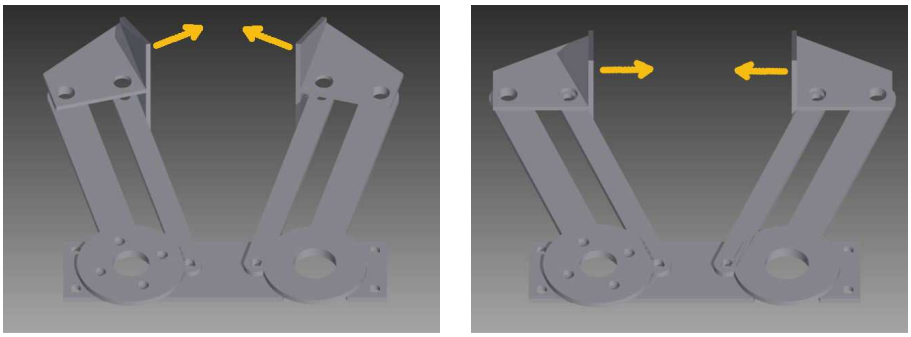
The function of the end effector is to grasp objects of interest, and grasping can be done a number of ways. First, negative pressure methods are discussed, secondly adhesive style grippers, and then mechanical grippers. Parallel grippers were eventually selected in the final design.

Negative pressure and a suction cup can be used to pick up objects of known shapes and sizes. Due to shape irregulaties and surface roughness, this option was eliminated. Another

option is using a jamming end effector [19]. Jamming involves the compression or relaxing of a granular material inside a rubber sheath and allows for the grasping of a wide assortment of geometries. There is a potential for shrapnel to puncture the rubber sheath holding together the granular material and it was decided that this solution was would not be robust enough. Negative pressure and jamming would also require extra weight in vacuum pumps on the GSR in order to create suction.

Another option is the use of adhesives on the sampler. A plate with an adhesive could be placed on objects of interest and the adhesive will bind them to the robot. Since the robot will be operating in a dirty or dusty environment, there is a potential for dirt collection which may prevent the collection of material. The addition of adhesive may also make the task of sample identification more difficult.

The use of a mechanical gripper will merge easily with existing hardware on the GSR. The potential gripper can simply draw from the same battery pack that is used to drive the GSR motors. There are two options for a mechanical gripper that would work in sample collection: angular and parallel grippers. With angular grippers, each finger rotates about a pivot point. Parallel grippers maintain orientation when the gripper opens and closes. An illustration comparing the two grippers when it comes to gripping objects can be seen in Figure 3.2. A component of force may force objects away from the end effector in angular grippers. Parallel grippers apply all their force in a way that maximizes the grip force applied to the sample.



(a) (b)

Figure 3.2: Angular grippers versus parallel grippers.

Grip Aids

After the method of gripping has been established, a way of improving the grip is necessary.

With plain parallel plates and an irregularly shaped object, there is chance of the object not making good contact with the plates and falling out. To remedy this a couple of grip aids were investigated. One way is to use a set of bristles, like a hair brush, that conform to the shape of the object of interest while maintaining a certain stiffness. This idea turned into the use of foam on the flat gripper surfaces. This requires a small amount of glue to attach to the plates and can be accomplished in a short time.

3.3 Motor Performance Requirements

After deciding on the overall shape of the sampling system, it is necessary to choose a type of actuator to create the desired motion. This section will go through the process by which the motors were selected. Motors dictated the geometry of the joints. A weight budget for the joints and linkages were decided upon.

A static calculation was done using a force balance in the static case. A dynamic model was also created in SimMechanics and used to confirm the results from the force balance. The motors were then selected based on the maximum torques found in simulation.

3.3.1 Static Torque Requirements Estimation

In order to get a better understanding of what sort of motors will be necessary for the robot arm, it is necessary to determine the torques the robot will see at the joints. An initial estimate of the torque requirements can be garnered from a simple moment balance with the arm horizontally outstretched. The free body diagram in Figure 3.3 will guide the calculations as follows.

$$M_1 = W_1(\frac{L_1}{2})+W_2(L_1)+W_3(L_1+\frac{L_2}{2})+W_4(L_1+L_2)+W_{ee}(L_1+L_2+L_3)+W_o(L_1+L_2+L_3+L_o) \tag{3.1}$$

$$M_2 = W_3\frac{L_2}{2} + W_4L_2 + (L_2 + L_3)W_{ee} + (L_2 + L_3L_o)W_o \tag{3.2}$$

From Equations 3.1 and 3.2, the maximum torques the shoulder and elbow joint will see

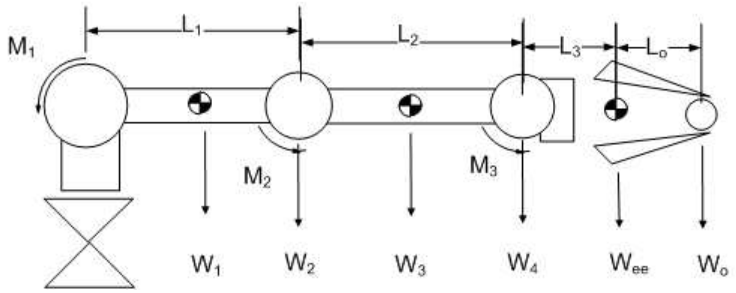


Figure 3.3: Free body diagram where the static torque at the first robot joint will be at a maximum. Weights (W) and lengths (L) are shown.

can be calculated. M_1 corresponds to the moment at the shoulder and M_2 is the moment at the elbow. Moments $|M_1| = 2.02$ Nm, (286 in-oz) and $|M_2| = 0.91$ Nm (124 in-oz) were calculated for estimated link weights, joint weights, and sample weight. The sample W_o was set to 0.25 kg, W_{ee} to 0.15 kg, the joints (W_2,W_4) were set to 0.1 kg, and the links (W_1,W_3) were set to 0.01 kg. The L values were set to sub-meter values as follows $L_1 = 0.2$ m, $L_2 = 0.15$m, $L_3 = 0.05$m, and $L_o = 0.03$m. These moments were compared with the dynamic simulation that is described in Section 3.3.2.

3.3.2 Dynamic Torque Requirements Simulation

SimMechanics was used to get another set of motor torque requirements. Angular positions, angular velocities, and angular accelerations were generated and input the SimMechanics model. These numbers were based on values for angular velocity and angular acceleration that are much higher than what will be seen with the actual robot arm, this is to ensure conservative estimates of the torque requirements.

The arm was modeled as 3 joints and 3 rigid bodies pinned to the ground with a revolute joint with an axis pointed in the direction gravity is acting. All the joints are set to revolute joints except for the end effector and the sample which are simply welded to the elbow linkage.

The moment of inertia for each link was calculated by assuming the links were simply thick walled cylindrical tubes. The moment of inertia for these tubes are represented by equations 3.3, 3.4, 3.5 [20]:

$$I_x = \frac{1}{12}m(3(r_1^2 + r_2^2) + h_1^2) \tag{3.3}$$

$$I_y = \frac{1}{12}m(3(r_1^2 + r_2^2) + h_1^2) \tag{3.4}$$

$$I_z = \frac{1}{2}m(r_1^2 + r_2^2) \tag{3.5}$$

These inertias for the hollow tubing were calculated about their centroid. The variable h_1 represents the height of the tube, the values of 0.178 m (7 in) for the first link and 0.127 m (5 in) for the second link were used in this simulation. r_1 and r_2 respectively represent the inner and outer diameter of the tube. The values of 0.0508 m (2.0 in) for r_1 and 0.0483 m (1.9 in) for r_2 were used. Finally, m is the mass of the link which were 0.04 kg (0.09 lbf) and 0.03 kg (0.07 lbf). These values are much higher than what was actually used on the robot arm. The reason for this is that the design was not finalized at this stage of the process and it was desired to get conservative estimates for torque requirements.

The end effector was modeled as a rectangular prism with equations 3.6, 3.7, 3.8.

$$I_x = \frac{1}{12}m_r(h_r^2 + w_r^2) \tag{3.6}$$

$$I_y = \frac{1}{12}m_r(w_r^2 + d_r^2) \tag{3.7}$$

$$I_z = \frac{1}{12}m_r(h_r^2 + d_r^2) \tag{3.8}$$

The mass of the end effector or rectangular prism is represented by m_r. The dimensions of the rectangular prism are represented with height h_r, depth d_r, and width w_r. The mass of the end effector (m_r) was assumed to be .250 kg (0.53 lbf). The height (h_r) as 0.02 m (0.8 in), the depth (d_r) as 0.02 m (0.8 in), and the width (w_r) was chosen as 0.1 m (4 in).

The sample was simulated as a sphere with inertia shown in equation 3.9

$$I_x, I_y, I_z = \frac{2}{5}m_s(r_s^2) \tag{3.9}$$

The mass of the sample is represented with m_s and the radius is r_s.

The angular velocity was set at 2 radians per second and the angular acceleration was set at 1.0 radians per squared second. This yields a position, velocity, and acceleration signal which is used to command each joint separately.

The maximum torques seen by the motors from SimMechanics were found to be 0.075 Nm (11 in-oz) at the base, 2.04 Nm (189 in-oz) at the shoulder joint and 1.03 Nm (146 in-oz) at the elbow. This is a marginal increase in torque from the static calculation (equation 3.3.1) and shows that as long as the motion of the arm is kept slow, the motors will not reach their limit. The SimMechanics block used is shown in Figure 3.4.

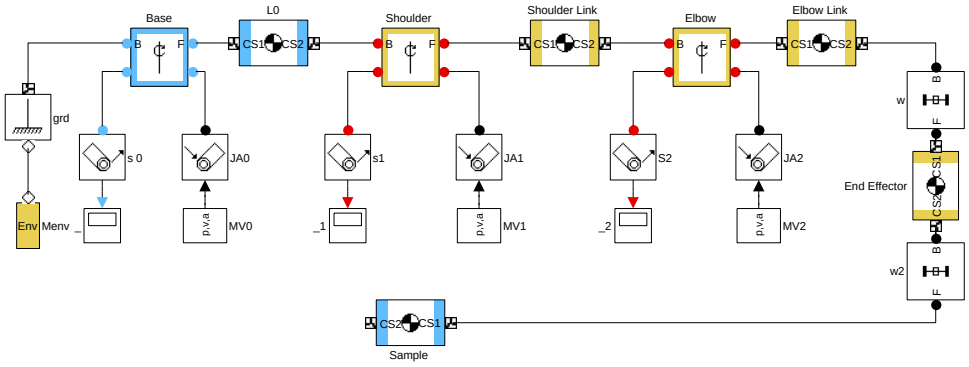


Figure 3.4: The SimMechanics block used to determine the max torques that will be seen by the robot arm joints is shown. The simulated arm has an end effector and a mass to simulate a worse case scenario.

3.3.3 Motors Selected

Brushed DC motors, stepper motors, and hobby servos were considered for the joints of the arm. In the end, hobby servos were selected as they would decrease development time and are inexpensive. These factors play an important role in making a materially inexpensive robot arm. The decreased development time is attributed to the fact that the servos already have feedback and controls working out of the box.

A single stepper motor was used for the base of the robot as it allowed for the actuator to be placed away from the base of the robot. Stepper motors were generally found to

be weaker than their servo counterparts. The advantage to using a stepper motor is that they can operate in open loop assuming that it is not in a high torque situation. Servo motors typically have limits to their motion, a stepper motor can be stepped in a direction continuously and precisely. Continuous operation is desirable as a belted solution or geared solution may be used for the base joint. Hobby servos can be modified to run in a continuous fashion, but it is difficult to determine where the servo has turned to without some sort of feedback. Since, stepper motors can be placed at an angular location precisely, a stepper was chosen for the base of the robot arm.

Plain DC motors were eliminated as they require a gearbox, encoders, and a control board. Plain DC motors examined for the purpose of the robot arm tended to weigh more than servos and were much more expensive than the hobby servos.

The hobby servos and the stepper motor picked meshed well with existing hardware and thus required minimal extra control hardware. All the motors are controlled by an Arduino that receives its commands over a serial link originating from a PC. For a breadboard representation of the circuit used see Appendix E.

The information from a set of commercially available servo motors were pooled and sorted. Figure 3.5 shows the motors positioned on a 'torque per mass' and 'price' axis. An ideal motor would be low cost and have a high torque to mass ratio and thus would appear in the upper left corner of the graph. Simultaneously, it is necessary to hit the torque requirements at 1.6 times the simulated results for the elbow and shoulder. The torque divided by mass metric was chosen as it penalized motors that were weak and had a large mass. The motors ultimately picked are shown with a larger marker and more detail is shown on Table 3.1. These motors were picked for their low cost and high torque to weight ratio.

Table 3.1: Motors picked for final arm. The base joint uses a stepper motor while the other joints make use of hobby servos. For the base stepper motor, the holding torque is used instead of the stall torque on this table.

Motor	Static Torque (Nm)	Dynamic Torque (Nm)	Stall Torque (Nm)	Actuator Mass (g)
Base- Pololu Stepper Motor	-	0.075	0.14	180
Shoulder- Hitec HS-7950TH	2.02	2.04	3.40	68
Elbow- Hitec HS-5685MH	0.91	1.03	1.65	60
Gripper- JRSport Standard	-	-	0.99	56

The motors picked exceed the torque requirements generated in Sections 3.3.1 and 3.3.2. All the motors were specified above their simulated targets. Operating a motor for an extended period of time near their torque limits reduces the life of the part. In order to avoid replacing motors frequently, the motors were specified with certain safety factors. The base was given more power to overcome the friction at the slewing bearing and to reduce the likelihood of skipped steps. The motor specified has a torque 1.8 times greater than the value garnered from the dynamic model. The torque from the base will be amplified by means of a timing belt pulley system described later in Section 3.7.1. The shoulder and the elbow were matched with servos that were 1.6 times more powerful than what the SimMechanics max torques output.

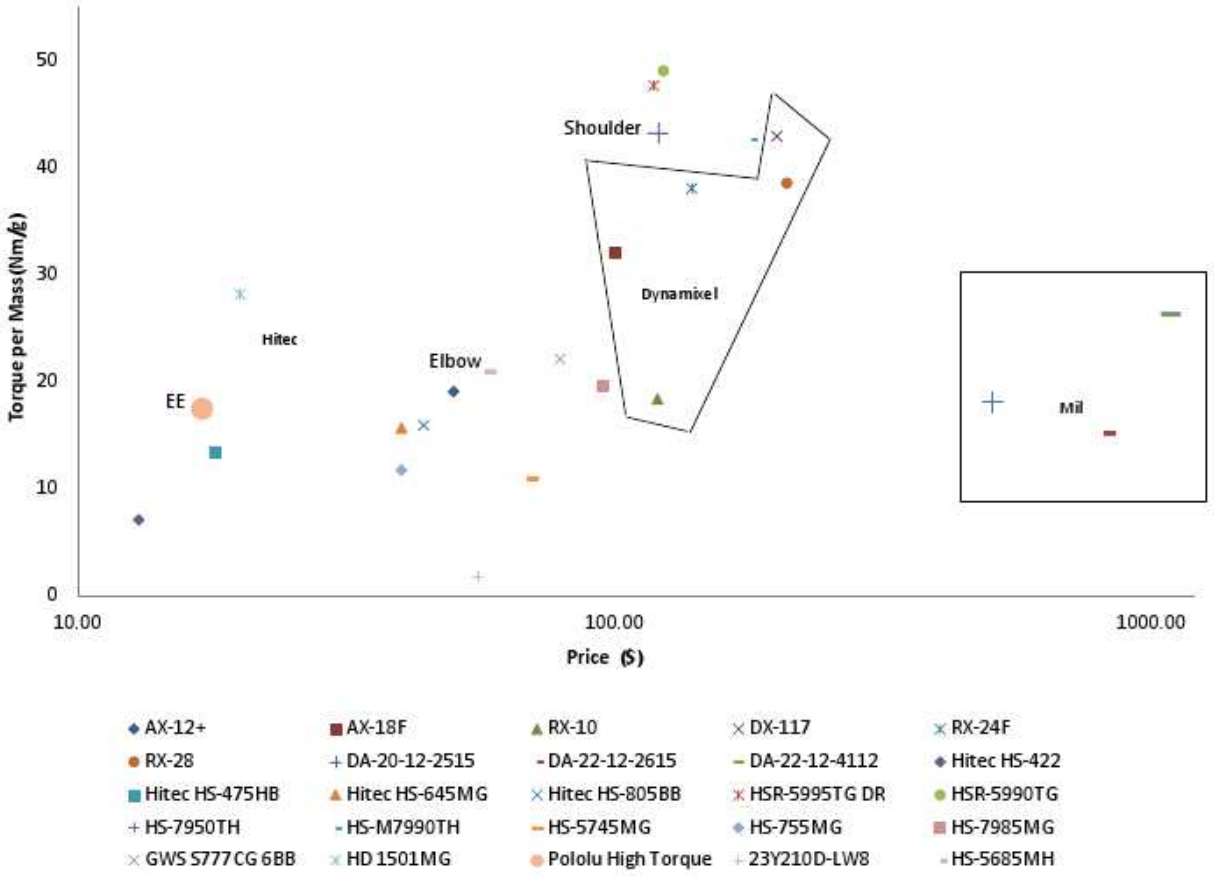


Figure 3.5: Plot of motors considered restricted to torque per mass and price. The motors picked are labeled as 'Shoulder,' 'Elbow,' and 'EE.' The motors are grouped by manufacturer. 'Mil' represents a set of military servos.

3.4 Link Members

The tubing used in this robot arm was a roll wrapped uni-directional carbon fiber tube. This tubing can be found at carbonfibertubeshop.com and has a nominal outer diameter of 0.70 inches and the inner diameter of 0.625 inches.

Tubing, independent of material, offers great stiffness regardless of which direction the forces act. Tube members also offer better resistance to the torsional loading [13] that will be developed from the offset nature of the joints required for collapsibility. This, and the fact that tube robot arms are prevalent in many military and space robotics applications leads to why it was chosen over other cross sectional shapes.

From the Table 3.2, it can be determined that carbon fiber offers high strengths and low weights while not adding much in the way of material costs. Yield strength would be the ideal choice when comparing metals but since yield strength information does not exist for carbon fiber, the ultimate strengths are compared.

As making a light-weight robot is the top priority, the link material was ultimately chosen to be a uni-directional carbon fiber. A stress calculation confirming that this link choice will withstand the static loading can be seen in Appendix B.

Table 3.2: Potential robot arm material compared based on density, ultimate strength, and pricing. Data was obtained from various distributors from the web.

Material	Density $\frac{\text{lbm}}{\text{in}^3}$	Ultimate Strength ksi	Price/Length $\frac{\$}{\text{Inch}}$
Titanium (Grade 9)	0.161	125	8.71
Aluminum (2024)	0.098	64	0.88
Carbon Fiber (braided)	0.080	275	2.81
Carbon Fiber (uni-directional)	0.065	7.2-217	0.95

3.5 Shoulder and Elbow Design

The shoulder and elbow for this robot arm has two functions. The joints must primarily serve as a pivot point for their links. They must also hold the servo in place and distribute

the loading created by the arm being fixed to the joint. The elbow joint has an additional purpose of joining the link being moved by the shoulder joint to the joint that is directly linked to the end effector. In order to distribute the loading, the joints are designed in a double shear configuration. To allow the mechanism to fold, the double shear joints must be offset in away that allows folding.

Clamps were used at the shoulder joint and the elbow joint to attach the carbon fiber links to the motors drive shaft. This was done as holes in the carbon fiber would weaken the material and produce larger stress concentration immediately around the hole. The clamps along with a few other components were water jet by Metal Processing Inc., in Fairlawn, Virginia.

Figure 3.6 shows an exploded view of the shoulder joint. 'A' is the Hitec HS-7950 which has a titanium gear train connected to a titanium spline. The spline is attached to an aluminum servo horn ('C') with a 4 hole pattern. These patterns are used to bolt into a machined T-bar piece that transfers the motor load to a set of water jet clamps ('B'). These clamps retain the carbon fiber tube in a slight interference fit. On the opposite side of the motor, the clamps meet together at another T-bar which is bolted into a 3/8" diameter set screw hub ('E') which turns with a shoulder bolt by action of the set screw. The 3/8" shoulder bolt rests within an IGUS® iglidur® M250 flanged bushing shown in 'F.' The shoulder bolt aids in distributing the loading of the arm. The whole assembly rests on a machined U-channel piece labeled as 'H.'

Figure 3.7 has the elbow in an exploded view. 'A' is the shoulder bolt that is used to distribute the loading at this joint and it rests on another flanged bushing ('B'). 'C' is a set screw hub that attaches to the shoulder bolt and a T-bar piece ('E'). The T-bar attaches to a set of clamps ('F') that go around a carbon fiber tube. The clamps meet on the motor side around a T-bar that bolts into a 3/8 inch bore clamping hub ('G') which clamps around a servo shaft attachment with the necessary spline to attach to the Hitec HS-5685MH ('H'). The added function for the elbow joint of needing to join the shoulder joint link with the elbow joint link is accomplished by a bolted box structure seen encasing the servo motor in 'I' and 'J'. The carbon fiber tube from the shoulder joint is press fit into 'I.' The parts labeled 'J', 'I', and the U-channel 'D' provide the main structural support for the elbow joint.

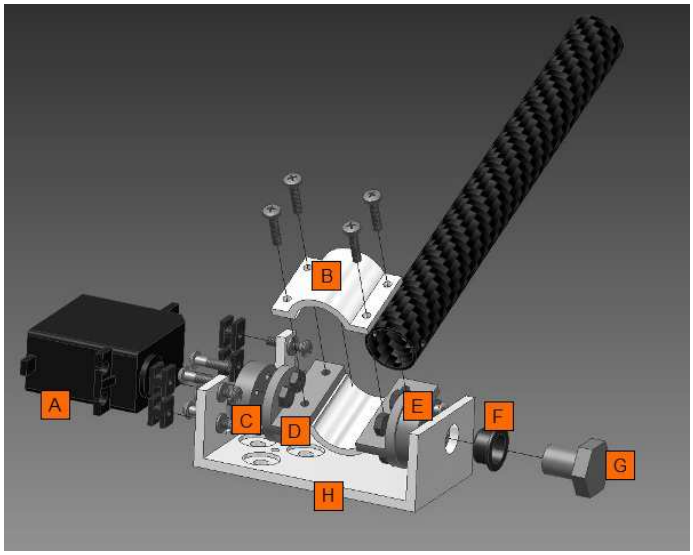


Figure 3.6: The shoulder joint, exploded.

3.6 End Effector

The function of the end effector is to provide a way for the robot arm to pick up objects of interest. The end effector needs to fit into the elbow linkage of the arm. The end effector uses two parallel linkages with foam covered grippers. The foam helps reduce the gripping force necessary by effectively increasing the coefficient of friction necessary to keep the sample between the two grippers.

The end effector is characterized by two parallelogram mechanisms that are geared together. Each parallelogram mechanism is a 4-bar mechanism with opposite links being equal in length. Using the Grashof condition below in (equation 3.10), the condition reduces to an equality.

$$S + L \leq P + Q \tag{3.10}$$

$$S = P, L = Q, S + L = P + Q$$

As a result of the equality, this linkage is classified as special-case Grashof [21] and has favorable properties. The special-case Grashof configuration has the advantage of allowing the gripper plates to remain parallel to each other throughout the motion of each linkage as

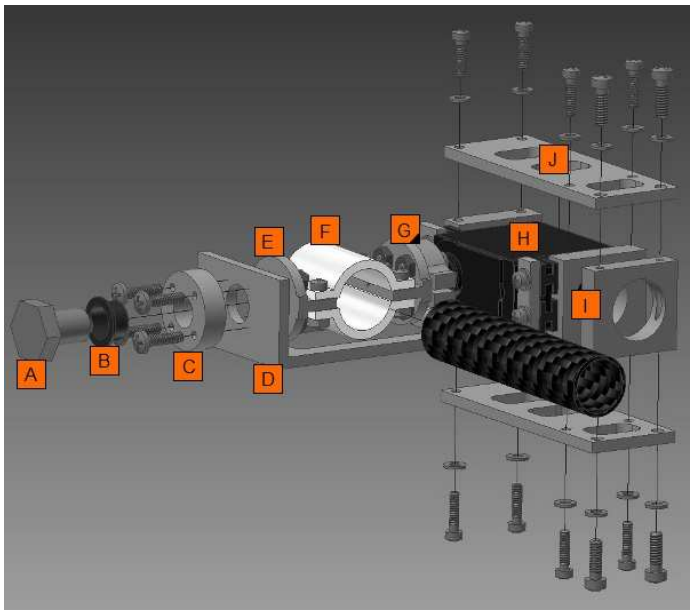


Figure 3.7: The elbow joint in an exploded view.

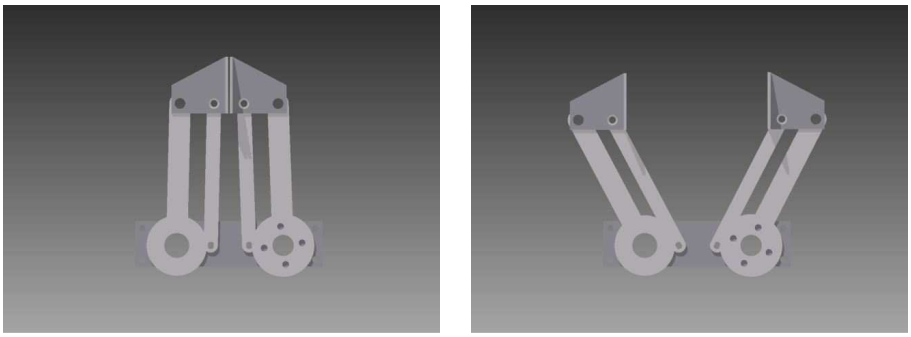
shown in Figure 3.8. This was done to make sure the gripper plates approach the sample in a similar orientation regardless of the sample width. The utility of this is illustrated in Figure 3.9, where a gripper plate welded to a link would exert a component of its normal force in an undesirable and ineffective direction. In addition to this, the special-case Grashof configuration allows the user to have a better view of the ground as shown in Figure 3.10.

The angle that the linkages sweeps out is approximately 15 degrees. The small angle prevents the gripper from over shooting the target when teleoperated and prevents the mechanism joints from becoming collinear.

Two 60 tooth acetal gears with a thickness of 3/16" were affixed to the largest of the two links shown in Figure 3.8. Using the Lewis equation to estimate the bending stress in a single gear tooth, in Equation C.1 found in Appendix C, a safety factor of 2.6 for static loading in the gears was chosen [20].

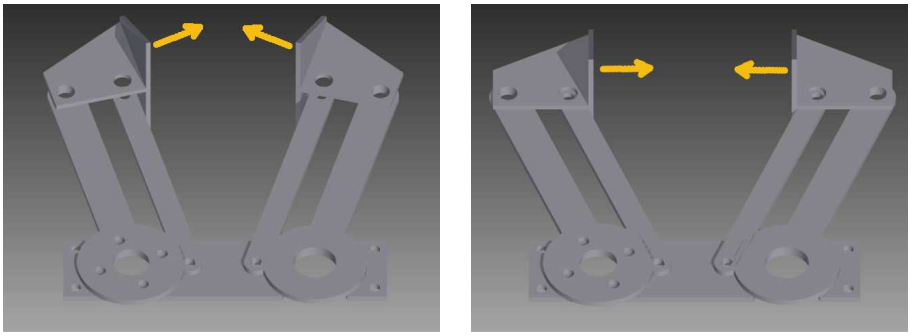
The links are made out of 1/16" aluminum. Appendix D has a series of calculations that confirm the strength of the linkages for picking up samples. The links move on separate planes as shown in Figure 3.11. This offset increases the effective second moment of inertia and disallows the linkages from deflecting as much as it would in a coplanar case. These links

pivot about wrench cut shafts held in place with circlips. The alternative to using circlips is the use of shoulder bolts and nuts to keep the assembly together and create an axis of rotation. Circlips were used as they are lighter than bolt heads and nuts.



(a) (b)

Figure 3.8: End effector or gripper closed and opened. The end effector has two parallelogram mechanisms that keep the gripper plates parallel to each other throughout their motion.

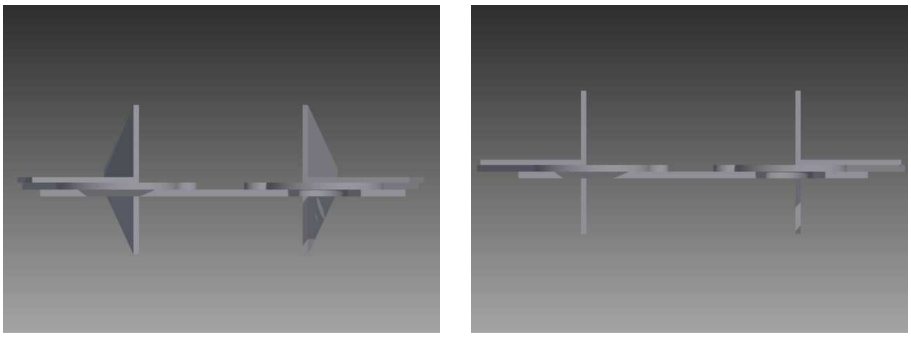


(a) (b)

Figure 3.9: Four-bar end effector versus alternative. The parallelogram mechanism ensures the sample is not inadvertently forced out of the grip. The image (a) shows the gripper plate angled at 15 degrees with the main motor linkage; as a result of this configuration, there is a force component that pushes the rock out of the grasp. The parallel mechanism (b) resolves this by ensuring the force vectors are collinear.

3.7 Arm Base Design

The base needs to provide a way for the robot arm to pivot around on an axis. This allows the user to pick up objects in a larger volume than without this joint and actuation. The base joint uses an IGUS® slewing bearing to take the loading developed by picking up rocks.



(a) (b)

Figure 3.10: View obstruction reduced with parallelogram mechanism. The gripper plates obstruct less of the view in a eye-in-hand camera configuration.

The base joint also takes the torque supplied by the stepper motor described in Section 3.3.3 and move the robot arm.

Slewing bearings can take the combination of axial, radial, and moments better than other bearing solutions. The slewing bearing used is an IGUS® PRT-20-AL. The outer ring is made out of anodized aluminum and is used to mount the slewing ring to the sampling tray. The moving inner ring is made out of a self lubricating plastic called igidur® J4. The bearing requires no maintenance, is dirt resistant and can withstand loading much higher than what the arm can supply. As seen in the product catalogue, the PRT-20-AL can take a tilting moment of 60 Nm, a max radial dynamic load of 500 N, and a max axial load of 4000 N. The movement of the slewing bearing allows for a low slop axis of motion but requires a large moment to overcome friction. The friction increases the torque requirement of the joint beyond that predicted by the dynamic model. The consequence of this is skipped steps at the base actuator which necessitate the use of an encoder at the joint or on the motor for 3 degrees of automation. The skipped steps do not affect teleoperation as a human operator can easly compensate for the skipping.

3.7.1 Drive Solution

A timing belt was chosen as the drive solution for the base. The MXL tooth profile was chosen as it is able to take the loading and has the lowest belt width (1/4") of timing belts on McMaster-Carr. In the final robot, a fiberglass reinforced neoprene timing belt of trade size 88 was used with two aluminum pulleys. The smaller of the two pulleys has 18 teeth

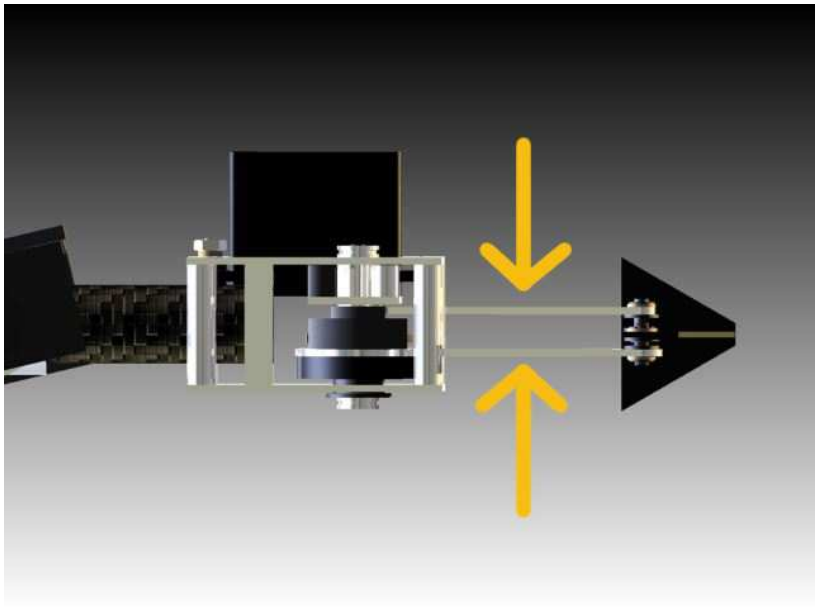


Figure 3.11: The profile of the end effector. The linkages for the mechanism are not coplanar. As a result, the linkages offer better deflection resistance.

while the bigger pulley has 60 teeth. This gives a motor torque amplification of 3.3, bringing the total torque output of the stepper motor to 0.46 Nm (65 in-oz).

The timing belt was seen as a more favorable solution than gears since it offers more flexibility in where the stepper motor is placed. A directly mounted servo solution was rejected as a result of the servo taking up to much vertical space. Drive solutions without teeth were considered but concerns about slip between the drive pulley and the belt prevented the idea from being developed further.

Additionally, the gear ratio of 3.3 increases the resolution by which the base can be controlled. At 200 steps per revolution, the stepper motor's step resolution is 1.8° at the pinion. This angular displacement gets reduced down to 0.6° per step on the larger timing belt pulley. This movement resolution can be reduced even further by taking advantage of microstepping by means of the stepper motor controller.

3.8 Sample Collection Vessel

The sample collection vessel (SCV) is where the rocks are placed after they are picked up by a user. The SCV functions to retain samples during transit. Mainly, the SCV is folded sheet metal that has been attached to the sampling tray. The arm places its elbow joint over the SCV and keeps samples from becoming lost during transit.

3.9 Material Costs

The direct material costs have been included in this paper and can be found in Appendix G. Two sets of prices are given, the total price paid and the price per robot. The price per robot takes the number of a certain item used, divides this value by the total number ordered and multiplies this by the total cost of the order. This gives a value that is closer to what would be paid if a higher volume of these robots were produced.

The costs fall under the 1000 USD set forward in Section 3.1 at $797.63 seen in table 3.3. The breakdown of costs for each subassembly can be found in table 3.3. The total cost is the summation of the costs of each subassembly. In addition to the total, a second total was calculated which is 1.2 times the total described previously. This is to account for costs such as shipping, use of scrap aluminum, and items found in Unmanned Systems Lab shop area.

Since all of the bushings were donated through IGUS's Y.E.S. program, the total in Table 3.3 is lower than what it would be if these robot arms had to be built without any donations. Table 3.4 shows that these bearings would cost about $150 if the least expensive alternatives were used. This brings the per robot cost up to $740.

Table 3.3: Truncated Sales Bill of Materials

	Price Paid ($)	Price per Arm ($)
Base	65.68	59.09
Shoulder	245.41	176.51
Elbow	89.26	113.35
End Effector	218.60	138.62
Tooling	34.09	0.00
Totals	653.04	487.58
Total + 20%	797.63	585.09

Table 3.4: Bearing cost estimate if purchased.

Qty	Name	Unit Price ($)	Price ($)
4	Flanged Bushing 3/8”	5.68	22.72
12	Flanged Bushing 1/8”	3.50	42.00
1	Slewing Bearing	88.00	88.00
—	Total	—	152.72

3.10 Command Modes

On the GSR, an Arduino Mega and an EasyDriver were used to control the motors of the robot arm. The Arduino Mega is a microcontroller kit and the EasyDriver is a stepper motor controller. These are inexpensive methods for controlling the robot arm. The ground sampling robot will make use of an Arduino to interpret serial commands and carry out operations based on what commands are received. Motor movement operations are controlled by PWM signals sent from the Arduino. Previously, this control was limited to the speed of rotation of each tread. This control has been extended to control the arm. The EasyDriver supplies the necessary current to the stepper in order to get motion out of it.

The arm motors have different addresses that are designed to be easily interpreted by a human. All arm commands start with the letter ‘A’ and is followed by a second letter signifying which motor will be moved in particular. The second letter could be a ‘B’ for base, a ‘S’ for shoulder, ‘L’ for elbow, or ‘E’ for the end effector.

There are two codes by which a user can control the robot arm: relational and absolute mode. In order to switch between the two modes, the Arduino needs to be loaded with the appropriate software for the application. A unified command packet structure that allows for both relational and absolute control of the robot arm is proposed in section 6.2.

In relational mode, a single command byte is given to a microprocessor and the motor moves to a new location that corresponds to the motor’s current position plus an adjustment factor of -10 or +10. 10 is the maximum number of steps that can be taken after a single serial communications exchange. This mode is used when the arm is teleoperated by a joystick.

In absolute command mode, the desired position is commanded from the main controller directly with 2 bytes of information. This command setup is used when the robot is in computer assisted sampling mode. In the final iteration of the code, it would be desirable

for a user to switch between absolute mode and relational mode so that they would be able to operate the arm using a joystick and switch to computer assisted mode at will.

3.11 Proposed Operations

The GSR and chunk sampler will both be controlled by a PS2^{TM} styled joystick controller. The teleoperator inputs are interpretted by a LabVIEW Virtual Instrument and relayed over a Zigbee radio to the GSR from the base station. An Arduino Mega interprets the serial commands from the Zigbee and determines the appropriate actions for the robot. This can involve moving the robot or moving the robot arm.

Once the GSR is delivered to a location for sampling, the robot will move from its resting position to a home location where the shoulder link is vertical as well as the elbow link. If the user is in manual control, the teleoperator will control the rate of motion of the robot links by moving the joystick's analog sticks. The user identifies an object of interest and moves towards the object.

Once the robot is in close proximity to the object of interest, the operator will switch to arm control and position the arm to capture the sample using a PS2 style game controller. Once the sample is in the manipulator's grasp, the operator will turn the arm to a mechanical limit and drop the sample into the sample collection vessel (SCV), which is simply folded aluminum that retains samples during transit. The operator then has to collapse the robot arm over the sample collection vessel, forming a lid that prevents collected material from leaving the ground robot during flight. The GSR, still tethered to the UAV, will then be returned to a predetermined location.

In computer assisted mode, instead of an operator using a game controller to pick up an object of interest the user uses a mouse click to select the object of interest. This would reduce the amount of skill and time needed to perform the sample collection. A computer calculates the distance out to an object of interest and moves the sampling mechanism out to the point of interest to retrieve the sample.

3.12 Design Verification

Verification confirms whether or not a design satisfies the initial set of requirements. The requirements are listed below as well as whether or not they were accomplished.

Weight

The USLArm meets the goal of 2.2 kg (5 lbm) at 1.5 kg (3.3 lbm).

Volume

The robot arm fits into the volume of 230 mm(9") by 265 mm(10.5") and 90 mm (3.5") on the GSR.

Collect Samples

The sample size and mass was seen as flexible and a robot arm solution was decided upon to satisfy the reach requirement. The desired sampler in the arm configuration will be able to pick up a soda can sized object with a diameter of 65 mm (2.5"). A soda can's volume turned to granite would yield a mass of 1000 g. For the robot arm to work, a static calculation of the shoulder torque requirement yields a necessary torque of 5 to 6 Nm. This is well above the torques that are deliverable by hobby servos with the reach desired. As a result, a subsample was designed for at 250 g and a diameter of 44 mm (1.75"). The USLarm will be able to collect a number of samples in increments of 250 g.

Multiple Samples

As mentioned above, the USLArm is capable of fitting multiple samples into the SCV.

Inexpensive

The direct material costs of the USLArm falls at $740 which is under the $1000 target.

Simple Interface

The underlying software is present for single click collection. A graphical user interface will improve the usability of the code. With the computer assisted mode, a user need only click on an object of interest and hit enter. The robot arm will then reach out and pick up the object of interest. Currently this mode of operation is run directly from Matlab and is not ready for direct user interaction as parts of a script are run.

An appropriate 3D sensor needs to be specified. A 3D sensor was not purchased due to a

funding shortage during the time of development. An ideal choice would be a device similar to a Kinect but designed specifically for ground robot operation.

Operate in an Outdoor Environment

The robot will be operating in an outdoor environment and needs to be dirt resistant. The device will use dirt resistant and enclosed components. The IGUS slewing bearing and the use of enclosed hobby servos help make the robot arm dirt resistant. An enclosure for the timing belt assembly will be necessary to prevent foreign matter from reducing the effectiveness of the power transfer.

Beyond Line of Site (BLOS) Operation

BLOS operation has not been tested on the current robot arm but has been tested on a similar robot arm with success. The USLArm is ready to recieve commands over a wireless link to the GSR via serial command packets. This functionality will be tested by a member of the USL.

Development Time

The USLArm was designed, built, and tested in 1 year.

Reach

The device is capable of reaching from 2.5 cm to 25 cm which exceeds the requirement on both extremes. Originally, the device was proposed to have a reach of 4 cm to 20 cm in front of the GSR.

Camera View Obstruction

Due to the collapsible nature of the robot, the operator can maintain view of the sample being collected.

Fabrication and Assembly

Most fabrication and assembly was accomplished in-house. Water jet cut parts were ordered from Metal Processing Inc, in Fairlawn, VA. Lathework was kept to a minimum, but few components in the end effector needed grooves cut for circlips. These cuts were done by an undergraduate student working for the USL. Many components were ordered from vendors that were not machinable in-house, particular components that required spline cuts to fit into the output shafts on the servos.

Life Span

As the device is used for a short period of time and destroyed due to contamination, frequency loading is not considered. A minimum safety factor of 1.5 is used. As this is a lab prototype, and replacing components frequently would be inconvenient, many parts are made much stronger than they need to be. An example of this is the end effector links shown in Appendix D.

Chapter 4

Display Concepts

This chapter will describe methods that were created in order to improve the user interface of the robotic manipulator. First, 3D models of the robot pose were generated. Secondly, IR sensor information was represented in the the 3D model. Third, a rock detection algorithm was prototyped.

This work attempts to solve an important issue in single camera view telemanipulation; a user has a difficult time determining depth from just a single camera view. Using vision alone, a user can determine depth by prodding a sample at various locations and seeing how the object responds to the robot arm. This method of feedback can take time, thus methods of improving the sensing of depth are investigated in this chapter. A 3D model along with point depth sensors were examined but found to convey too little information.

Using a 3D model splits the attention of the operator but can be used as a training tool for use of the robot arm [10]. With the IR sensor display in 3D, the correspondence between the 3D model and the single camera view was hard to establish. The output of the IR sensors were difficult to link to depths in the environment as displayed by the single camera. The Rockfinder algorithm, discussed later in the chapter, was found to detect too many false positives but was used in Chapter 5 for the computer assisted mode.

4.1 3D Model of Robot Pose

A 3D model of a 3 DOF arm was produced in LabVIEW. Command data given to a robot arm can be used to move the virtual robot arm. This acts as an aid to the user by providing views of the robot arm that would have not been possible from an *in situ* camera view.

The model was created with a series of cylinders that are stacked and given commands that cause rotation to desired angles. The LabVIEW 3D picture control toolbox was used to create the scene. This code can be adapted to any 3 DOF manipulator.

Figure 4.1 shows the 3D robot on a platform that simulates the sample collection tray. On the tray is a fix link labeled 0 by which the rest of the robot arm rotates about. This link is proportional to how high the robot arm is raised off the platform. On the actual robot, it encompasses the IGUS® slewing bearing and the lower U-channel piece up to the axis of rotation of the shoulder joint. Link 1 represents the shoulder link and goes from the axis of the shoulder motor to the axis of the elbow. Link 2 represents the elbow link and goes from the elbow axis out to the end effector.

It was found that the information obtained through the 3D scene was not enough as the user still did not know the location of rocks in respect to the arm's frame of reference. In order to help this, two IR sensors were mounted to the robot arm and will be described in section 4.2.

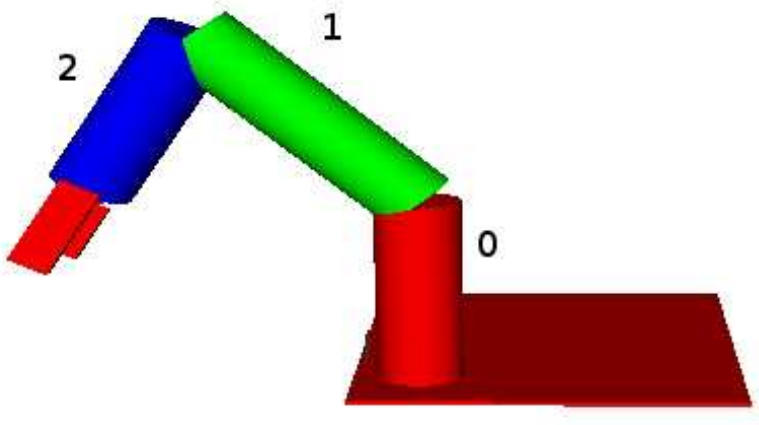


Figure 4.1: Labview generated 3D representation of the robot arm.

4.2 Infrared Sensor Depth to Objective

Two Sharp® IR sensors were mounted to the robot arm. One was mounted to the end effector and the other was mounted to the base of the robot arm. The output was shown as elongating or shrinking rectangular prisms on the graphic as shown in Figure 4.2. The user would be able to determine the location of an object out in space by moving the end effector around and keep track of where minimum locations were found. This sweeping motion was determined to be the non-contact analogue to nudging or prodding the rock and determing contact based on the camera image.

This solution was seen to be too similar to the user simply nudging the rock to determine its location in space as a result of the sweep pattern that was necessary. Additionally, the user would not be able to identify the object based on the IR information conveyed through the 3D model. This would reduce the overall user confidence in the accuracy of the data and would be detrimental to SA [22]. In place of the IR sensors, the use of the Kinect™ in computer assisted mode will be discussed in Chapter 5. The Kinect™ or an equivalent sensor allows the user to to see the scene and identify objects as they appear in the real world from the sensor data in itself.

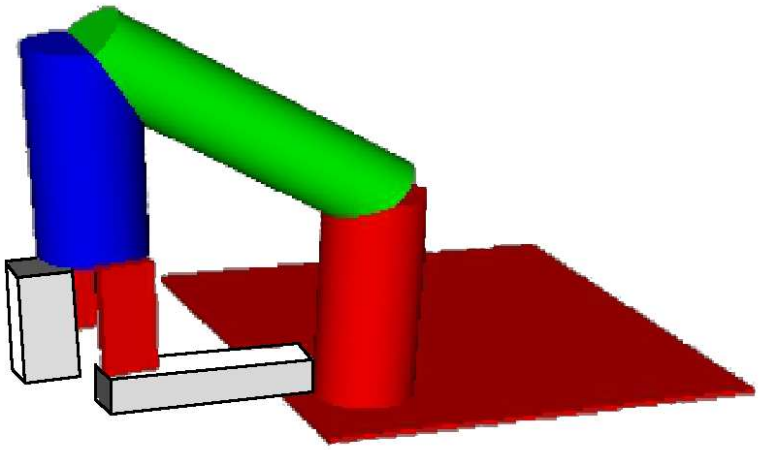


Figure 4.2: The 3D representation of the robot arm with the IR sensor data included.

4.3 Rock Detection Using a Single Camera

Methods for rock detection were investigated to improve a user interface for the robot manipulator arm. The author decided on using a particular rock detection algorithm that involves using two noise reducing filtering techniques and the Canny edge detector. The algorithm implemented was heavily influenced by the Rockfinder algorithm developed by the OASIS research group at JPL [23]. Matlab image segmentation routines were then applied to the edges detected to create closed contours. The original idea is to highlight areas of interest in a single camera view while the user is navigating a terrain, in order to aid the user in determining the location to collect a sample. The Rockfinder algorithm was applied to a series of still images of rocks. In section 5.3, the Rockfinder algorithm is applied to point cloud data.

The Rockfinder algorithm has three parts: denoising, bilateral filtering, and edge detection. The original OASIS group Rockfinder has additional code that isolates rocks from the edge information. Instead of this, image segmentation routines in the Matlab Image Processing Toolbox were used to close edges and remove extraneous lines detected.

The denoising involves putting the image into Fourier space where information above a certain threshold is removed. The frequency space information is then transformed back into a normal image [24]. Denoising has the effect of reducing the noise and some of the finer details of the image while preserving the phase and magnitude information of the original image. The function 'noisecomp.m' was used for denoising from the Kovesi website. The inputs to this function are the image and a series of parameters that control the filter properties.

The next step is bilateral filtering which works on both the spatial and intensity information of the image [25]. It is desirable in our case because it preserves edges in the image. The resulting image has a quality where details within edges look homogeneous.

The fact that the bilateral filter takes into account both spatial and range information allows it to preserve edges. For pixels to be averaged, they must have similar intensity values. Pixels that are not close in intensity are left alone, thus making sure edges do not blur. The areas within these edges then obtain an average intensity. A representative equation of the filter is shown in equation 4.1. A new pixel intensity is calculated based on the intensity values of its neighbors and how similar they are.

$$k(x) = \int_{\infty}^{\infty} \int_{\infty}^{\infty} c(\xi, x) s(\xi, x) \, \mathrm{d}\xi \tag{4.1}$$

The function 'c' assigns spatial weights based on how far a neighboring pixel is from the current pixel. The function 's' returns a value based on how similar in intensity values two pixels are. The symbol ξ represents a neighboring pixel while x is the current pixel being evaluated. The k(x) represents the intensity value given to a particular pixel, x. In the actual implementation, the integral is taken over a neighborhood, instead of a box with edges sized at infinity.

After these two filter techniques are applied, the Matlab implementation of Canny edge detection was applied to the image to create a matrix of logicals that describe the location of the edge. The Canny edge detector has three criteria that will be briefly discussed. There are three requirements for accurate edge detection as outlined by Canny: good detection, good localization, and single detection of a single edge [26]. Good detection means that the algorithm will infrequently detect edges where there are none. Good localization means that the edge location calculated by the algorithm corresponds to the position of the edge in the actual frame. A single detection of a single edge means that only one edge is detected for an edge in the image in cases where edges may not be sharp.

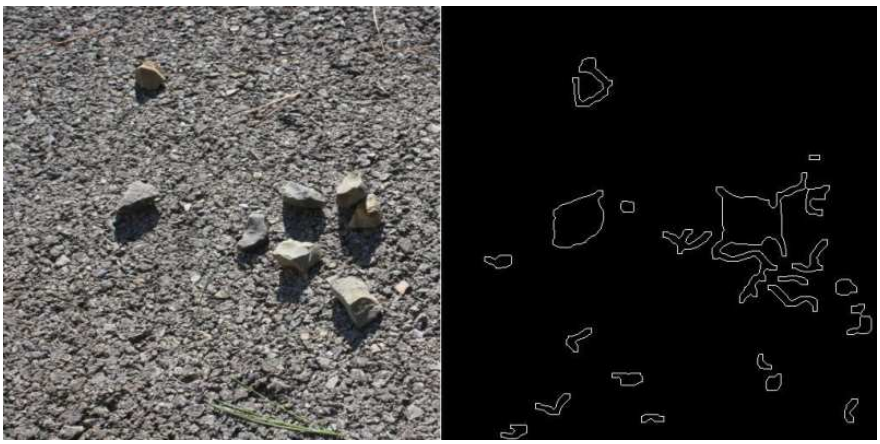


Figure 4.3: Rockfinder demonstration on an image. On the left there are 8 rocks strewn on pavement and on the right is the binary image that is the result of the Rockfinder code.

Rock detection was found to work well in approximating the location of rocks but not getting the correct shape of the rock. There were 15 false positives detected in this particular image shown in Figure 4.3. The algorithm detected 21 blobs in the image while there were only

8 rocks. In this particular case, the algorithm was picking up gravel in the black top. The shapes of the edges detected in the image did not match the shape of the rocks as a result of shadows in the image or the rocks were just too close to each other. The code can be tweaked to work well at particular cases to improve detection of the larger rocks in this case. This algorithm is later adapted in use with depth images taken from a Kinect^{TM}.

Chapter 5

Computer Assisted Sampling

This chapter will explain a method through which the user can attempt to pick up an object using software. Users operating a telemanipulator system will have their attention split between the remote robot and the area around them [27]. In order to allow an operator to respond to dangers immediately near them, a means of picking up rocks automatically was investigated. The method uses a commercial structure from light system to get a point cloud of a scene. The location of a rock relative to the arm is determined. The arm can then be commanded to pick up an object of the operator's choice.

Currently, the user is presented with a top down view of the area of interest and must choose a rock that is in grabbing distance of the arm. The object of interest must also be lined up in a straight line in front of the GSR. This is due to the base stepper motor skipping steps during operation. The computer assist code is restricted to moving the shoulder and elbow joint. The code developed for the computer assist mode can be found in the Appendix H.

5.1 Supervisory Control

Supervisory control (SC) describes the interaction between a human operator issuing commands to a computer which interprets these commands and completes a control loop [28]. SC lies between full automation of a task to manual control with very little computer computation to assist the user in the actions. Two instances of SC are used with the Ground Sampling Robot (GSR), the first is the human operator remotely controlling the tread move-

ment of the GSR. The second is the direct joint movement of the robot arm manipulator. Both take a human's joystick inputs and translates them into motion at the remote site. A single camera relays the movements back to the ground control station.

Situational awareness (SA) is defined in Endsley [22] as:

> ...the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.

Endsley moves on to state that a good SA will improve the chances of task completion. Vozar mentions how making tasks easier helps increase SA [27]. Lumelsky recommends providing an 'autopilot' mode where sensor information is used to aid in manipulation task completion in higher DOF robot arms. This idea is adapted to the USLArm. As a result, a computer assisted mode was prototyped as outlined in Chapter 5. This mode requires a user's input to pick a sample and the computer will retrieve the sample.

5.2 The Kinect™

The Kinect™ is an inexpensive 3D vision system that is commercially available for 150 USD. It has a 640x480 camera resolution and a 320 x 240 depth resolution. The Kinect™ is not affected by surface textures and color changes that may cause errors in a single camera view as seen in Figure 4.3. With Figure 4.3, the Rockfinder algorithm seems to pick up edges related to shadows and gravel as opposed to the rocks in the frame. The application of the Rockfinder algorithm to pure depth data is thought to pick up the sharp contours related to the rocks much easier.

Ideally, the Kinect™ or equivalent sensor will be mounted to the GSR in place of the pan/tilt camera. The Kinect™, as is, would not work on the ground sampling robot due to its size and the sensor's deadzone. It has a 30 inch deadband which is about the size of the ground sampler and the robot arm combined. This fact makes it difficult to mount the Kinect™ to the GSR and obtain valuable information of the robot arm's workspace. For the purposes of prototyping this code, the sensor was mounted 32 inches above the robot arm's work area. As a result of this, a sensor designed for objects closer than 28" would be ideal for this application. It would be desired to mount the 3D sensor directly to GSR.

5.3 Sample Centroid Determination

The user is presented with a point cloud snapshot generated the Kinect™. The user clicks on a rock of interest and these coordinates are recorded. This point is a 3D Cartesian coordinate (x, y, z). These coordinates are used to find the centroid of the rock. The data around the mouse click is isolated to reduce the number of data points processed during filtering and edge finding. Figure 5.1 shows an object of interested isolated and plotted. This region is processed using a centroid finding code.

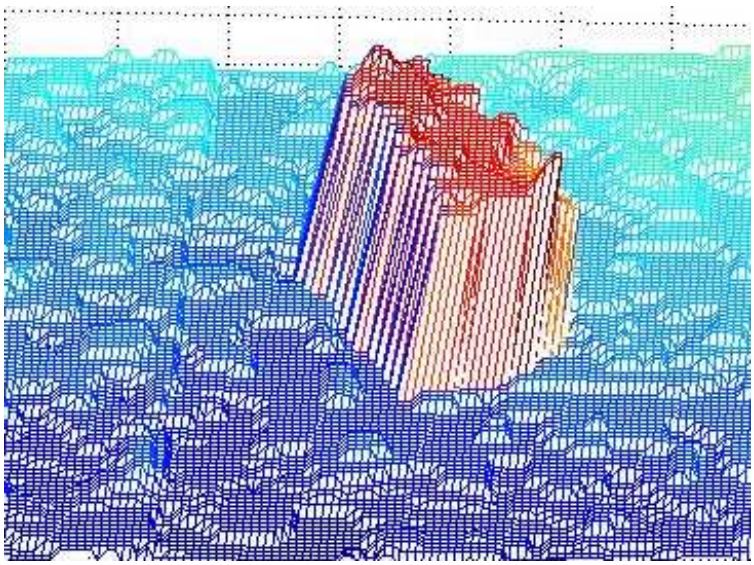


Figure 5.1: This is a plot of a D battery as seen by the Kinect™. It bears very little resemblance to its original cylindrical form.

The depth information seen in Figure 5.1 is transfered into image intensity space in order to be processed by the rock finding code. This is done by finding the range of depths available and mapping them to values between 0 and 1; where 1 corresponds to a maximum found depth value and zero represents a minimum. This produces the grayscale image shown in Figure 5.2(a).

A modified Rockfinder algorithm first discussed in Section 4.3 is used to find the edges of rocks within an image. This particular algorithm uses a denoising process, a bilateral filter and a Canny edge detector. In the Thompson and Castaño's paper, an additional step is added which closes the contours and establish rock boundaries on the frame [23]. In place of

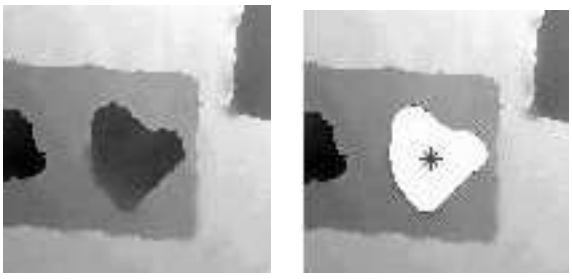
this, a flood fill routine and an erosion operation in tandem with a generous set of thresholds on the Canny filter help isolate a single closed loop in the image. This takes advantage of the fact that a human operator is present and is able to guide the rock finding process. The asssumption is that the point the user provides lies within the area of the rock. The 2D centroid is then determined using a set of Matlab commands associated with 'regionprops' command. Figure 5.3 shows the final result, a 2D centroid location and a matrix of logicals defining the region of the rock.

This centroid is relative to the image coordinate frame and not to the world coordinate frame centered on the Kinect^{TM} sensor. In order to transform between the two, the inverse of the Kinect^{TM} projection matrix needs to be premultiplied by the image coordinates to yield the 3D centroid in the world coordinate frame as illustrated in Equation 5.1.

$$R^{-1} f \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \tag{5.1}$$

Where 'f' is the scale factor; the Z depth at the mouse click was used for this value. 'R' is the camera perspective matrix. For the Kinect^{TM} the matrix is:

$$R = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 600 & 0 & 320 \\ 0 & 600 & 240 \\ 0 & 0 & 1 \end{bmatrix}$$



(a) (b)

Figure 5.2: Figure (a) shows a depth map of a region of interest. Figure (b) shows the processed image with the centroid of the rock determined.

The centroid in the 'Z' direction needs to be determined before the arm can reach out for

points. The centroid in 'Z' is found by taking the average of the top surface of the rock and averaging these values with points near the rock that correspond to the location of the underlying surface.

Once the centroid has been located in the 3D Cartesian camera coordinates, this information needs to be transfered over to the arm coordinate system using a transformation. In the case of the test setup used, this transform is a simple rotation about the Kinect^{TM} 'X' axis by 180 degrees and a translation to the arm's base. As would be expected, this homogeneous transform will change with any alterations in the Kinect^{TM} location or the arm location.

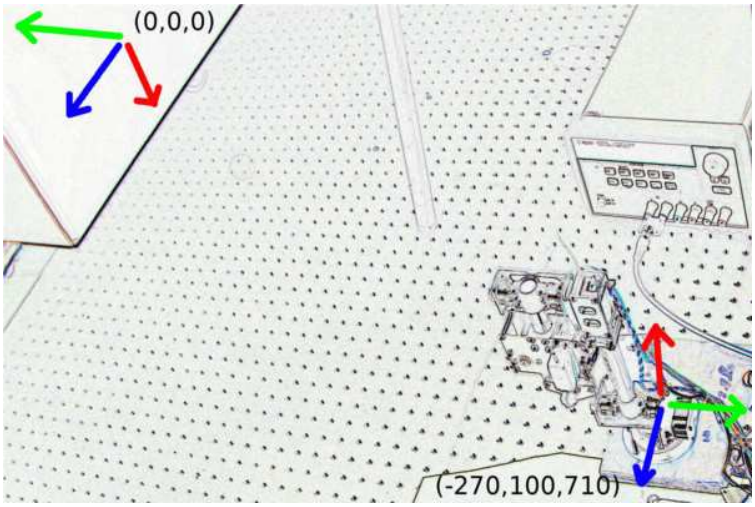


Figure 5.3: This illustrates the necessary rotation and translation necessary to convert from sensor coordinate frame to arm coordinates. The two coordinate systems are represented on the image as 2 sets of 3 arrows in different colors. The red arrow represent the 'Z' axis, the green arrow represents the 'Y' axis, and the blue arrow represents the 'X' axis.

Now that the coordinates of the rock's centroid is in the arm coordinate frame, a position can be commanded.

5.4 Robot End Effector Positioning for Grasp

In order to command the arm to move to a position in space, the inverse kinematics needs to be determined. Since there is no feedback on the base joint, it will be ignored and the 2 DOF case will be examined. This relies on moving the shoulder and elbow joint as shown

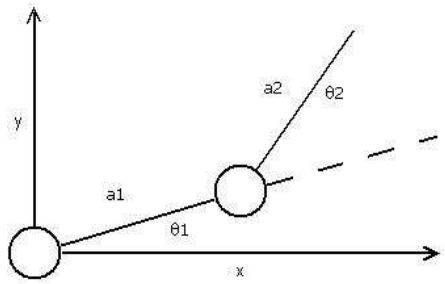


Figure 5.4: Restricted 2 linkage schematic.

in Figure 5.4. Given a desired position, x_d and y_d, the angles necessary for placing the end effector can be determined [18].

$$\cos\theta_2 = \frac{x_d^2 + y_d^2 - a_1^2 - a_2^2}{2a_1a_2} = D \tag{5.2}$$

Representing the relationship between the angle θ_2 and D with the Pythagorean theorem:

$$\sin\theta_2 = \pm\sqrt{1 - D^2}$$

Dividing the two representations, a relationship is established that yields two possible solutions.

$$\theta_2 = \tan^{-1}\frac{\pm\sqrt{1 - D^2}}{D} \tag{5.3}$$

From here the Cartesian coordinate for where the end of link 1 (x_1, x_2) is located can be found with the following:

$$x_1 = a_1\sin\theta_1 = x_d - a_2\sin\theta_2, y_2 = a_1\cos\theta_1 = y_d - a_2\cos\theta_2$$

Dividing the two equations and solving for θ_1:

$$\theta_1 = \tan^{-1}\frac{y_d - a_2\sin\theta_2}{x_d - a_2\cos\theta_2} \tag{5.4}$$

After the goal location is determined in joint angle space (θ_1, θ_2), the robot is sent to the

home location and moves to the goal location with the end effector open. The end effector then closes at the goal location and the rock is lifted into the air. From here, the rock can be dropped into the SCV by moving the arm to a mechanical stop at an extreme of the base joint's motion. There is no feedback in these motions as it is assumed that user will step in and assume manual control or simply attempt to pick up a more manageable rock.

5.5 Servo Position Analysis

Servos are positioned by sending a PWM signal out to the motor. The servo has the necessary electronics to interpret this signal and position the output shaft of the motor to a particular orientation. This orientation has a resolution that is different for each motor. The shoulder servo has a resolution of $0.17\frac{^\circ}{\mu S}$ and the elbow has a resolution of $0.1\frac{^\circ}{\mu S}$.

The positional resolution can be determined using the forward kinematics for a planar 2 DOF case shown in Equation 5.5 and 5.6.

$$x = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2) \tag{5.5}$$

$$y = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2) \tag{5.6}$$

The effect of the resolution of the servos on the Cartesian coordinates can be represented by 5.7 and 5.8.

$$\Delta x = (-a_1 \sin\theta_1 - a_2 \sin(\theta_1 + \theta_2))\Delta\theta_1 - (a_2 \sin(\theta_1 + \theta_2))\Delta\theta_2 \tag{5.7}$$

$$\Delta y = (a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2))\Delta\theta_1 + (a_2 \cos(\theta_1 + \theta_2))\Delta\theta_2 \tag{5.8}$$

The resolutions are a function θ_1 and θ_2. The resulting 'x' and 'y' resolution is a function of the joint angles. The error as a result of the two resolutions ('E') can be represented by a Euclidean norm shown in Equation 5.9 and plotted for theta values in the domain of

movement in Figure 5.5.

$$E = \sqrt{\Delta\theta_1^2 + \Delta\theta_2^2} \tag{5.9}$$

The maximum hypotenuse that is formed by the two errors was found to be 0.05". This value is tolerable, as the robot grippers are large enough to compensate for this error. The servo errors are just a single part of the total position error.

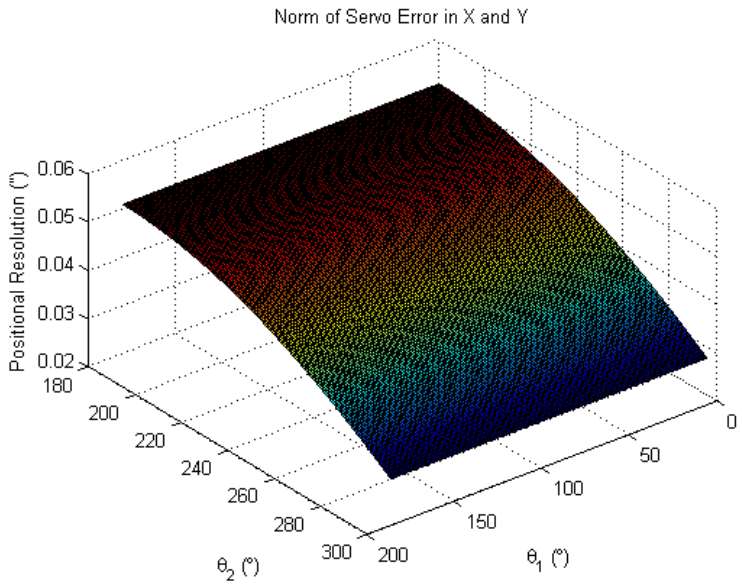


Figure 5.5: 'E' over a domain of robot motion. The maximum value was found to be 0.05". The error is greatest when the robot is fully unfolded.

When error in link length measurements are taken into account, the max error moves up to 0.26" which is 5 times the error from the servo resolution alone. This is assuming that the error in the link measurement is 1/8". The error is small enough where the user can make adjustments to the position based on the camera position seen on screen. This error does pose a problem during autonomous operation as testing in Section 5.7 shows.

5.6 Centroid Detection Calibration

A calibration of the centroid detection algorithm along with the Kinect^{TM} was performed using objects in measured locations relative to the Kinect^{TM}. There is an offset that was observed that seemed to increase the farther away the sample was from the robot arm. The

goals of the test are to determine the degree noise played a role in centroid detection and to characterize any error that may be occurring in sample localization.

5.6.1 Centroid Detection Test Setup

A calibration involves measuring using at least two methods, one with unknown values and error and another with known values and error. In this case, the output of the rock localization algorithm using a Kinect^{TM} point cloud was compared to known measured values of a 3D Cartesian coordinate of an object location on an optical table.

The materials used in the calibration are a Kinect^{TM}, M6 bolts 25 mm, 1" outer diameter flat washers, paper, an optical table, calipers, measuring tape, and a three inch diameter steel cylinder. The Kinect^{TM} in conjunction with the rock localization algorithm was used to estimate the location of objects of interest. The objects of interest were M6 bolts with flat washers glued to the bolt heads. This allowed for an easy to assemble, reproducable test object. Paper was glued to the tops of these washers to elimate specular reflection which is known to create errors in the depth readings from the Kinect^{TM}. An illustration of the test pieces can be seen in Figure 5.6.

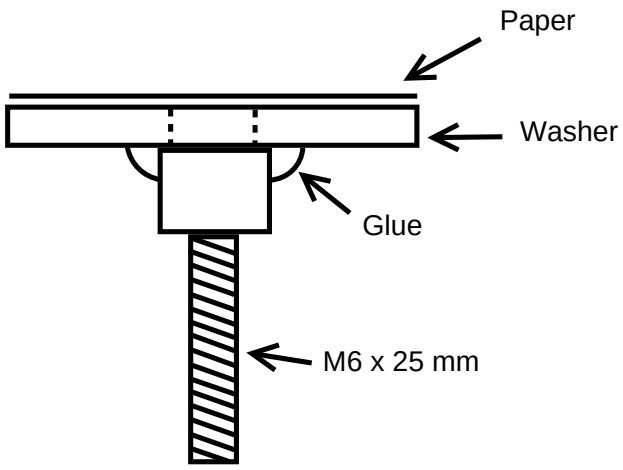


Figure 5.6: Bolt washer assembly used for Kinect^{TM} calibration.

The centroid detection code's output was calibrated against measured locations on the optical table. The bolt and washer assemblies were screwed into the optical table with a grid bolt pattern. This allows for Kinect^{TM} calibration in the X and Y direction as the holes are

placed 25 ±.13 mm in rows and columns as illustrated in Figure 5.7. In the Z direction, two measurements are used to calibrate the distance from the sensor out to the centroid of the object of interest. The subtraction of two measurements is required to get the distance in Z relative to the Kinect™. The height that the Kinect™ was mounted was measured using a measuring tape as 790 ±3 mm. A set of calipers were used to measure the height of the bolt washer assembly and this value was divided by two to get the distance from the table to the centroid of the object of interest. The accuracy of the set of calipers were ±0.01 mm. In this Z measurement, the confidence interval of the measuring tape dominate at ±3 mm. An illustration of this measurement is shown in Figure 5.8 This interval is sufficient as an error of a few millimeters is well within the size of the gripper. An illustration of the gripper along with an estimated maximum error of 10 mm from the center of the gripper is proposed in Figure 5.9.

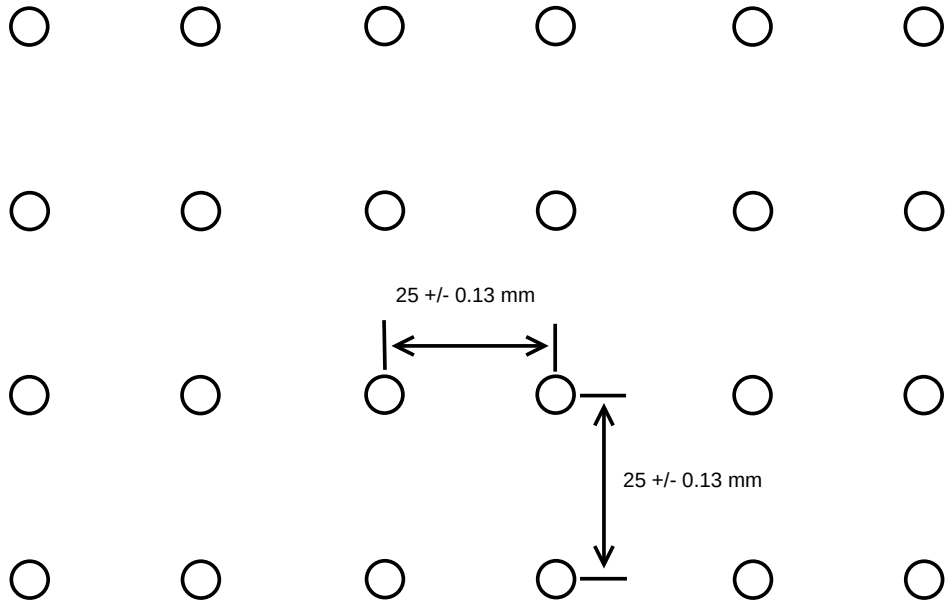


Figure 5.7: Optical table hole pattern.

In order to line up the Kinect™ coordinate frame with that of the table's, the Kinect™ needed to moved until its XY origin and orientation matched up with the one desired on the table. The combined motion involved is shown in Figure 5.10. The X and Y axis of the Kinect™ are lined up with a bolt on the table with a 3 inch diameter steel cylinder. The Kinect™ was moved parallel to the table's surface until the cylinder's center matched the projection of the X and Y origin onto the table. The Kinect™ was also rotated about it's

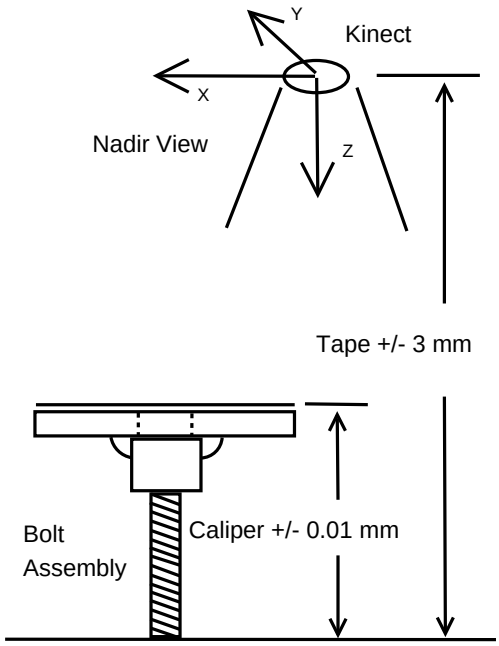


Figure 5.8: Measurements used to calibrate the output of the centroid detection algorithm in the Z direction.

Z axis until two bolts were lined up with the X axis on the Kinect^{TM}. This allows for the coordinate frame desired on the table to match the Kinect^{TM} coordinate frame.

5.6.2 Centroid Detection Test Procedure

From simply looking at a single frame of the Kinect^{TM}, it is apparent that there is a degree of distortion that occurs as seen in Figure 5.1. This figure has an image of a D cell battery resting on the optical table mentioned in Section 5.6.1. Two types of error can be observed from this image, random noise and systematic error. The battery has a cylindrical shape that is shown to have spikes at random locations on the battery's surface. The surface also has a wrinkled appearance, which is characteristic of noise. This noise should have very little effect on the centroid calculation as the algorithm averages a number of points based on rock size. There is also a lean that is observable from in the image that could be the explanation for the bias seen in the calibration test results.

In order to rule out noise as significant source of error in the centroid calibration code, it was

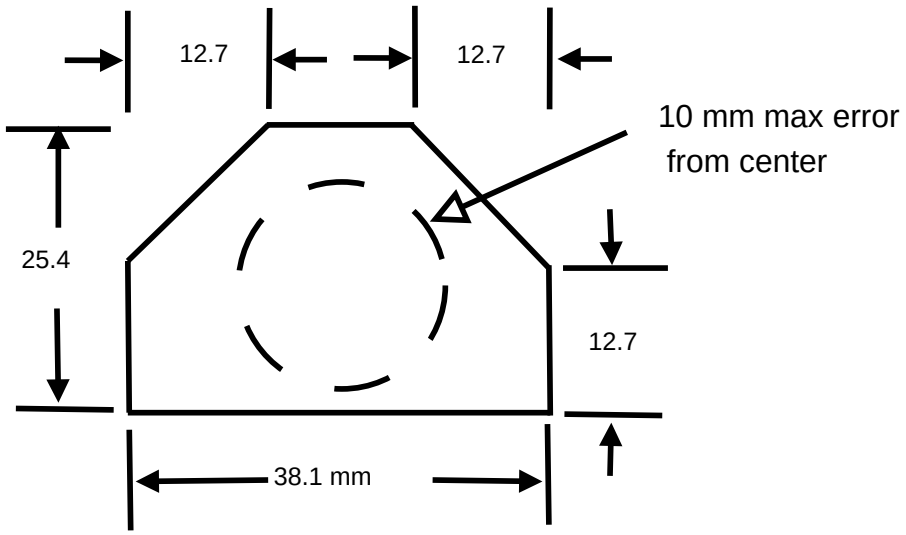


Figure 5.9: Estimated maximum gripper error. A target centroid can be grasped within the dotted line. This assumes that the target is smaller than the gripper as a larger target can be grabbed with higher errors than a smaller one.

necessary to see how much the centroid estimate varied over a series of measurements, over multiple frames of the Kinect^{TM}. For a confidence interval of ± 1.5 mm in any direction at 95% certainty, it was found that a single measurement was sufficient. This was determined after taking 30 centroid measurements of 3 bolts on the optical table surface. Based on the highest standard deviance in position, the confidence interval was generated.

To determine the systematic error that is seen in the centroid calculation, a rectangle with dimensions 375 mm by 225 mm on the table was isolated and the centroid detection code was tested at 24 locations in X and Y and 2 for Z for a total of 48 points. The grid used is shown in 5.11. Z was varied by simply loosening or tightening the bolts in the table. Two values tested in Z are where the washer top was 24 mm above the table and about 15 mm above the table.

5.6.3 Centroid Detection Test Results

The centroid detection calibration reveals an offset that gets larger the farther away from the origin an object is located as shown by Figure 5.12. This systematic error can be corrected for in the X and Y directions by using the linear relationships established.

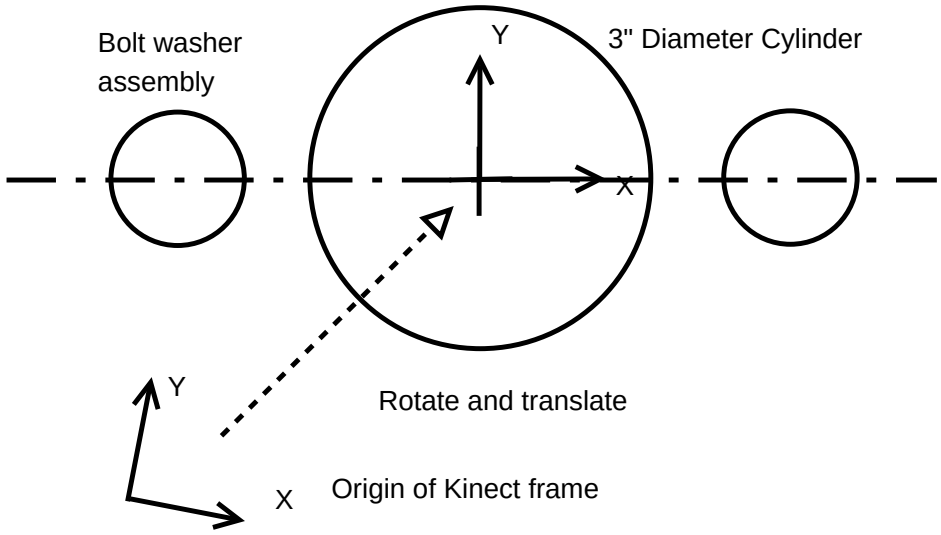


Figure 5.10: Matching the Kinect^{TM} coordinate frame to the table's coordinate frame.

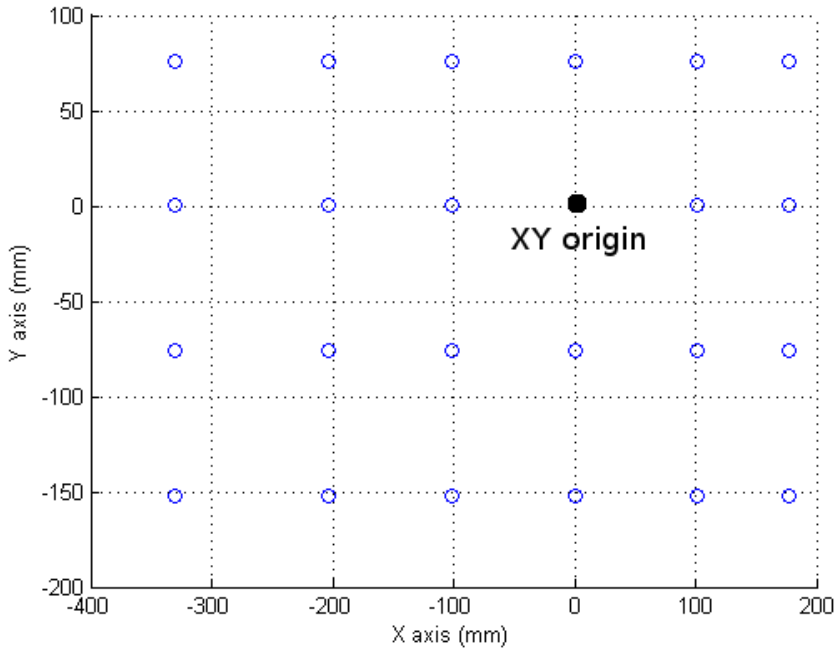


Figure 5.11: Calibration test points on optical table. Each location was tested at two bolt-washer heights.

The further away from the origin of the table, the error between the Kinect^{TM} centroid estimate and the measured position increases. A correction factor in X and Y can be established based on the centroid code output along with the corresponding error seen at that particular X coordinate. A plot of X position output by the centroid code versus the error observed is seen in Figures 5.13 and 5.14.

Based on the coordinates returned by the centroid detection algorithm, a correction factor can be applied which predicts the amount of error that will be seen in X and Y shown in equation 5.10 and 5.11. C_x and C_y represent the correction factor and K_x and K_y represent the centroid estimate given by the centroid calculation code. This predicted error is then added to K_x and K_y.

$$C_x = 0.146(K_x) + 2.12 \tag{5.10}$$

$$C_y = 0.146(K_y) - 3.96 \tag{5.11}$$

An illustration of the effectiveness of this approach can be shown in two methods. The first involves taking the sum of the norm-2 errors over the 48 data points produces a value of 1200 mm originally. This value divided over the 48 data points yields an average error of 25 mm per location. When the correction factor is applied, this sum of errors drops down to 250 mm or 5.2 mm per location. Secondly, a quantitative illustration of the improvement in grip accuracy can be seen in Figures 5.15 and 5.16. The corrected commanded positions place closer to the center of the object of interest. This leads to more accurate object centroid estimation.

5.7 Arm Movement Precision Testing

The goal of the repeatability testing of the robot arm is to see if the placement errors of the robot arm will affect rock picking performance. This test is done independently of the Kinect centroid estimates. If the error found in this section exceeds that shown in Figure 5.9 at 10 mm, then the arm's error would be too large to be effective in autonomous rock picking operations. The experiment ultimately helped to eliminate the robot arm as a significant source of error. Since the overall autonomous mode is restricted to 2 DOF, the test will be

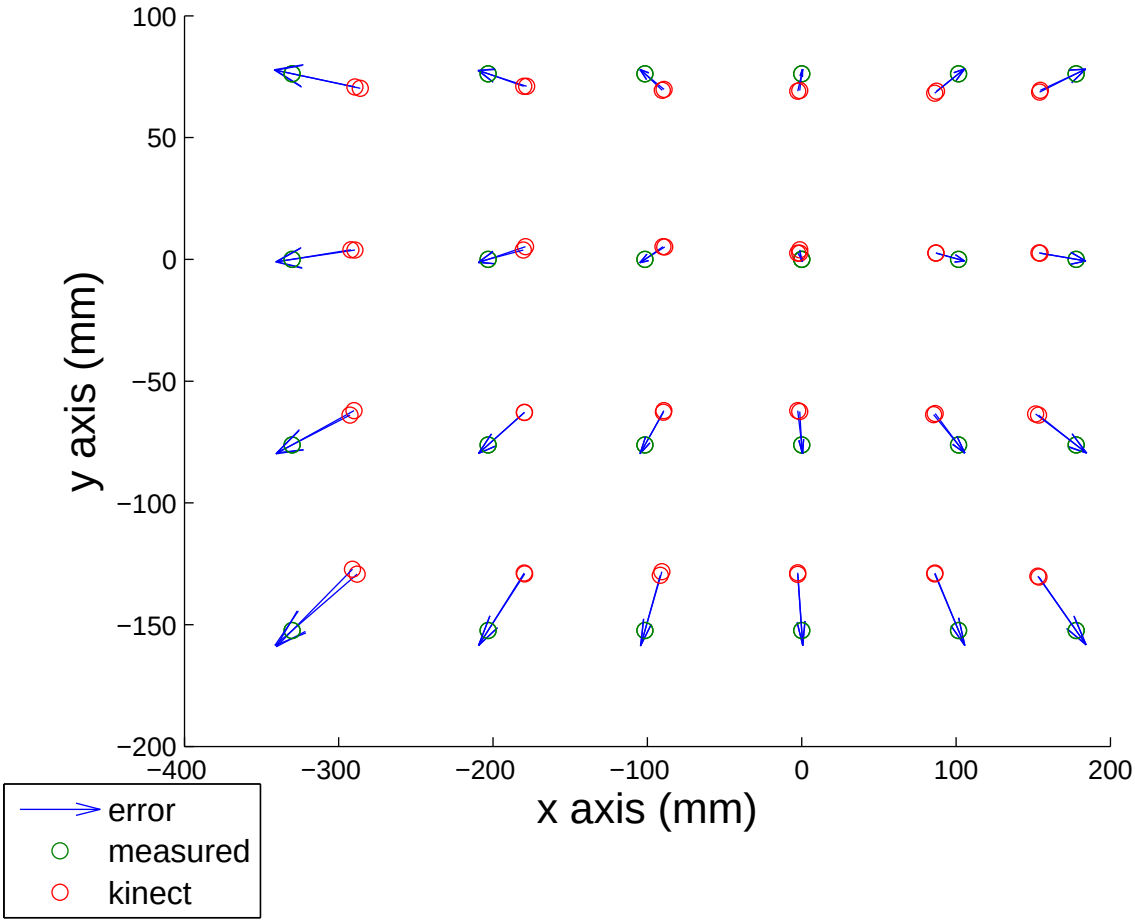


Figure 5.12: Kinect^{TM} centroid plot versus expected values.

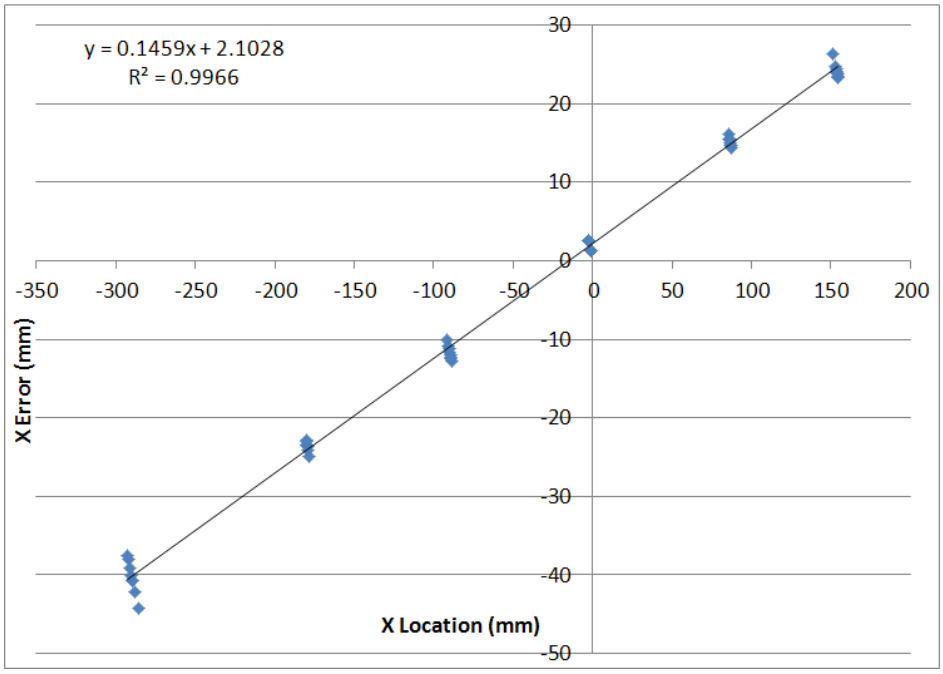


Figure 5.13: Kinect^{TM} centroid X values versus error observed.

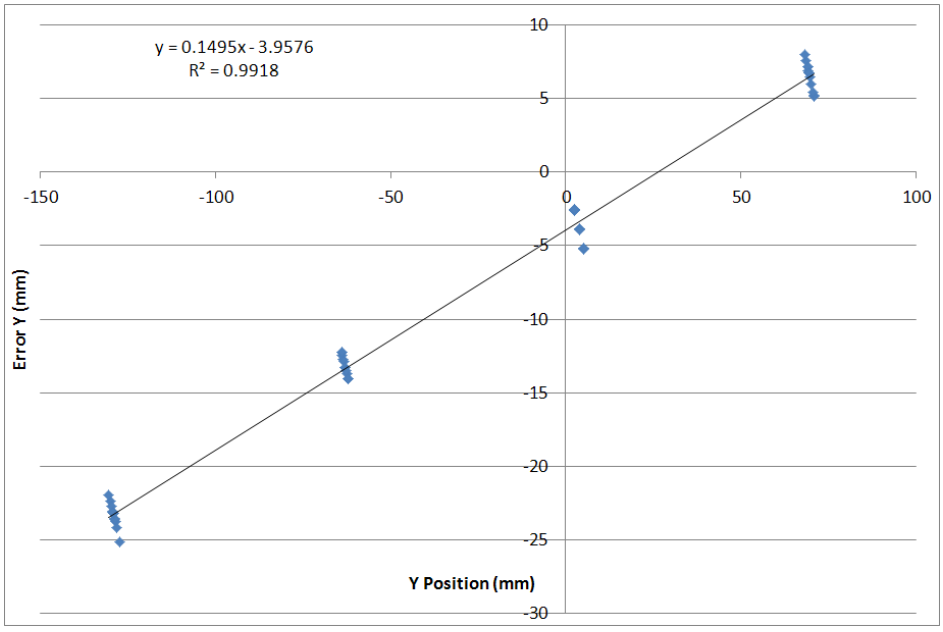
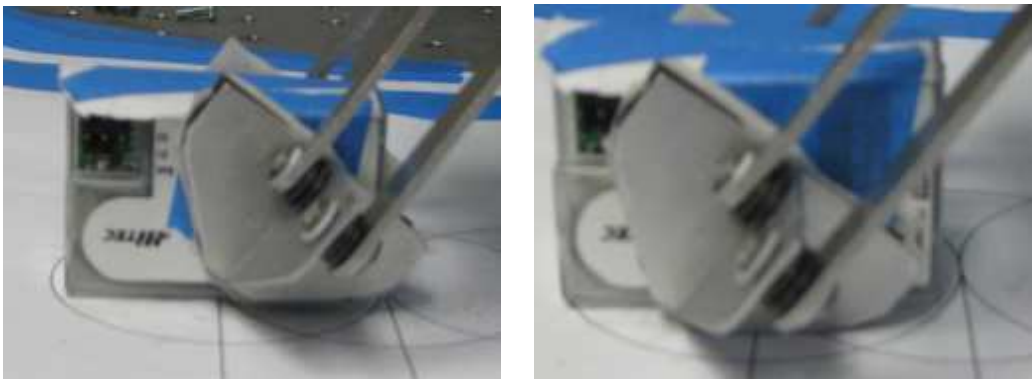


Figure 5.14: Kinect^{TM} centroid Y values versus error observed



(a) (b)

Figure 5.15: End effector grabbing a block 8 inches away from front lip of the sampling tray. Figure (a) shows a grab without correction while (b) shows a grab with the correction factor added. The grip has moved towards the center of the object of interest.

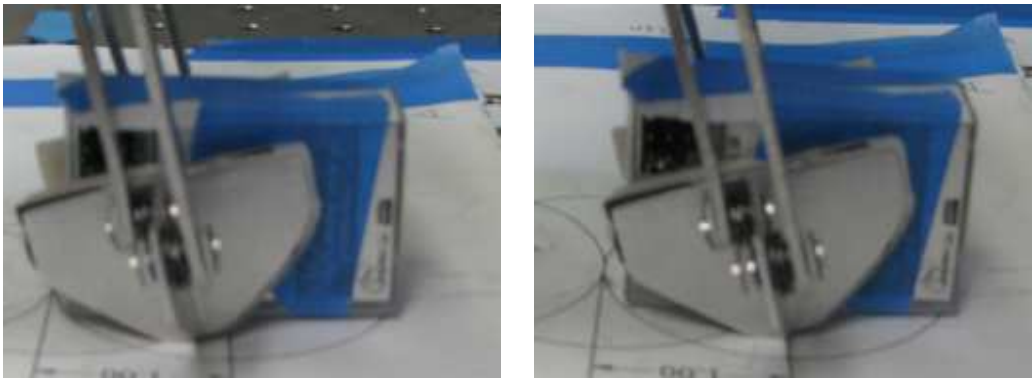
performed using 2D Cartesian coordinates with an origin located at the the base of the robot arm. A total of 4 sets of movements were tested and are described in Section 5.7.1.

A single measurement from the table top to the end effector is used to describe the positioning error. Only one measurement of this 2 DOF system is necessary as the joint error affects both values of X and Y.

5.7.1 Arm Movement Test Procedure

The first and second set of movements tested involved moving the robot arm's end effector in the vertical direction (Y). A home location was picked and the arm's position was measured with a square with accuracy of $\pm 1/16$". The end effector was then moved to an away position and the square was used to measure it's height from the table. An inch displacement in the end effector can then be compared to an inch on the square. A number of displacements in the Y direction were tested -50, -25, 25, 50, 75, 100 mm. A schematic of this movement is shown in Figure 5.17. This vertical movement was tested at X=101 mm (4 in) and X=202 mm (8 in).

The third and fourth movements tested varied the coordinates parallel to the table or in the X direction. These movements tested if there was noticable sag in the arm as the arm outstretched. A home position was picked and the arm was moved to different horizontal locations. Measurements were taken at the home location and a measurement taken at the



(a) (b)

Figure 5.16: End effector grabbing a block 50 mm away from front lip of the sampling tray. Figure (a) shows a grab without correction while (b) shows a grab with a correction factor added. The corrected grip position is closer to the center of the object

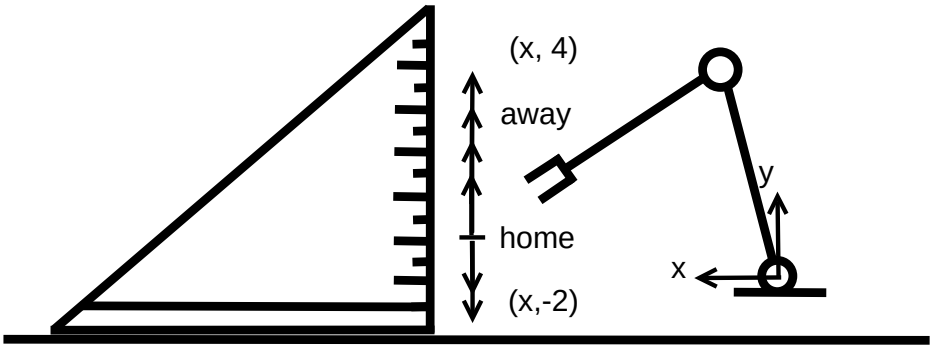


Figure 5.17: Set of movements used to test movement precision. Movements were vertical restricted at X=101 mm (4") and X=202 mm (8").

away position. A schematic of the movement can be seen in Figure 5.18. The arm movements were held at Y=-50 mm (-1") and Y= 0 mm.

5.7.2 Arm Movement Test Results

The first and second test showed a maximum error of 3 mm (1/8"). An error was only present in 2 out of twelve displacement measurements. The rest of the movements were accurate to within the accuracy of the square at ±1/16". The maximum error is within the maximum error described in Figure 5.9 of 10 mm.

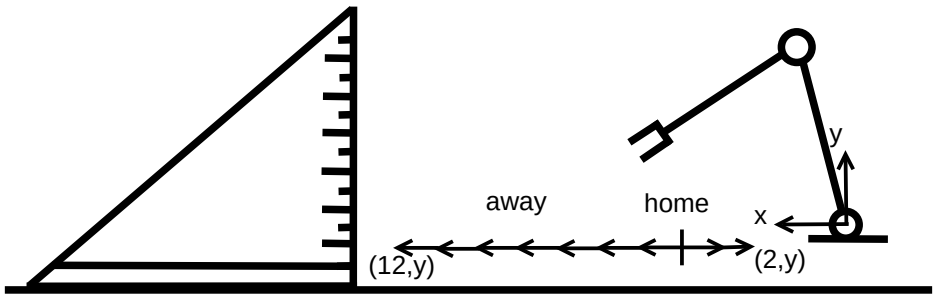


Figure 5.18: Set of movements used to test movement precision. Movements were vertical restricted at Y=50 mm (-2") and Y=0 mm.

The arm movement tests along the X direction yield a low error that increases at the extremes of the motion. This is visible in Figure 5.19 and 5.20. This sharp increase is a direct result of moving beyond the commandable range of the robot arm. The shoulder is able to adjust to the desired angle, but the elbow joint remains unmoved. This lack of motion is only a concern when the arm is commanded to move past 266 mm (10.5").

Overall, the amount of error was deemed as not significant enough for concern. Again, the error in these movements did not exceed 10 mm with a maximum error seen at 6 mm. The grippers of the end effector are large enough to compensate for this error.

5.8 Combined Error

Ideally, the computer assisted mode will work precisely when picking up objects of interest. Errors from the centroid estimate and the arm placement converge to form a total error. This total error represents the total end effector placement error. The error is going to be represented as a single bias, as the errors in X, Y, and Z are averaged and normed to create a scalar, B_c. This average bias has a precision associated with it based on how much the centroid values vary, P_c. The arm placement error is represented by a bias, B_a, and precision of, P_a. Precision is determined at 95% confidence. Equation 5.12 shows the total end effector positioning error given the source bias limit, B, and the measurement precision index, P. The bias limit (equation 5.13) is represented as the Euclidean norm of the bias from the centroid calculation and the bias from arm positioning. Likewise, the precision index (equation 5.14) is the Euclidian norm of the precision of the centroid estimate and the precision of the arm

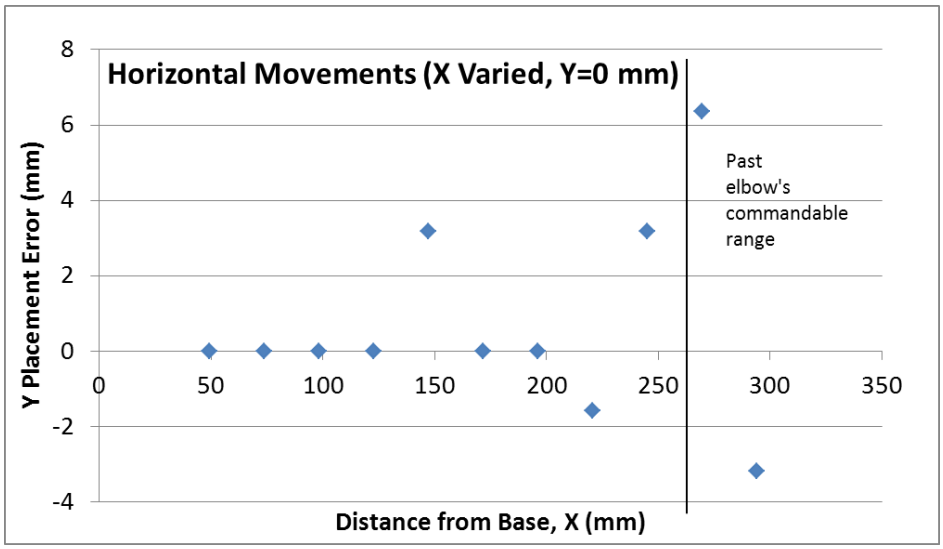


Figure 5.19: Y placement error at Y=0 mm. The average error below X=250 mm is 0.72 mm.

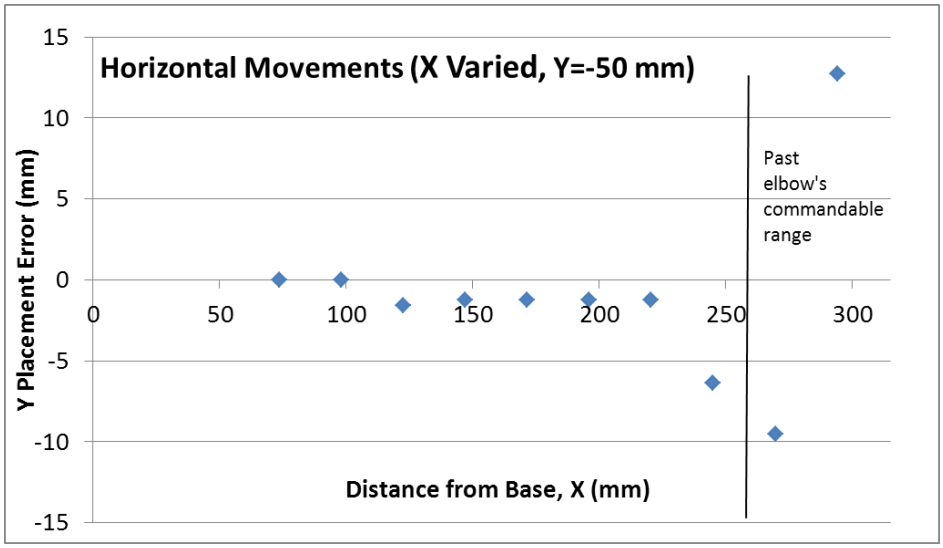


Figure 5.20: Y placement error at Y=-50 mm. The average error below X=250 mm is 1.6 mm.

positioning. These values, B and P, are displayed in Table 5.1.

$$E_t = \pm\sqrt{B^2 + P^2} \tag{5.12}$$

$$B = \sqrt{B_c^2 + B_a^2} \tag{5.13}$$

$$P = \sqrt{P_c^2 + P_a^2} \tag{5.14}$$

Table 5.1: Precisions and biases referred to in equations 5.13 and 5.14.

	Centroid (mm)	Arm (mm)
Bias	5.4	1.5
Precision	1.5	1.6
Euler Norm	±5.6	±2.2

Evaluating equation 5.12 by taking the Euler norm of the final row of Table 5.1 yields a total placement error of ±6 mm at 95% confidence. This is within the 10 mm maximum proposed previously, based on the gripper size and a small target proposed in Section 5.6.1. This error is valid for the movement range of the robot arm and when the Kinect™ is mounted 0.79 m above the workspace. If the Kinect is moved to a new position, the errors will need to be identified again. The errors associated with the arm will remain the same.

5.9 Gripper Test

The effectiveness of the grippers will be shown in this section. The success rate gives a metric that characterises the error for the grippers alone. As the objective is to pick up samples, a measure of the success rate yields the most valuable information in regards to how well the grippers are working. It was found that success is highly dependent on the shape of the sample.

Three sets of tests were carried out on 3 different samples. The first two were rocks of different shapes and the third sample was a rectangular prism of aluminum with dimensions

45 mm x 25 mm x 45 mm. Grip success is highly dependent on where the end effector applies force on the rock. Failure was defined as a sample slipping out of the gripper.

5.9.1 Setup and Procedure

To simulate conditions that a user might see when the system is deployed, sample shape and distance away from the front of the robot were varied. Shape was varied with two rocks of very different shapes referred to as sample 'A', 'B', and 'C'. Rock 'A' is a flat rock that has large flat surfaces. Rock 'B' is a small stout rock with angled surfaces. Sample 'C' is a block of aluminum with dimensions 45 mm x 25 mm x 45 mm. The distance away from the sample collection platform was varied at 1" intervals between 2 3/8" and 8 3/8". A single sample involved placing either rock 'A', 'B', or 'C' in 1 of 7 locations, commanding the robot to pick up the rock and lift it into the air. If the rock fell out of the robot grip during motion, the trial was considered a failure. If the robot arm held the rock up, then the trial was considered a success. These trials were carried out 15 times for rock 'A', 13 times for rock 'B', and 12 times for sample 'C'.

5.9.2 Test results

Rock 'A' was picked up 13 out of 15 attempts for a success rate of 86%. Rock 'B' had 10 successes out of 13 for a success rate of 77%. Rock 'C' had 12 successes out of 12 for a 100% success rate. Overall, the aluminum block was much easier to secure than the grippers simply due to shape. Sample 'C' was pointed near the top of the sample, which deflected the grippers in a way that disallowed collection in certain instances. Likewise, the angles surfaces of rock 'B' made it difficult to grasp at certain gripper orientations.

This test illustrates that success of the sampling mechanism is highly dependent on the shape of the rock. This test shows that the centroid of an object may not always be the best place to grab depending on the shape of the object of interest. If a sample is too difficult for the computer to pick up, the user will have to assume manual control and manipulate the sample.

Chapter 6

Conclusion and Improvements

The work presented in this thesis increases the functionality of the the GSR by allowing the operator to do more than simply observe a disaster area but also interact with objects in the environment. A computer assisted mode was prototyped to make the task of picking up a sample easier. With the right sensor, this will reduce the amount of effort and training required to use the system.

6.1 Summary of Work

A remote sampling device was designed, built, and tested. A single click user interface was created to make the use of the sampling mechanism easier. This device will be used in missions at the USL and lays the ground work for manipulators used in conjunction with a UGV and a UAV. It allows for manipulation in environments without endangering human beings due to the robot arm's BLOS capabilities.

In order to fit on to the RMAX, the USLArm was built within a 230 mm x 265 mm x 90 mm volume when collapsed for. To reach well into the camera view of the operator, the reach is 260 mm fully outstretched. The final mass of the system is 1.5 kg. The USLArm is capable of lifting samples that are 250 grams and 44 mm in diameter.

A 3D USLArm training tool is presented. Different means of displaying robot pose information and sensor data were also examined which led directly into the recommendation of using a 3D sensor system. It is difficult for a user to correlate the output of point sensors

with a single camera feed of the scene. Point sensors will also not work well when it comes to determining the size of a sample which dictates how the sampler will grab the object of interest.

Additionally, a computer assisted mode that allows a user to pick up an object of interest autonomously was created. The assist relies on computer vision techniques to localize the rock and then traditional kinematic analysis to position the robot arm. From here, the user provides the necessary feedback to drop the sample onto a storage container on the GSR.

The precision of movement of the arm was tested and found to have a max error that are well within the 10 mm limit placed, at 6 mm. A calibration on the rock localization code was performed and found to have low noise but a systematic error that grew the farther away from the XY origin a point was. This error was predicted and corrected for and found to yield grip locations closer to the center of the object of interest.

6.2 Proposed Improvements

Improvements to the chunk sampling system are established below:

To allow for greater collapsibility and reduce the size of the end effector as a whole a low profile servo motor can be used in the end effector. Low profile servos have a height of about 25 mm while the current motor has a height of 35 mm.

The amount of metal used on the arm can be reduced by examining locations on the arm seen as greatly reinforced and determing a slimmer method of accomplishing the same tasks. One such application would reduce the weight of the elbow joint, the two plates that encase the elbow motor can be reduced in thickness. A stress analysis of the bolting should be done to ensure that the new plates can take the loading necessary. Reducing the material thickness by half will decrease the weight of the elbow by 12 grams.

Using rectangular tubing instead of round tubing may reduce the number of parts and manufacturing a significant amount. The servo horn and clamping bore or set screw hubs can be fixed directly to the tubing. This completely removes the need for tube clamps or the modified T-bar pieces that transfer the loading to the link material.

With lab use, the 3/8” shoulder bolt at the shoulder joint has started to swivel around in the bearing. This creates a small amount of slop in the mechanism. This issue can be remedied

by increasing the wall thickness of the housing to give the bolt greater support.

The sample collection width can be increased by modifying the end effector to accommodate larger rocks. Pushing the stand offs on the front of the end effector or cutting grooves for the linkages will increase the width to which the end effector can be opened. Care must be taken to not make this too wide as it may cause a teleoperator to overshoot rock positions.

To reduce the chance of long term Galvanic corrosion of the device during storage, the design needs to be altered to take only aluminum components. Currently, the arm has components in direct contact with aluminum and steel bolts in contact with aluminum as well. This was done as it would be unreasonable to replace a stock of bolts and nuts with aluminum parts for a single project. A future iteration of the device would need to maintain the same metal throughout the design or simply ensure that future contacts are chemically nonreactive for the operating environment desired.

A more sophisticated command packet structure is desirable to command both the robot treads and the arm. The arm should have the option to be operated in relative or absolute commands without having the user reprogram the Arduino so that they may switch between the two modes seamlessly.

The Kinect^{TM} will need to be replaced with a 3D sensor mounted directly to the GSR in place of the pan tilt camera currently in place. The 3D sensor will help the robot navigate and help localize samples for retrieval. The Kinect^{TM} does not have the capabilities to be used directly in our application as there is a 28" deadband where the robot arm workspace is located.

Since a nadir 3D sensor viewpoint will not longer be used. The computer assist code will need to be modified to take into account a long shadow effect that will be observable from mounting a 3D sensor on the GSR. The code will need to ignore the shadow when calculating the centroid for an object of interest.

For full 3 DOF autonomouse operation of the robot arm, the base stepper motor skipping steps needs to be addressed. If an encoder is put on the base slewing bearing or on the stepper motor output shaft, one can estimate the rotation of the arm and extend the computer assist code to work in 3 DOF.

The improvements listed will enhance the chunk sampling system. Overall, the robot arm extends the capablities of the GSR by providing a means for an operator to pick up objects

and store them for transit. This task is made easier using a semi-autonomous method of locating and picking rocks provided an appropriate sensor is found and implemented.

Bibliography

[1] Jessie Y. C. Chen, Ellen C. Haas, and Michael J. Barnes. Human Performance Issues and User Interface Design for Teleoperated Robots. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(6):1231–1245, November 2007.

[2] Michael R Couch. *Pneumatic Particulate Collection System for an Unmanned Ground Sampling Robot*. PhD thesis, Virginia Polytechnic Institute and State University, 2010.

[3] Jr. et al. Salisbury. US Patent No. 5046375, 1991.

[4] Rainer Bischoff. HERMES A Humanoid Mobile Manipulator for Service Tasks Institute of Measurement Science. *New Design*, (December):1–8, 1997.

[5] G Hirzinger, B Brunner, K Landzettel, N Sporer, J Butterfass, and M Schedl. Space robotics–DLR's telerobotic concepts, lightweight arms and articulated hands. *Autonomous robots*, 14(2-3):127–45, 2003.

[6] J.L. Novak, J.T. Feddema, N.E. Miner, and S.A. Stansfield. Collision Avoidance During Teleoperation Using Whole Arm Proximity Sensors Coupled to a Virtual Environment. *SPIE*, 2057(May 2011):2–9.

[7] Frank Daerden and Dirk Lefeber. Pneumatic Artificial Muscles : actuators for robotics and automation.

[8] Hari Das, X. Bao, Y. Bar-Cohen, R Bonitz, R. Lindemann, M. Maimone, I. Nesnas, and C. Voorhees. Robot manipulator technologies for planetary exploration. *Proceedings of SPIE*, (3668):175–182, 1999.

[9] John V Draper. Human Factors in Telemanipulation: Perspectives from the Oak Ridge National Laboratory Experience. *SPIE*, 2057(May 2011):162–174, 2008.

[10] Antal K. Bejczy. Teleoperation, telerobotics and telepresence: A progress report. Technical Report 2, California Institute of Technology, Pasadena, California.

[11] Paul S. Schenker, E.T. Baumgartner, R.A. Lindemann, H. Aghazarian, D.Q. Zhu, and A.J. Ganino. New planetary rovers for long-range Mars science and sample return. *Proceedings of SPIE*, 3522(May 2011):2–15, 1998.

[12] Paul S. Schenker. Composite manipulator utilizing rotary piezoelectric motors: new robotic technologies for Mars in-situ planetary science. *Proceedings of SPIE*, pages 918–926, 1997.

[13] Eric Byler. Design and Control of Ultralight Manipulators for Interplanetary Exploration. *SPIE*, 1387(May 2011):313–327, 1990.

[14] Brian M. Yamauchi. PackBot: a versatile platform for military robotics. *Proceedings of SPIE*, pages 228–237, 2004.

[15] Erik Schoenfeld, Lawrence Parrington, and Stephan von Muehlen. Door breaching robotic manipulator. *Proceedings of SPIE*, 6962(May 2011):69620S–69620S–10, 2008.

[16] C. Fagerer, Dirk Dickmanns, and Ernst D. Dickmanns. Visual grasping with long delay time of a free floating object in orbit. *Autonomous Robots*, 1(1):53–68, 1994.

[17] Matthew Berkemeier, Eric Poulson, Eric Aston, Josh Johnston, and Brian Smith. Development and Enhancement of Mobile Robot arms for EOD Applications. *Proceedings of SPIE*, 6962(May 2011):69620P–69620P–12, 2008.

[18] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Inc, 2006.

[19] E. Brown, N. Rodenberg, J. Amend, a. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, and H. M. Jaeger. From the Cover: Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, October 2010.

[20] Richard G. Budynas and J. Keith Nisbett. *Shigley's Mechanical Engineering Design*. McGraw Hill, New York, eigth edition, 2008.

[21] Robert L. Norton. *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*. McGraw Hill, New York, third edition, 2004.

[22] M.R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human*, 37(1):32–64, 1995.

[23] David R. Thompson and Rebecca Castano. Performance Comparison of Rock Detection Algorithms for Autonomous Planetary Geology. *IEEE Aerospace Conference*, (2):1–9, 2007.

[24] Peter Kovesi. Phase preserving denoising of images. *Science*, 4(3):1.

[25] C Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *IEEE International Conference on Computer Vision*, 1998.

[26] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[27] Steven Vozar and Dawn M. Tilbury. Augmented reality user interface for mobile ground robots with manipulator arms. *SPIE*, 7878(May):1–9, 2011.

[28] Michael S Rose. *Design of a Helicopter Deployable Ground Robotic System for Hazardous Environments*. PhD thesis, Virginia Polytechnic Institute and State University, 2010.

[29] Peter Corke. MATLAB toolboxes: robotics and vision for students and teachers. *IEEE Robotics & Automation Magazine*, 14(4):16–17, December 2007.

Appendix A

Denavit-Hartenberg Representation of the Arm

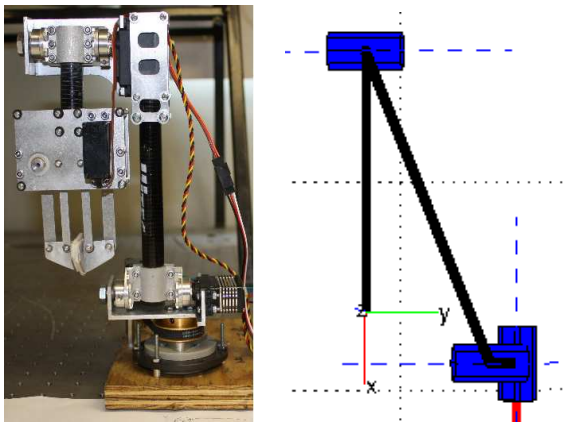
This section establishes a set of coordinate frames on the robot arm that allow for the solving of the forward kinematics. This information will be useful when the arm is operated in 3D. Knowledge of the end effector location relative to the base of the robot must be known in order to reach locations that correspond to locations viewed through the Kinect^{TM}. This is known as the forward kinematics problem and can be determined using a Denavit-Hartenberg (DH) representation of the robot. Assuming knowledge of the joint angles, one can determine the position of the end effector knowing the geometry of the robot arm.

The modified DH parameters used to simulate the robot arm are as follows:

Table A.1: DH parameters for the full robot arm.

	α	a	θ	d
Base	0	0	θ_1	0
Shoulder	$-\frac{\pi}{2}$	$\frac{1}{4}$	θ_2	$-\frac{5}{8}$
Elbow	0	7.5	θ_3	-3
EE	$-\frac{\pi}{2}$	6.0	0	0

Figure A.1 shows the actual robot arm compared to what the DH parameters generate in the a robot schematic generation code. The DH parameters will be useful in solving the inverse kinematics and forward kinematics in full 3D operation. After feedback on the base joint is established, these parameters can be used to determine the end effectors orientation and location in space.



(a) (b)

Figure A.1: Figure (a) shows the real robot. Figure (b) shows a schematic used to confirm the DH parameters. The image was generated using the Peter Corke Robot Toolbox [29] in conjunction with Matlab

Appendix B

Stress Calculations: Arm Member Stresses

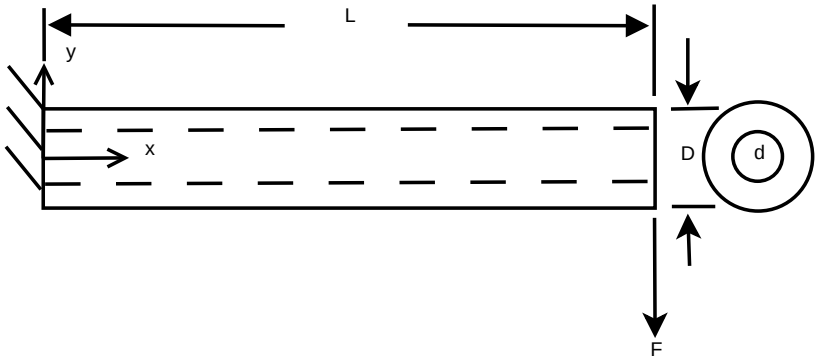


Figure B.1: Free body diagram for the shoulder link with a force at the end.

The dimensions and force in figure B.1 are: D =0.0175 m, d= 0.0172 m, L =0.127 m and F= (0.75 kg)(9.81 m/s^2)= 7.36 N

In bending, the max stress is represented by:

$$\sigma_{max} = \frac{Mc}{I}$$

The second moment of inertia I and c is represented by:

$$I = \frac{\pi}{64}(D^4 - d^4) =, c = \frac{D}{2}$$

The new stress equation becomes:

$$\sigma_{max} = \frac{MD}{2\frac{\pi}{64}(D^4 - d^4)}$$

The moment (M) becomes equal to

$$M = FL = (7.36\,\text{N})(0.127\,\text{m}) = 0.934\,\text{Nm}$$

$$\sigma_{max} = \frac{0.934\,\text{Nm} \cdot 0.0175\,\text{m}}{\frac{\pi}{32}((0.0175\,\text{m})^4 - (0.0172\,\text{m})^4)} = 31.3\,\text{MPa}[4500\,\text{psi}]$$

Since the joint is offset, torsion will be seen on the shoulder member. The sample, the elbow joint, and the end effector weight all contribute to a twist about the x-axis in figure B.1. This mass will be estimated at 0.7 kg. The distance out to the centroid of the collective mass could be used to calculated for the as the average of the centroids of each individual piece. This value can be over estimated as the distance between the centers of the two links as illustrated in figure B.2. The equation for max torsional shear stress is:

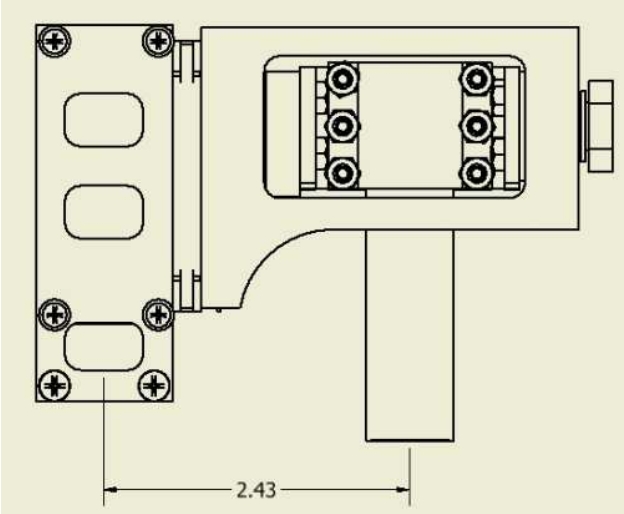


Figure B.2: Distance between centers of the links in inches.

$$\tau_{z,x} = \frac{Tr}{J}$$

$$= \frac{T\frac{D}{2}}{J}$$

Using the mass estimate of 0.7 kg and the distance of 2.43 in (0.0617 m), the torque at the

joint (T) is 0.424 Nm. The polar moment of inertia (J) for this case is:

$$J_G = \frac{\pi}{32}(D^4 - d^4)$$

The torsional shear stress becomes:

$$\tau_{z,x} = 16\frac{T(D)}{\pi(D^4 - d^4)} = 16\frac{0.424\,\text{Nm}(0.0175\,\text{m})}{\pi((0.0175\,\text{m})^4 - (0.0172\,\text{m})^4)} = 7.24\,\text{MPa}[1050\,\text{psi}]$$

The von Mises stress is calculated with:

$$\sigma' = \sqrt{\sigma_{max}^2 + 3\tau_{zx}^2} = 31.4\,\text{MPa}[4500\,\text{psi}]$$

Referring back to the material strengths in table 3.2, the uni-direction carbon fiber tube has a minimum ultimate strength of 7200 psi. This makes the factor of safety 1.6.

Appendix C

Application of Lewis Equation for Gearing

The Lewis equation is shown below and was applied to the acetal gears in the end effector.

$$\sigma = \frac{W^t P}{FY} \tag{C.1}$$

W^t is the tangential force on the gear which will be assumed to be the stall torque at the motor divided by the radius of the pitch circle. The value for W^t is 9.4 lbf. The diametrical pitch is the number of teeth divided by the diameter of the pitch circle. For this particular gear, P is 32 teeth/inch. Y is the Lewis factor and is 0.422 [20]. 'F' is the thickness of the gear at 0.1875 inches. From the equation C.1, the stress estimate is 3800 psi which is below the tensile strength of Delrin® at 10,000 psi; this gives a safety factor of 2.6.

Appendix D

End Effector Link Loading

This section will show that the end effector links are able to withstand the max static forces in bending that will occur. These two calculations consider the case where the gripper closes around an object at the stall torque of the motor. The second calculation shows that the link can support the weight of the sample.

Figure D.1 shows a free body diagram for when the link closes around a rock. The loading can be approximated as a cantilever beam with a force acting at the tip. The maximum force seen is:

$$\sigma_{max} = \frac{Mc}{I} = \frac{M}{Z}$$

In this particular case, Z is the section modulus defined by the second moment of inertia (I) divided by the half the height (h) of the linkage. Represented another way:

$$Z = \frac{I}{c}$$
$$Z = \frac{bh^2}{6}$$

The moment (M) corresponds to the stall torque at the end effector motor.

The maximum stress becomes:

$$\sigma_{max} = \frac{6M}{bh^2} = \frac{6(0.99\,\mathrm{Nm})}{0.00158\,\mathrm{m}(0.00956\,\mathrm{m})^2} = 41.1\,\mathrm{MPa}$$

Aluminum 6020 has a yield strength of 338 MPa which gives a safety factor of 8.2.

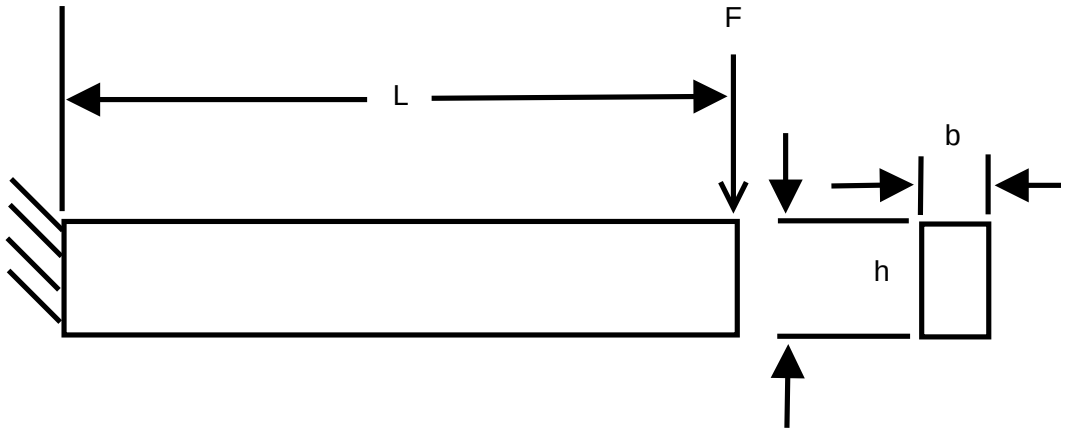


Figure D.1: Single end effector link resisting gripping force applied by the motors.

Rotating the link from figure D.1 by 90 degrees and changing the loading is how D.2 was created. The above work can be adapted for the situation where a single end effector link is holding up a 0.250 kg sample. Both link widths, as seen in figure 3.8 , will be accounted for below. The loading seen in the actual end effector is different in that the rock is supported by 4 links as oppose to just a single linkage. These calculations simulate a worse case scenario where only a single link is supporting a rock.

$$\sigma_{max,top} = \frac{6FL}{bh^2} = \frac{6(0.25\,\mathrm{N})(0.0635\,\mathrm{m})}{0.00953\,\mathrm{m}(0.001578\,\mathrm{m})^2} = 39\,\mathrm{MPa}$$

$$\sigma_{max,bottom} = \frac{6FL}{bh^2} = \frac{6(0.25\,\mathrm{N})(0.0635\,\mathrm{m})}{0.0066\,\mathrm{m}(0.001578\,\mathrm{m})^2} = 57\,\mathrm{MPa}$$

The top link is the wider (b) of the two links. The larger stress is seen in the lower and smaller linkage.

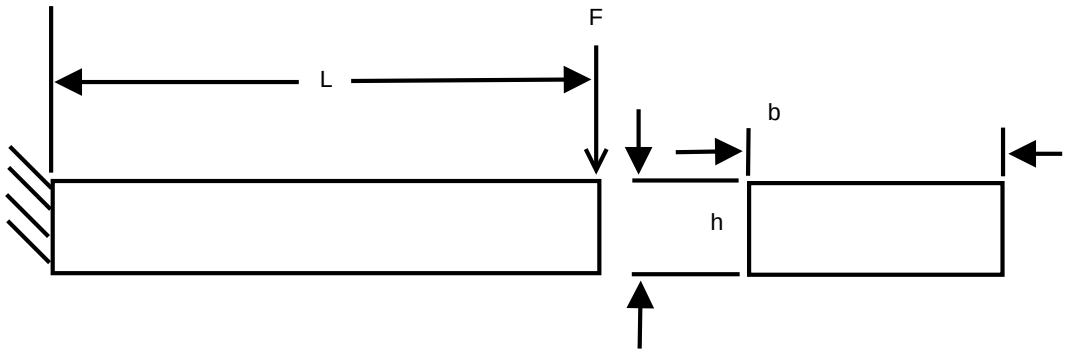


Figure D.2: Single end effector link resisting the weight of the rock. The variables 'b' and 'h' are redefined.

Appendix E

Power and Command Circuit

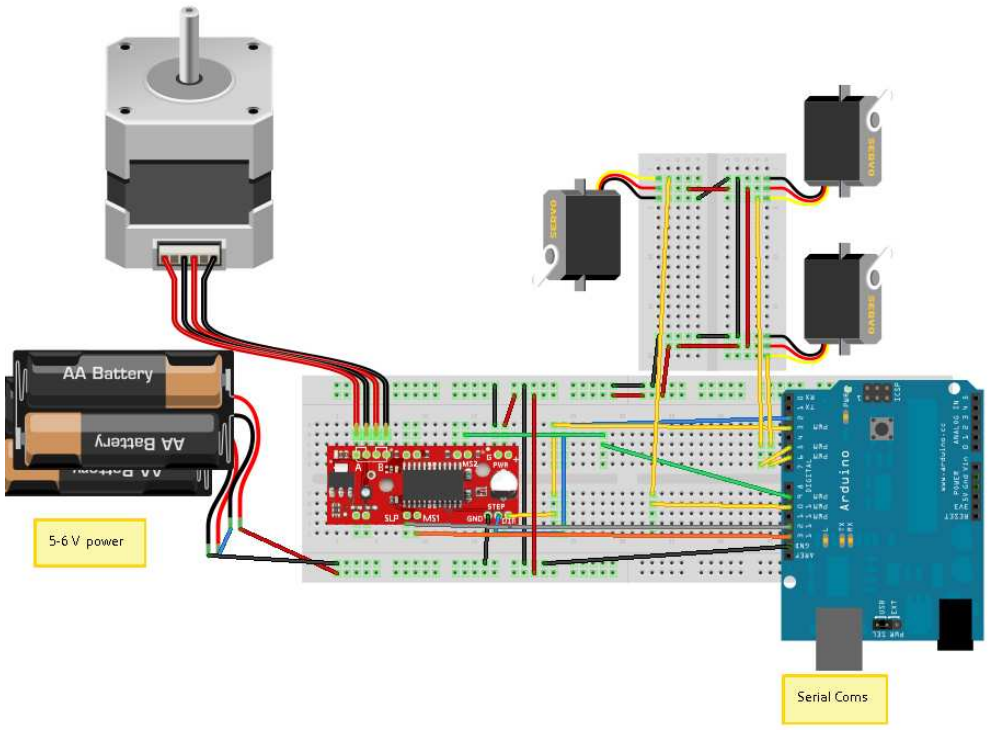


Figure E.1: Power and motor command circuit. Double A batteries are used in place of a power supply on the diagram.

Appendix F

Autodesk Timing Belt Strength Test

Table F.1: Timing belt software strength test done with the Autodesk 2011 design accelerator.

Power	P	0.001 hp
Torque	T	0.100 lbforce ft
Speed	n	30.000 rpm
Efficiency	η	0.970
Service factor	c_p	1.000
Resultant service factor	c_{PR}	1.159
Belt Speed	v	0.060 fps
Effective pull	F_p	5.236 lbforce
Centrifugal force	F_c	0.000 lbforce
Belt installation tension	F_t	3.403 lbforce
Maximum tension in belt span	F_{tmax}	6.021 lbforce
Teeth in mesh factor	k_z	1.000
Tension factor	k_1	1.1500

The maximum tension in the belt span (F_{tmax}) at 6.283 lbf does not exceed the maximum allowable tension in the belt (T_a= 6.07 lbf). Default software parameters were used for this particular case.

Appendix G

Bill of Materials

Items marked as having a price of zero were received through the IGUS Y.E.S. program or were found in the Unmanned Systems Lab shop area.

Table G.1: Sales Bill of Material:Base

Qty	**Name**	**Price Paid**	**Price/Arm**
1	Stepper Motor	14.95	14.95
1	Slewing bearing	0.00	0.00
1	MXL Timing Belt 18 tooth Drive Pulley 1/4" belt	8.60	8.60
1	MXL Timing Belt 60 tooth Drive pulley	14.63	14.63
1	MXL series neoprene timing belt 88 trade size	3.66	3.66
6	Standoff 1/4" length ID 0.166" OD 1/4"	1.62	1.62
4	Spacer for motor M3	15.36	15.36
4	Machine Screws M3 10s Motor	6.86	0.27
4	Machine Screws M3 40s	0.00	0.00
24	Washers No. 8	0.00	0.00
6	Bolts No. 8	0.00	0.00
	Sub Total	**65.68**	**59.09**

Table G.2: Sales Bill of Material:Shoulder

Qty	**Name**	**Price Paid**	**Price per Arm**
1	Servo Hitec HS-7950	130.00	130.00
1	Shoulder Screw - 3/8”	5.01	5.01
1	Flanged Bushing	0.00	0.00
1	Set Screw Hub	4.99	4.99
1	Hitec Servo Hub	9.99	9.99
0.5’	Aluminum T-bar	28.06	1.75
4	4-40s	0.00	0.00
8	4-40 washers	0.00	0.00
8	4-40 nuts	0.00	0.00
5	5-40 machine screws plain	3.81	0.19
0.25’	U-channel	37.96	1.19
4	Steel Nylon-Insert Thin Hex Locknut	2.29	0.09
2	Aluminum Clamps WJ	23.30	23.30
	Sub Total	**245.41**	**176.51**

Table G.3: Sales Bill of Material:Elbow. The price per arm is higher than the price of the arm as a result of the T-bar and the U-channel being already factored into the price of the shoulder.

Qty	**Name**	**Price Paid**	**Price per Arm**
1	Servo HS-5685MH	56.99	56.99
1	Shoulder Screw - 3/8”	5.01	5.01
1	Flanged Bushing	0.00	0.00
1	Servo Shaft Attachment (0.375)	9.99	9.99
0.5’	Aluminum T-bar	0.00	1.75
1	Set Screw Hub	4.99	4.99
1	Hitec Servo Hub	9.99	9.99
4	M 2.5 for motor mount	0.00	0.00
6	Steel Nylon-Insert Thin Hex Locknut	2.29	0.14
0.25’	U-channel	0.00	1.19
2	Aluminum Clamps WJ	23.30	23.30
	Sub Total	**89.26**	**113.35**

Table G.4: Sales Bill of Material:End Effector

Qty	**Name**	**Price Paid**	**Price per Arm**
1	Motor-JR Sport ST47BB	16.95	16.95
4	60 Pitch Acetal Hub Gears 3/16 face	23.68	23.68
2	Flanged Bushing	0.00	0.00
12	MFI-0203	0.00	0.00
0.16’	Aluminum (Alloy 6061) T-Bar	10.15	0.20
3	External Retaining Ring 3/8” Dia	11.99	0.36
12	External Retaining Ring 1/8” Shaft Dia	4.47	0.54
1	3/8 Bore Clamping Hub	7.95	7.95
4	Acetal Flat Washer	11.71	0.94
4	Female Round threaded standoffs	7.38	4.92
4	Small Standoff - Mid plate	2.40	1.60
16	EPDM Rubber Washers #6	4.61	0.74
4	Motor Bolts- 4-40 Machine Screws	3.96	0.16
4	Steel Nylon-Insert Thin Hex Locknut	2.29	0.09
4	Motor Washers #4 Nylon	4.78	0.19
4	#4 for mid cap plate	5.56	0.22
0.25 ’	Aluminum Tubing	7.27	0.30
0.11 %	Foam for End Effector Grippers	6.91	0.76
8	Socket Cap Screws	5.73	1.83
5	5-40 Machine Screws Plain	3.81	0.19
	Sub Total	**141.60**	**61.62**

Appendix H

Source Code

H.1 Computer Assisted Mode

```
1
2  % Shajan Thomas
3  % Matlab Rock Picking Code
4  % Rockpicking involves 2 steps: finding where the rock centroid is, and
5  % moving the robot arm to the desired position.
6
7  % Last Revised: July 26, 2011
8  % Originally Created: May 5,2011
9
10 close all; clear all;clc;
11
12
13 addpath('./robot')                    % robot visualization toolbox
14 addpath('./OpenNI/Mex');              % the Kinect code
15 addpath('./OpenNI/rockcollection')    %
16
17
18 debug=0;  % set to 1 for additional figure displays
19
20 %% Create context with xml file
21 context = mxNiCreateContext('OpenNI/Config/SamplesConfig.xml');
22
23 %%  Initialise FIGURE
24 width = 640; height = 480;
25 close all;
26 bigfig=figure; h5 = mesh(zeros(height,width,'double'),...
27             zeros(height,width,'double'),zeros(height,width,'double'));
28             axis([-500 400 -400 400 1 1000]); view(180,90);
29             xlabel('X [mm]'); ylabel('Y [mm]'); zlabel('Z [mm]');
30             title('XYZ The world coordinate system');
31
```

```
32 if(debug==1)
33 smallfig=figure; h6 = mesh(zeros(height,width,'double'),...
34               zeros(height,width,'double'),zeros(height,width,'double'));
35             axis([-300 300 -300 300 1 1000]); view(180,90);
36             xlabel('X [mm]'); ylabel('Y [mm]'); zlabel('Z [mm]');
37             title('XYZ Subset');
38 end
39
40 %% Read point cloud off Kinect
41     %align Depth onto RGB
42     option.adjust_view_point = true;
43     % Acquire RGB and Depth image
44     mxNiUpdateContext(context, option);
45     [rgb, depth] = mxNiImage(context);
46     % convert uvZ[pix,pix,mm] to XYZ[mm,mm,mm]
47     real_XYZ = mxNiConvertProjectiveToRealWorld(context, depth);
48     %convert XYZ[mm,mm,mm] to uvZ[pix,pix,mm]
49     projective_uvZ = mxNiConvertRealWorldToProjective(context, real_XYZ);
50
51     tmp = real_XYZ(:);
52     tmp(find(real_XYZ==0)) = NaN;
53
54     tmp = reshape(tmp,[height width 3]);
55     set(h5,'XData',-tmp(:,:,1));
56     set(h5,'YData',-tmp(:,:,2));
57     set(h5,'ZData',tmp(:,:,3));
58     %colormap hsv;
59     drawnow;
60
61     figure(bigfig)
62      % colormap hsv;
63
64 %% Extracting Region of Interest
65
66 figure(bigfig);
67 datacursormode on
68 dcm_obj = datacursormode(bigfig);
69 disp('Place Datatip and Hit Return.')
70 pause
71 %ginput(1)   % does not work as expected
72 datacursormode off
73
74 c_info = getCursorInfo(dcm_obj);
75
76 clear dcm_obj
77 xyz = c_info.Position';
78
79 if(debug)
80 figure(smallfig);  % switches current figure back to the last figure made
81 end
82 wind=34;
```

```
83
84  fx= 600;
85  fy=600;
86
87  cx=320;
88  cy= 240;
89
90  RT= [fx 0 cx; 0 fy cy; 0 0 1];
91
92  coord=RT*xyz;
93  scalefactor=coord(3);
94  coord=coord/scalefactor;
95
96  row=uint16(coord(2)); %
97  col=uint16(640-coord(1));
98
99  % extract region of interest from row,col information above
100
101 minrow=row-wind;
102 mincol=col-wind;  % will be necessary later to get centroid coordinates
103                   % into new frame
104 maxrow=row+wind;
105 maxcol=col+wind;
106
107 subsettmp=tmp(minrow:row+wind, mincol:col+wind, :);
108
109  if(debug==1)
110      set(h6,'XData',-subsettmp(:,:,1));  % removed negative sign here
111      set(h6,'YData',-subsettmp(:,:,2));
112      set(h6,'ZData',subsettmp(:,:,3));
113
114
115  % now set max for h6 so we can rotate around rocks
116  axis([min(min(-subsettmp(:,:,1))) max(max(-subsettmp(:,:,1)))...
117       min(min(-subsettmp(:,:,2))) max(max(-subsettmp(:,:,2)))...
118      min(min(subsettmp(:,:,3))) max(max(subsettmp(:,:,3)))]);
119  end
120
121 %flipud(fliplr)
122 img1=subsettmp(:,:,3);
123
124 minimg1= min(min(img1));
125 maximg1=max(max(img1));
126
127 img1mod=uint8((img1-minimg1)/((maximg1-minimg1)/255));
128
129 % for maximum processing efficiency the image needs to be converted to a
130 % size that is a multiple of 2
131 C=img1mod(1:2*wind,1:2*wind);
132 %figure;imshow(edge(C,'canny',thresh,sigma))
133
```

```
134 C=double(C)/255;
135 %figure;
136
137
138 disp('Commence image Denoising')
139 k       =  3;        %   number of standard deviations of noise to reject 2
140 nscale  =  7;        %   number of filters to use  orignally 7
141 mult    = 2.5;        %   multiplying factor between scales
142 norient =  6;        %   number of orientations to use
143 softness=  0;        %   hard or soft thresholding 0-hard  1 -soft
144
145 image2=noisecomp(C, k, nscale, mult, norient, softness);
146
147 % rescale image to bring intensity domain back to 0 to 1
148 minimg1= min(min(image2));
149 maximg1=max(max(image2));
150
151 image2=((image2-minimg1)/(maximg1-minimg1));
152 denoised=image2;
153
154
155 % bi lateral filter
156 disp('Commence Bi-lateral filtering')
157
158 % filter half window size
159 w=5 ; %2^4;
160 % standard deviations in the spatial and intensity domains
161 sigma= [3  .1];
162 image2= jbfilter2(image2,C,w,sigma);
163 image3=image2;
164 bilat=image2;
165
166 % figure;
167 % imshow(image2)
168 % title('Bi-Lateral Filtering')
169
170 % Canny Edge Detection
171 sigma=2;
172 thresh=[.2  .9];
173 grad2 = edge(image2,'canny',thresh,sigma);
174  if(debug==1)
175     figure; imshow(grad2);
176  end
177
178 % additional edge processing to try to isolate the rock
179 BW=imfill(grad2,'holes');
180 dim = size(BW);
181
182 % Dilation to remove extraneous lines
183 seD = strel('diamond',1);
184 BWfinal = imerode(BW,seD);
```

```
185
186 BWfinal2=imdilate(BWfinal,seD);
187  if(debug==1)
188      figure, imshow(BWfinal2);
189  end
190
191 % display final result
192 I=C;
193 I(BWfinal2)=1;
194 s = regionprops(BWfinal2, I, {'Centroid'});
195
196 elements=numel(s);
197
198 if elements==0
199    disp('no rock detected')
200 end
201
202 if(debug==1)
203      imshow(I);
204      hold on
205      for k = 1 : elements
206          plot(s(k).Centroid(1), s(k).Centroid(2), 'r*');
207      end
208      hold off
209      title('Rock Hilighted and Boundaries Detected')
210  end
211
212      % calc centroid on the object
213
214      depthinfo=img1(1:wind*2,1:wind*2);
215      justboundary= edge(BWfinal2,'canny',thresh,sigma);
216      extboundary= zeros(size(justboundary));
217      alpha= 1.2;  % dilation factor
218
219      [rs,cs]= find(justboundary==1);
220      rc=double(s.Centroid(2));
221      cc=double(s.Centroid(1));
222
223      rs=alpha*(double(rs)-rc)+rc;
224      cs=alpha*(double(cs)-cc)+cc;
225
226      rs=int16(rs);
227      cs=int16(cs);
228
229      for ii=1:length(rs)
230
231          extboundary(rs(ii), cs(ii))= 1;
232
233      end
234
235
```

```
236     depthinfo(isnan(depthinfo)) = -scalefactor;
237
238     zavgtop=mean(depthinfo(BWfinal2));
239     zavgbottom=mean(depthinfo(logical(extboundary)));
240     depthinfo(logical(extboundary))=zavgbottom;
241
242
243     zavg=(zavgtop+zavgbottom)/2;
244
245     clear rs cs vs alpha extboundary
246
247     ccdcoords=[s.Centroid(1)+ mincol; s.Centroid(2)+ minrow; 1];
248     centroidWorld=inv(RT)*(double(ccdcoords).*scalefactor);
249     centroidWorld(1)=-centroidWorld(1);
250
251     %  Replace z which was the scalefactor with average of zs
252     centroidWorld(3)=zavg;
253     disp('World Coordinate Centroid:')
254     disp(centroidWorld')
255
256 %      % Apply experimentally determined correction factor
257 %     centroidWorld(1)=(centroidWorld(1)*0.1459+2.1028) + centroidWorld(1);
258 %     centroidWorld(2)=(centroidWorld(2)*0.1495-3.957) + centroidWorld(2);
259
260     disp('Corrected World Coordinate Centroid')
261     disp(centroidWorld')
262
263     CWI=centroidWorld/25.4;
264     CWI=[CWI; 1];
265     TranMM= [155.67; -81.33; 715.1];
266
267     TranIn=[TranMM/25.4; 1];
268     R=[1 0 0; 0 -1 0; 0 0 -1;0 0 0];
269
270     T= [R TranIn];
271     CAI=T*CWI;
272     CAI(4)=[];
273     disp('Centroid relative to the arm')
274     disp(CAI');
275
276     %% Open Port
277
278     ser = serial('COM8');
279     pause(1)
280     set(ser,'BaudRate',9600, 'Terminator', 'CR');
281     fopen(ser);
282
283     %% Pick up rock
284     pause(1)
285     pickuprock(CAI,ser)
286
```

```
287     %% Close Serial Port
288     fclose(ser);
289     delete(ser)
290     clear ser
291
292     % Delete the context object
293     mxNiDeleteContext(context);
294
295     % close all
296     close all
```

H.2 Pick Up Rock

```
1  function []=pickuprock(Tt,s)
2  % input a column vector of [x;y;z]
3  % outputs:  no variables- motion should be visible on arm
4
5  addpath('./inversekinfunctions')
6  addpath('./robot')
7
8  % end effector commands
9  open = 850;
10 closed = 650;
11
12 % Home previously set
13 qh=[-1.5902 1.625 0];
14
15 qt=ikcustom(Tt);
16
17 % Create two PWM commands for each motor
18 pwm(1)=rad2Should(qh(1));
19 pwm(2)=rad2Elbow(qh(2));  % out of range need to investigate
20
21
22 if~(isnan(pwm(1)) || isnan(pwm(2)))
23     % send out serial command
24     disp(pwm)
25
26     MoveArmMotor('S', pwm(1),s);
27     pause(delayval)
28     MoveArmMotor('L', pwm(2),s);
29     pause(delayval)
30 end
31
32 % open gripper
33 MoveArmMotor('E', open,s)
34 pause(delayval)
35
36 tstep=0.5;
37 t=0:tstep:3;
```

```
38
39 jt=jtraj(qh, qt, t);    % trajectory to target
40
41
42 for q=jt',
43
44     pwm(1)=rad2Should(q(1));
45     pwm(2)=rad2Elbow(q(2));
46     pause(delayval)
47     MoveArmMotor('S', pwm(1),s);
48     pause(delayval)
49     MoveArmMotor('L', pwm(2),s);
50     pause(delayval)
51 end
52
53
54 %close gripper to capture sample
55 pause(delayval)
56 pause(5);
57 MoveArmMotor('E', closed,s)
58 pause(delayval)
59
60 % move motor %%%
61 % move to home location
62
63 % return to a home location
64 % jt=jtraj(qt, qh, t);
65 %
66 % for q=jt',
67 %     pwm(1)=rad2Should(q(1));
68 %     pwm(2)=rad2Elbow(q(2));
69 %     pause(delayval)
70 %     MoveArmMotor('S', pwm(1),s);
71 %     pause(delayval)
72 %     pause(0.020);
73 %     MoveArmMotor('L', pwm(2),s);
74 %     pause(delayval)
75 % end
```

H.3 Motor Command Interpretting

```
1 // servo control code- sends out absolute commands
2 // Interprets signal coming in from Matlab code and creates corresponding arm
      motion
3 // currently supports open loop teleoperated operation
4 // by Shajan Thomas   April 18,2011
5
6
7 #include <Servo.h>
8
```

```
 9
10 ////base stepper motor variables
11 int DIR = 3;            // PIN  3 = DIR
12 int STEP = 2;           // PIN  2 = STEP
13 int MS1 = 13;           // PIN 13 = MS
14 int MS2 = 9;            // PIN  9 = MS2
15 int SLEEP = 12;         // PIN 12 = SLP
16 int currentPos=0;    //  rest position of the stepper
17
18 //Base vars
19 int Basepos;
20 int Baselimits[2]={0,333};
21 int Basehome=150;    //needs to be experimentally determined
22
23
24 byte pinEE=5;          // PIN 5 = SIGNAL EE
25 byte pinElbow=6;       // PIN 6 = SIGNAL Elbow
26 byte pinShould=10;     // PIN 10 = SIGNAL Shoulder
27
28 //Serial Com variables
29 int val=0;            //variable that determines the number of steps to take
30 byte inByte1= 0;
31 byte inByte2= 0;
32 byte inByte3= 0;
33 byte inByte4=0;
34
35 //End Effector vars
36 Servo EE;
37 int EEpos;
38 int EElimits[2]={650,850};
39 int EEhome=650;
40
41 //Elbow vars
42 Servo Elbow;
43 int Elbowpos;
44 int Elbowlimits[2]={900,2100};
45 int Elbowhome=2100;
46
47 //Shoulder vars
48 Servo Should;
49 int Shouldpos;
50 int Shouldlimits[2]={1000,2127};
51 int Shouldhome=1500;
52
53
54 void setup()
55 {
56   EE.attach(pinEE,EElimits[0],EElimits[1] );  // attaches the servo on pin 9
         to the servo object
57    Serial.begin(9600);
58    EE.writeMicroseconds(EEhome);  //start closed
```

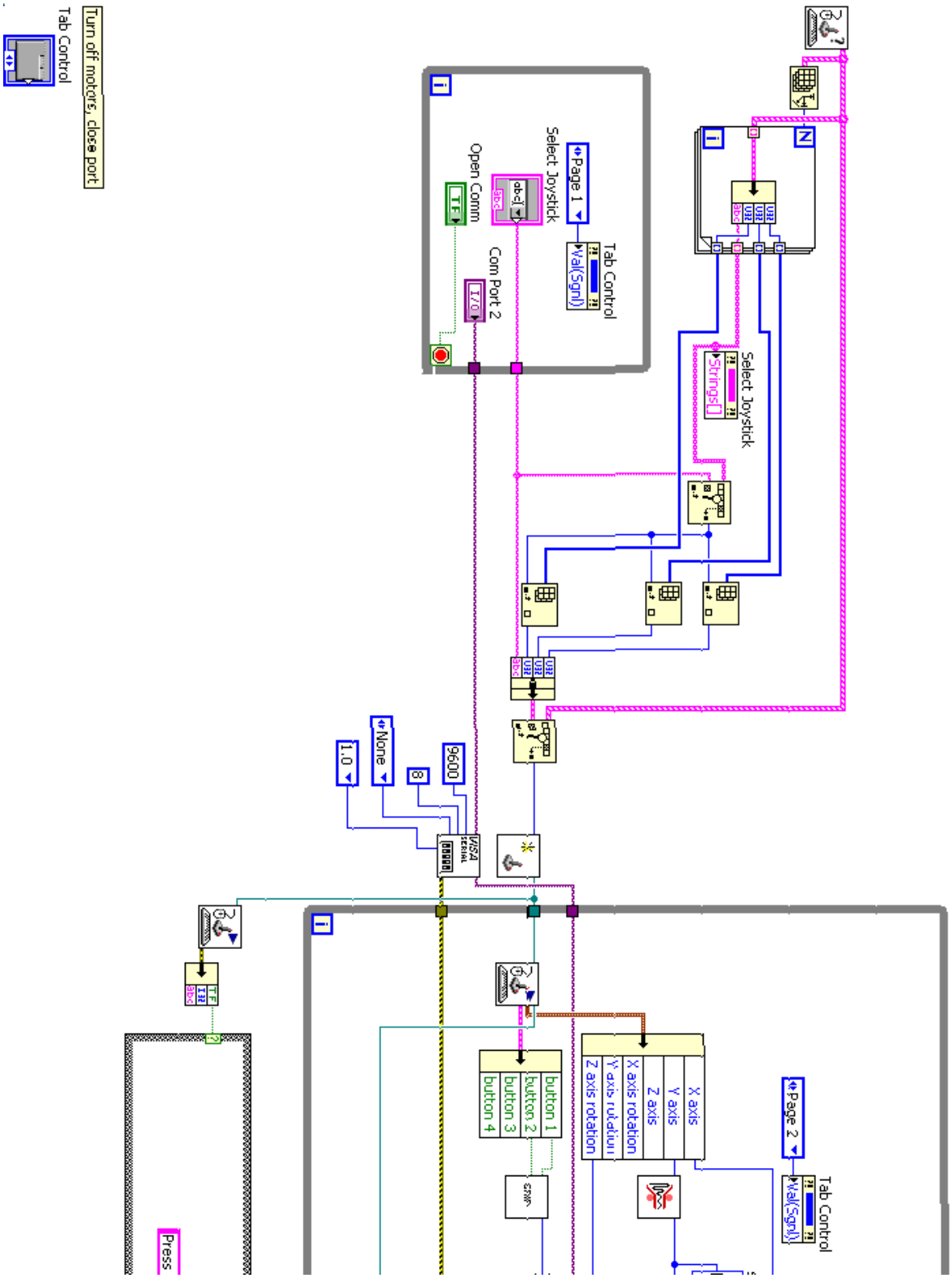
```
59    EEpos=EEhome;
60
61    Elbow.attach(pinElbow,Elbowlimits[0],Elbowlimits[1] );  // attaches the
          servo on pin 9 to the servo object
62    Elbow.writeMicroseconds(Elbowhome);  //start closed
63    Elbowpos=Elbowhome;
64
65    Should.attach(pinShould, Shouldlimits[0],Shouldlimits[1]);
66    Should.writeMicroseconds(Shouldhome);
67    Shouldpos=Shouldhome;
68
69     Basepos=Basehome;
70
71    pinMode(DIR, OUTPUT);    // set pin 3 to output
72    pinMode(STEP, OUTPUT);   // set pin 2 to output
73    pinMode(MS1, OUTPUT);    // set pin 13 to output
74    pinMode(MS2, OUTPUT);    // set pin 9 to output
75    pinMode(SLEEP, OUTPUT); // set pin 12 to output
76    digitalWrite(SLEEP, LOW);
77  }
78
79  void loop()
80  {
81    if(Serial.available() >3) {
82      inByte1=Serial.read();
83
84      if(inByte1=='A'){
85        inByte2=Serial.read();
86
87        // End Effector motion
88        if(inByte2=='E'){
89
90            inByte3=Serial.read();    //high byte
91            inByte4=Serial.read();    //low byte
92
93            //Serial.print(inByte3,DEC); Serial.print(','); Serial.println(
                  inByte4,DEC);
94            val= (int(inByte3) << 8) + int(inByte4);
95
96            //delay(10);
97            //Serial.println(val,DEC);
98
99            if (val>=4000)
100           {
101              // read current value
102              Serial.println(EEpos,DEC);
103           }
104           else if (val==0)
105           {
106              EE.detach();
107           }
```

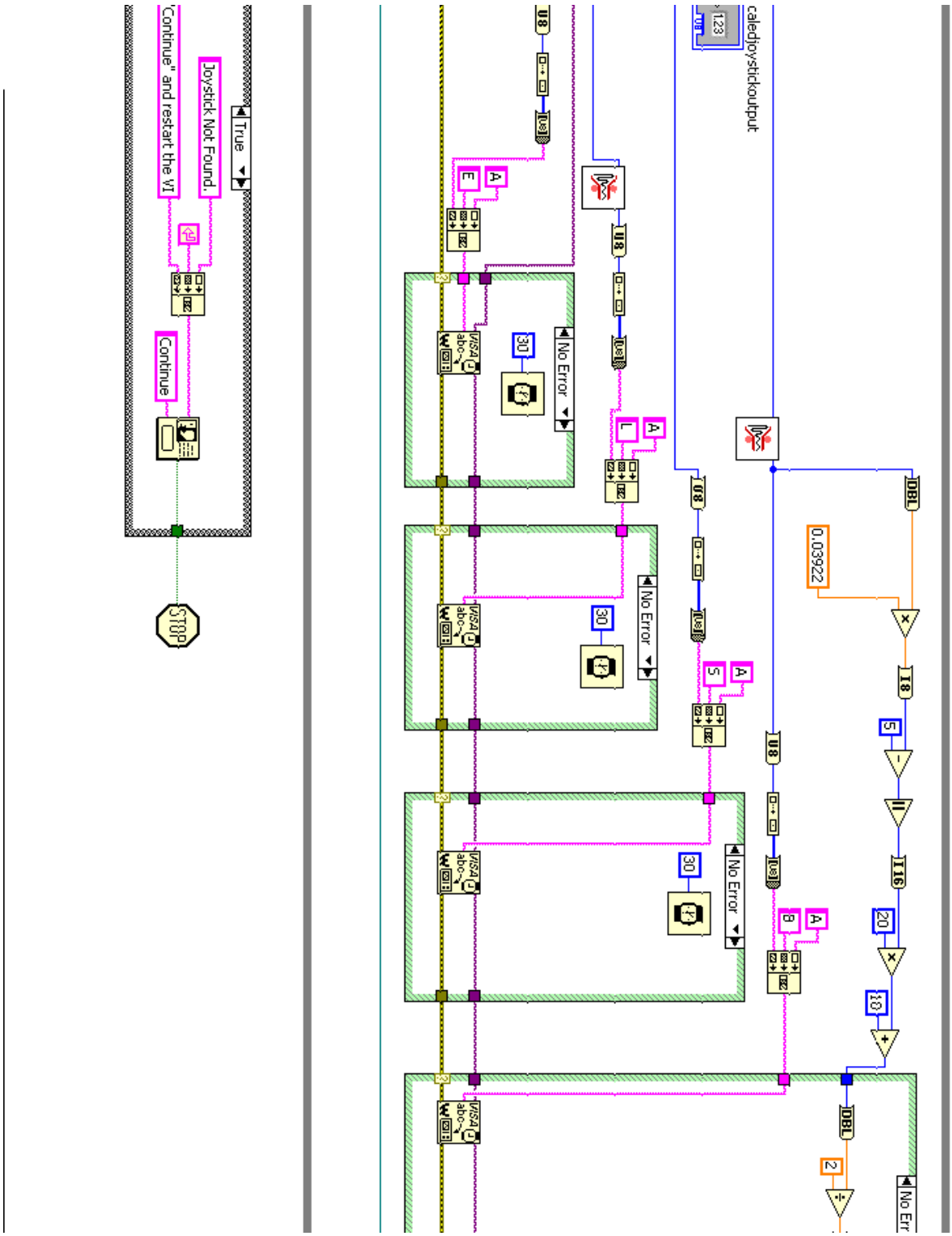
```
108           else if (val>=EElimits[0] && val<=EElimits[1])
109           {
110             if(!EE.attached()) {EE.attach(pinEE); EE.writeMicroseconds(EEhome)
                    ; EEpos=EEhome;}
111             EEpos=val;
112             EE.write(EEpos);
113           }
114         } // end effector level
115
116
117        else if(inByte2=='L'){
118         inByte3=Serial.read();
119         inByte4=Serial.read();
120
121          val= (int(inByte3) << 8) + int(inByte4);
122
123
124          if( val == 0) {
125              Elbow.detach();
126         }
127         else if( val >=4000){
128           val=Elbow.read();
129           Serial.println(Elbowpos,DEC);
130         }
131         else if(val>=Elbowlimits[0] && val<=Elbowlimits[1]){
132           if(!Elbow.attached()) {Elbow.attach(pinElbow); Elbow.
                writeMicroseconds(Elbowhome); Elbowpos=Elbowhome;}
133
134             Elbowpos=val;
135             Elbow.write(Elbowpos);
136
137         }
138       }// elbow level
139
140
141
142         else if(inByte2=='S'){
143         inByte3=Serial.read();
144         inByte4=Serial.read();
145
146          val= (int(inByte3) << 8) + int(inByte4);
147
148
149          if( val == 0) {
150              Elbow.detach();
151         }
152         else if( val >=4000){
153           val=Should.read();
154           Serial.println(Shouldpos,DEC);
155         }
156         else if(val>=Shouldlimits[0] && val<=Shouldlimits[1]){
```

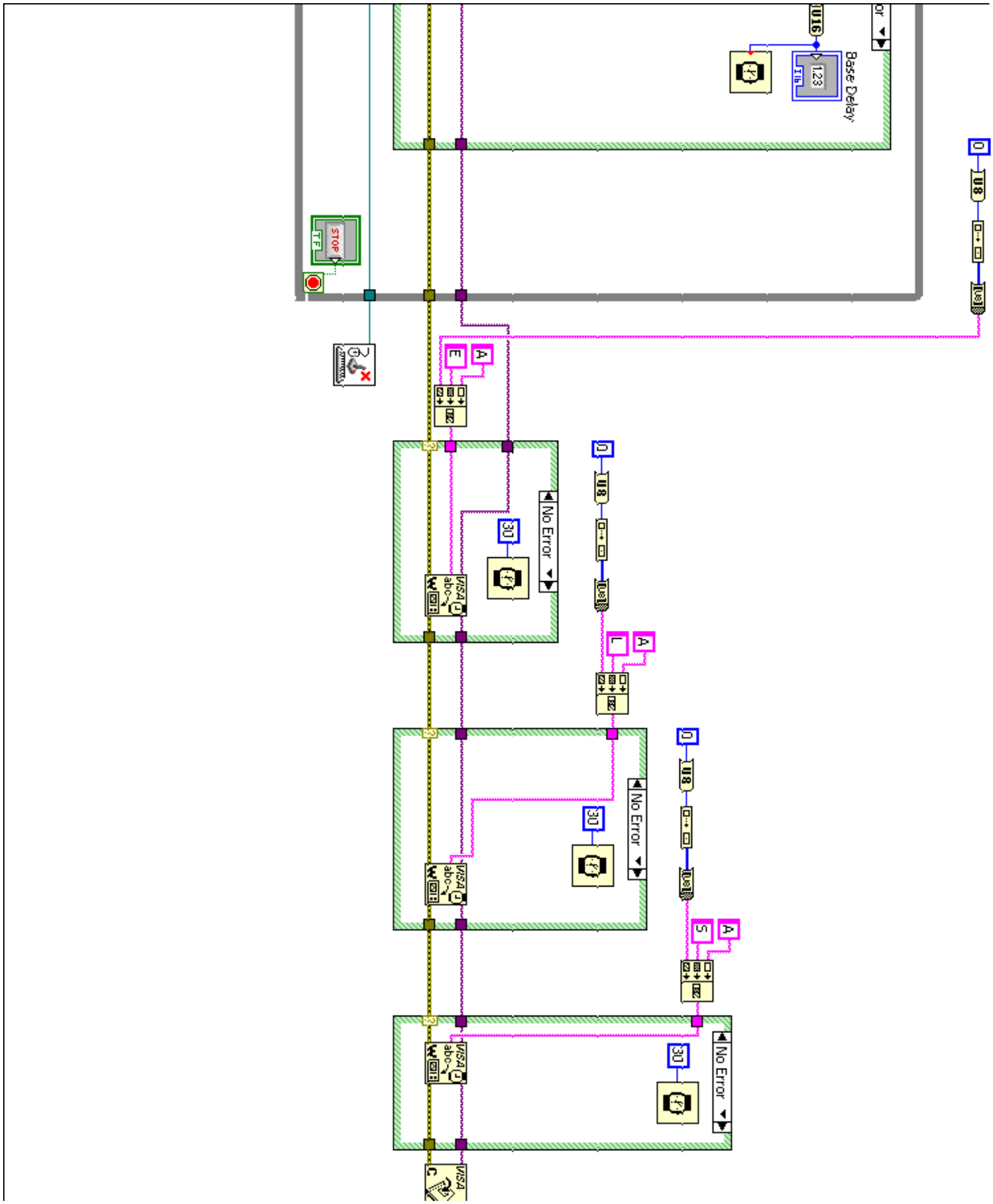
```
157             if(!Should.attached()) {Should.attach(pinShould); Should.
                    writeMicroseconds(Shouldhome); Shouldpos=Shouldhome;}
158
159               Shouldpos=val;
160               Should.write(Shouldpos);
161
162           }
163         }// shoulder level
164
165
166
167         else if(inByte2=='B'){
168           inByte3=Serial.read();
169
170
171          if(inByte3==0 ) {
172               digitalWrite(SLEEP,LOW);
173          }
174
175          else if(inByte3==255){
176            //output base position
177            Serial.println(Basepos);
178          }
179          else if(inByte3!=-1){
180               val=map(inByte3, 1, 254, -5,5);  // was -5 to +5 originally
181               if (val < 2 && val > -2){
182                val=0;
183                //Serial.flush();
184                digitalWrite(SLEEP,LOW);
185              }
186               // Set the direction change LOW to HIGH to go in opposite
                      direction
187               if(val>0)
188               {
189                  digitalWrite(DIR, LOW);
190               }
191               else if (val<0)
192               {
193                  digitalWrite(DIR, HIGH);
194               }
195
196               //running at 200 steps per revolution
197               digitalWrite(MS1, 0);  // Set state of MS1 based on the returned
                      value from the MS1_MODE() switch statement.
198               digitalWrite(MS2, 0);  // Set state of MS2 based on the returned
                      value from the MS2_MODE() switch statement.
199                // Set the Sleep mode to AWAKE.
200
201
202               //determine if goal position will be outside of our bounds
203               if (!(val==0)){
```

```
204                 if(Basepos+val >= Baselimits[1]){
205                   //val=Baselimits[1]-Basepos;
206                 }
207                 else if(Basepos+val>=Baselimits[0]){
208                  // val=Basepos-Baselimits[0];
209                 }
210                   digitalWrite(SLEEP, HIGH);
211                   int i = 0;
212                   while(i<(abs(val)))
213                   {
214                     digitalWrite(STEP, LOW);
215                     digitalWrite(STEP, HIGH);
216                     //delayMicroseconds(1600*4);
217                     delay(20);
218                     //delayMicroseconds(1600)                    ;
219                     i++;
220                   }
221                   Basepos=Basepos+val;
222                   //Serial.flush();
223                 }
224                 //delay(30);
225             }// number byte interpretation
226         } // base
227
228
229
230         }// arm effector level
231
232
233       }// serial level
234        //delay(10);
235        //Serial.print(EEpos, DEC);
236       // Serial.print(" ");
237       // Serial.print(Elbowpos,DEC);
238 }// main loop read level
```

H.4 LabVIEW Joystick Command


Turn off motors, close port
Tab Control
Select Joystick
Page 1
Val(Sgnl)
Tab Control
Open Comm
Com Port 2
Strings[]
Select Joystick
None
9600
8
1.0
button 1
button 2
button 3
button 4
X axis
Y axis
Z axis
X axis rotation
Y axis rotation
Z axis rotation
Page 2
Val(Sgnl)
Tab Control
Press


caledjoystickoutput
0.03922
No Error
Joystick Not Found.
Continue" and restart the VI
Continue
True
STOP


Base Delay
STOP
No Error
No Error
No Error