

# 2017

## Bipedal Robot and Human Balance Test Bed | BRUTE



Eric Bryant

Matthew Kennedy

Ezana Mulugeta

Lisa Reynolds

Tiffany Hamstreet

Alexander Steele

Sponsor: Dr. Hunt

6/9/2017

# I. Executive Summary

## Objective Statement

Design, build, and test an automated balancing platform for Dr. Hunt's bipedal robot. The platform must be capable of dynamic input in three degrees of freedom and feedback control to monitor dynamic performance with a budget of 1800 USD by June 8, 2017.

## Final Status of Design Project

The platform is intended to test robot balance, which requires that the system be capable of the following:

- Rotational dynamic capabilities for maximum load with:
  - Angular displacement up to 7.5 degrees
  - Angular velocity up to 50 degrees/s
  - Step and oscillatory inputs over the dynamic range
  - Axes of rotation through robot ankle height as defined by project sponsor
- Horizontal dynamic capabilities for maximum load with:
  - Acceleration up to 5 m/s<sup>2</sup>
  - Velocity up to 1.5 m/s
  - Displacement up to 0.15 m
  - Step inputs only
- A user interface enables user inputs and extraction of dynamics data

The team successfully delivered a platform capable of one fully functional rotational degree of freedom. A concentric ring design was chosen to facilitate the axis of rotation above the platform surface through the robot's ankle joint. This design has the benefit of the frame providing some support for the weight of the platform and robot. Two rotational degrees of freedom are actuated by gear reduced motors and four-bar linkages.

A microcontroller with feedback provides real time control of the system from rotary encoders, and dynamics are controlled with a proportional-integral controller. A user interface enables dynamic inputs to the system, displays system dynamics in real time, and produces a data file with the system's dynamic input and output signals. Bearings and rails have been installed for the linear subsystem, and the platform has been built to accommodate addition of linear motion in the future.

## Key Performance Metrics

For step and oscillatory inputs on the enabled rotational degree of freedom, the average error of the system output over the required dynamic range is within design error. Friction was a significant challenge at the very low speeds and amplitudes of operation. As friction is nonlinear and variable, the response is not as smooth and repeatable as desired. All mechanical system components and sampling and data transfer speeds on electronic components were measured according to design criterion and were within the acceptable range.

# Table of Contents

I - Executive Summary	i
II - Team Member Roles and Contributions	iv
1.0 - Client Requirements	1
1.1 - About the sponsor	1
1.2 - Original Deliverables	1
1.3 - Modified Deliverables	1
1.4 - Further Design Considerations	1-2
2.0 - Conceptual Design Summary	2
2.1 - Design Concepts Considered	2-5
2.2 - User Interface and Data Management System	5-6
2.3 - Control System	6
3.0 - Subsystem Highlights	6-7
3.1 - Mechanical Subsystem – Frame	7-8
3.2 - Mechanical Subsystem – Motors & Linkages	8-9
3.3 - Controls Subsystem – Controller Design	9-14
3.4 - Controls Subsystem – User Interface	14-15
4.0 - Performance Summary	15
4.1 - Mechanical Subsystem Performance	15-17
4.2 - Control Subsystem Performance	17-20
5.0 - Final Status	20-21

6.0 - References	22
7.0 - Appendix A – Supplemental Images	23
8.0 - Appendix B – Catalog of Design Artifacts	24-25
9.0 - Appendix C – Concept Analysis	26-28
10.0 - Appendix D – System-level Design and Performance Matrices	29-30
11.0 - Appendix E – Mechanical Subsystem	31-32
12.0 - Appendix F – Controls Subsystem	33-35
13.0 - Appendix G – User Interface	36-39
14.0 - Appendix H – Bill of Materials	40

## II. Team Member Roles and Contributions

Eric Bryant: [erbryant@pdx.edu](mailto:erbryant@pdx.edu)

Controls design and testing. Linux operating system installation on Beaglebone board. Contributed in mechanical system prototyping and ideation.

Tiffany Hamstreet: [tstager@pdx.edu](mailto:tstager@pdx.edu)

Atmel Xmega research and programming, encoder error and sample time analysis, developed Excel tool for 4 bar linkage dynamic model, interfaced Xmega controls with UI and mechanical system, extensive data collection and assistance with control system. Completed all writing and catalog materials for these topics, and created all figures for Controls section and one for Mechanical section.

Matthew Kennedy: [mdk3@pdx.edu](mailto:mdk3@pdx.edu)

Embedded Linux board research. Raspberry Pi setup and programming including: ssh and ssh File Transfer Protocol server setup. User interface development using node.js and socket.io. Raspberry pi interfaced with Xmega over Serial Peripheral Interface (SPI) bus and general purpose input/output (GPIO) pins.

Ezana Mulugeta: [mulugeta@pdx.edu](mailto:mulugeta@pdx.edu)

Researched engineering analysis and created a MathCAD file for sizing power requirements. Helped machine and assemble the platform.

Lisa Reynolds: [lar8@pdx.edu](mailto:lar8@pdx.edu)

Scribe for team meetings. Machine shop helper. DIY linear actuator research and development. Product photographer. Document editing. Creative development for poster and presentations.

Alex Steele: [ajmar@pdx.edu](mailto:ajmar@pdx.edu)

Researched and created the 3D design of the current platform and the previous iterations. Sourced parts for the platform, then machined and assembled the frame and platform. Designed and created 3D models for linkages, encoder brackets, and motor brackets, then 3D printed them. Found the equations of motion for a four bar linkage system. Did the mechanical testing and physical measurements for the mechanical side of the platform. Finalized the layout, formatting, and theme for the reports.

# 1.0 – Client Requirements

## 1.1 – About the Sponsor

The project sponsor Dr. Hunt is a professor and head of the Agile and Adaptive Robotics lab at Portland State University. His work involves using principles of biology to develop robotic technology. The goal of this capstone project was to deliver a platform that would aid in testing the balance of a bipedal robot.

## 1.2 – Original Deliverables

The original client requirements detailed a platform that would test a bipedal robot weighing 80 kg at a height of 75-150 cm. The platform needed to perform 3-6 degrees of freedom with a minimum of two linear translations and one rotation about the sagittal plane. Physical responses of the system and center of pressure needed to be recorded. The user interface would allow for specification of the direction, type, speed, and duration of platform motion and dynamics data would be stored. Ideally, the platform would allow for sway referencing.

## 1.3 – Modified Deliverables

During meetings with the project sponsor, project deliverables were solidified, and modifications of project scope agreed upon. It was decided that constructing a platform capable of testing an 80kg robot was not feasible within the period and given budget. Instead, the team was asked to create a system capable of testing a 9 kg robot, with a design that could be scaled up in the future. One linear translation and two rotational degrees of freedom were specified that would function as follows:

- Rotational dynamic capabilities for maximum load with:
  - Angular displacement up to 7.5 degrees
  - Angular velocity up to 50 degrees/s
  - Step and oscillatory inputs over the dynamic range
  - Axes of rotation through robot ankle
- Horizontal dynamic capabilities for maximum load with:
  - Acceleration up to 5 m/s<sup>2</sup>
  - Velocity up to 1.5 m/s
  - Displacement up to 0.15 m
  - Step inputs only
- A user interface enables user inputs and extraction of platform performance data
- Oscillation perpendicular to the direction of translation is most desired

## 1.4 – Further Design Considerations

Investigation into the design problem and operating environment indicated that the following features are desirable, and were taken into account where possible:

- Data is displayed to the user in real time
- Design is modular, may be constructed in phases and enables future modification

- Device is portable via lifting (total weight under 30 lbs.) or wheels and/or easily disassembled (modular components each weighing less than 30 lbs.)
- The user may input a data file to define a particular waveform or set of inputs
- The three degrees of freedom may be operated simultaneously for complex motion
- Safety measures are present to prevent robot damage in the event that the robot falls off the platform
- Total expenditure is less than the specified budget

The system level requirement-measurement matrix and updated performance measures developed during the opportunity development phase are shown in Fig. D-1 and Table D-1 respectively. The complete project bill of materials is shown in Appendix H.

## 2.0 - Conceptual Design Summary

A catalog of design artifacts will be provided to the instructor and project sponsor, and will henceforth be referred to as the “catalog.” Some items referenced in this paper are included in the catalog but not in this paper. A complete table of contents for the catalog is shown in Appendix B.

### 2.1 – Design Concepts Considered

#### Stewart Platform

The Stewart platform is a well-established design used in a variety of applications. An example of a Stewart platform is shown in Fig. 1. Six linear actuators work in tandem to support and move the platform. This design allows for more degrees of freedom than specified by the customer. The added degrees of freedom add unwanted complexity to the dynamic model of the system. Another issue is the cost of six actuators, which would need to have the power to support and move the loaded platform.

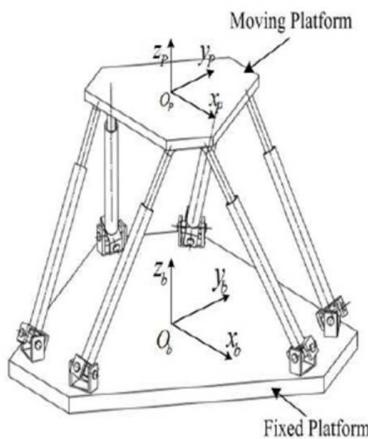


Figure 1 - Example of Stewart platform

## Central Supporting Pillar

The central supporting pillar design balances the platform on a central joint with two actuators acting as support at two edges. This design reduces the power requirements of the actuators because the central pillar supports the weight of the load. Additionally, the dynamic model is simplified due to each actuator's being constrained into a specific angular degree of freedom. From the construction of an initial physical prototype of this system, shown in Fig. 2, and with further research it was determined that a complicated central linkage and actuator joints would be needed to align the axes of rotation. These linkages would add increased noise in the play of each linkage making fine motor control difficult.

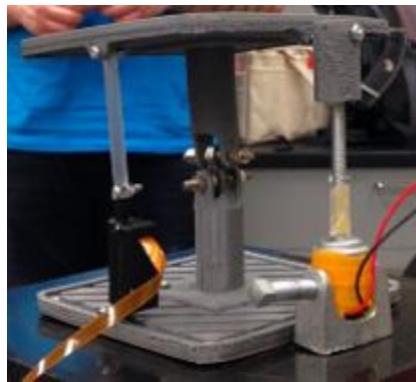


Figure 2 - Simple working prototype of central pillar design concept

## Concentric Rings

In the concentric ring design, the platform is suspended in a frame connected horizontally by bearings allowing one angular degree of freedom. This frame is embedded in a second frame by bearings perpendicular to the alignment of the inner bearings. A preliminary CAD model of this design is shown in Fig. 3. This design has a simple dynamic model and constrains movement to only the two desired angular degrees of freedom. Using bearings as linkages is expected to significantly reduce vibration and slip in the system. Furthermore, the axis of rotation is located at the test robot's ankle joint, which aligns with research methods for human balance testing.

A decision matrix was used to help compare the considered designs, shown in Table C-1. Of these three designs, the concentric rings design was selected because the frame instead of by the actuators supports the load, its dynamic model is simple, and it places the axes of rotation at the most desirable location. CAD software was used to visualize potential issues and compile a subsystem bill of materials. This analysis concludes that this design is implementable, within budget, and provides the most robust design solution for angular movement of the platform.



Figure 3 - 3D model of complete design concept

## Rotational Actuation

To provide actuation for angular motion two design options were considered: linear actuators and four-bar linkages. The required speed and torque for the drive systems of the central pillar and concentric rings models were calculated using MathCad. The calculations are conservative, placing the robot at a location on the platform resulting in the greatest moment arm relative to the location of the drive system. The results, shown in Table 1, were used to size the lead screws, motors, and power requirements. The complete calculations are provided in a MathCad document available in the catalog.

Table 1 - Motor sizing calculation results

Model	Drive Type	Peak Torque (Nm)	Speed (RPM)	Peak Power (W)
Center Pillar	Linear Actuator	0.132	2952.4	240
Concentric Circle	Four-bar linkage	2.936	10.3	48

## Linear Actuator

Initial platform design concepts included using a linear actuator to induce rotational movement of the platform. Because linear actuators capable of the speed and torque requirements are expensive, designing a linear actuator was considered. A combination of power screw and motor that would fit the size and cost constraints could not be found. Furthermore, the interface between the linear actuator and the platform would require multiple degrees of freedom at the base and attachment point, and would have added problematic play to the system.

## Four-Bar Linkages

The tradeoff between torque, speed, and the power requirement facilitated selection of the motor and linkage actuator model for angular motion. The peak power required in this model, 48W, is reasonably achieved with a motor and power supply that will be drawing 6A maximum at a 12V reference.

To aid in linkage design, a tool was created in Excel using the model shown in Fig. C-2 and equations for 4-bar linkage shown in the chapter of “Kinematic Synthesis of Linkages” included in the catalog [6]. A fully labeled model and corresponding equations are shown in Fig. C-1. The tool, included in the catalog with a sample shown in Table C-2, enables the user to input link lengths and critical mechanical dimensions and view what angles at the input shaft are required to produce desired angles at the platform shaft. In a 4-bar linkage system, the mechanical advantage is non-linear throughout the range of input angles. In this case, it would be advantageous to have the input at the motor shaft be significantly larger than the output at the platform shaft, to reduce the effects of friction in the system. However, as a low platform profile was prioritized, the link lengths and paths were constrained to provide a low input: output ratio between 1.36 and 2.15, throughout the displacement range.

## 2.2 – User interface and data management system

Customer requirements state that the platform must be controlled through a graphical user interface (GUI). The GUI will allow user input of step commands, sinusoidal commands, and a file containing complex waveform commands. Given the limited memory of controllers, the computer system must be able to do complex math to calculate command positions, velocities, and accelerations. In addition, the customer required platform performance data is stored during operation. To accomplish this task two-design approaches were considered: embedded Linux board or National Instruments (NI) DAQ in conjunction with Labview on a Windows PC.

### National Instruments and Labview

National Instruments offers a range of low-cost DAQ [1]. While all of the NI low-cost DAQs meet the required sampling frequency and the sampling resolution, they do not meet the requirement for analog output to provide calculated command positions, velocities, and accelerations to the controller. The most affordable option that meets the analog output requirement is the NI USB-6001 priced at 191 USD. The NI DAQ communicates over serial USB to a desktop computer running Labview. Once the DAQ is acquired, a Labview virtual instrument must be made to provide the necessary inputs.

### Embedded Linux board

There are many embedded Linux boards on the market capable of providing the resources required for this project. The most popular options are the Raspberry PI, Beaglebone Black (BBB), Arduino Yun, and Intel Galileo. The Raspberry PI lacks analog to digital converters in addition to only have one SPI interface. This severely limits the connectivity between the Raspberry PI and the controller and external sensors [2].

An embedded Linux system would operate independently of a desktop computer and would therefore require a built in video output. Both the Arduino Yun and Intel Galileo lack a basic video output and would require further interfacing into a more complicated system [2]. The BBB remains as the prime competitor in the embedded Linux systems category. The BBB costs 50 USD, making it three times less expensive than the NI DAQ. In addition to lower cost, the BBB is more customizable due to compatibility with a variety of programming languages [3].

### 2.3 – Control system

The microcontroller in this project will be used to control the dynamics of the system in real time. Therefore, the most important hardware features are performance, reliability, and versatility. Furthermore, choosing a controller with which group members have previous experience will help ensure the team's success. Initially, Arduino Mega 2560 and the Atmel Xmega A family of controllers were compared, and the Xmega was selected for its superiority in speed and real time performance. Another Arduino IDE compatible board based on the Atmel SAMD21 controller was recommended later in the process, and appeared to compete with the Xmega. An in-depth comparison of these boards was conducted as shown in Table C-3. The Xmega controller was ultimately selected as the lower risk option with a greater chance of success for this application based on research, member experience, and strong support from the faculty project mentor, Dr. Turcic.

## 3.0 – Subsystem Highlights

The primary subsystems in this design are detailed in Fig. 4. There are two mechanical subsystems: Frame and Actuation. The Frame includes two modules: one that provides two rotations and one that provides one direction of translation. The Actuation subsystem includes three actuators: two providing rotation and one providing translation. There are two controls subsystems: the Microcontroller, which provides sensor incorporation and real time control, and the User Interface, which provides the control signal for the system, and an interface for data inputs and display. Requirements-measurements matrices were developed for each subsystem, shown in their respective appendices, E-G, which give the important target requirements and performance measures to help ensure that customer requirements are met.

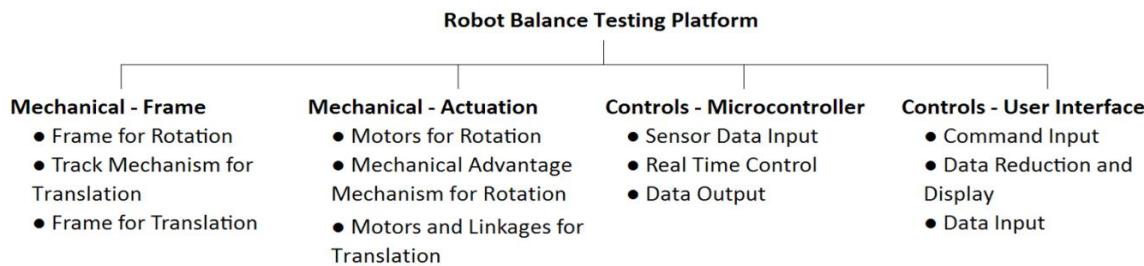


Figure 4- System decomposition schematic of the robot balance-testing platform shows the primary subsystems and their components

The interfaces between subsystems are shown graphically in Fig. 5 and associated performance measures and responsible parties are detailed in Table 2.

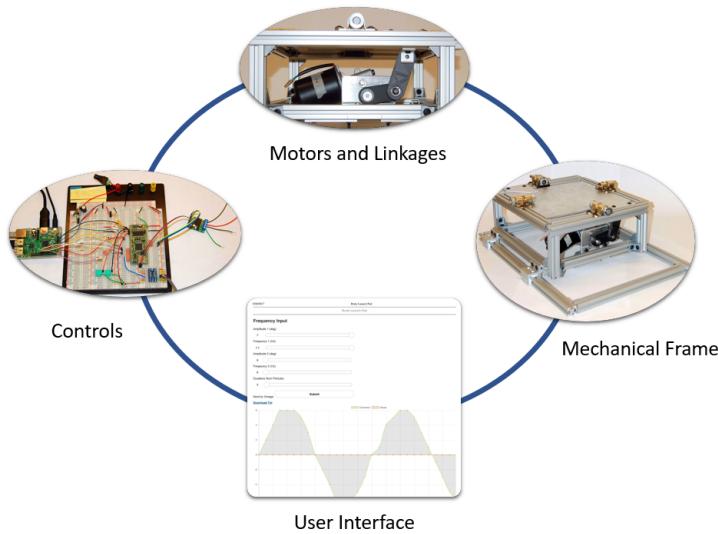


Figure 5 - Illustration of platform subsystems and interfaces between them

Table 2 - Interface definitions for robot balance testing platform

Interface Between	Interface Functions	Performance Measures	Responsible Team
Mechanical - Frame & Mechanical - Actuation	<ul style="list-style-type: none"> <li>Attach actuators to frame</li> <li>Transmit torque</li> <li>Provide desired DOF of relative motion</li> <li>Constrain motion in undesirable DOF</li> </ul>	<ul style="list-style-type: none"> <li>15 degrees displacement at platform; no relative motion at actuator base and translational frame</li> <li>Each motor is capable of 2.044Nm torque</li> <li>Translation 1.5 m/s; rotation 50 degrees/s</li> <li>Less than 1mm motion parallel to platform edge at max speed</li> </ul>	Mechanical
Mechanical - Frame & Controls - Microcontroller	<ul style="list-style-type: none"> <li>Attach sensors to frame</li> <li>Attach/support wires neatly and securely</li> </ul>	<ul style="list-style-type: none"> <li>No relative motion between sensors and frame</li> <li>Limited relative motion sufficient to prevent disconnection</li> </ul>	Mechanical
Mechanical - Actuation & Controls - Microcontroller	<ul style="list-style-type: none"> <li>Attach/support wires to motor, encoder inputs, and microcontroller</li> </ul>	<ul style="list-style-type: none"> <li>Limited relative motion sufficient to prevent disconnection</li> </ul>	Controls
Controls - Microcontroller & Controls - User Interface	<ul style="list-style-type: none"> <li>Attach wires for control signal and data transfer</li> </ul>	<ul style="list-style-type: none"> <li>Limited relative motion sufficient to prevent disconnection</li> </ul>	Controls

### 3.1 - Mechanical Subsystem – Frame

Several factors needed to be taken into account to design the platform frame. The frame needed to be rigid, support the weight requirements, be easily machined, and be made with off-the-shelf parts. To meet the off-the-shelf requirement 80/20 T-slot aluminum extrusion were chosen for the frame. This allows the frame to be lightweight, low cost, and easily modular.

It was important to reduce the weight of the frame to help reduce the added momentum from the rotational and linear motions. A 1-inch square T-slot extrusion was selected over the 1.5-inch square T-slot extrusion, which is stronger but heavier. Both sizes were capable of meeting the weight requirements, so the lighter and lower cost option was the clear choice.

The two degrees of rotation needed to be about the ankle of the robot, located above the

platform instead of level with the platform. This presented a design challenge since the linkages ideally would be in the plane of rotation to eliminate unwanted movement of the linkages while the platform is rotating. A concentric ring design was selected because it minimized the added complexity.

To allow for the two degrees of rotation, bearing blocks were used for rotational movement. A 0.5-inch high carbon steel shaft was used as the axle. A high-carbon steel shaft was selected over a similar aluminum shaft to help eliminate flexion resulting from weight added to the platform. The bearing blocks are specifically designed to eliminate mechanical noise, so mechanical noise from the motors will be minimized on the platform itself, resulting in higher accuracy of installed sensors.

### 3.2 – Mechanical Subsystem – Motors & Linkages

Motor selection and linkage design depended upon three main requirements. The subsystem needed to be capable of moving the desired load at the specified speed, be lightweight so the platform could be transported easily, and be compact enough to fit below the platform's low profile.

These requirements initially made motor selection within the given budget a challenge. The necessary torque and speed requirements were large enough that motors in the average cost class initially researched would not meet the requirements. Conversely, motors in the "high performance" class could easily meet the requirements but were cost prohibitive.

The solution was to direct drive the platform from the motor instead of using a lead screw to drive the motion. With a four bar linkage design, the motor could run at a relatively slower speed with a higher torque, characteristic of the lower cost motors initially considered. After calculating the needed torque and speed to move the platform with the addition of a four bar linkage, a low cost motor that greatly exceeded both the minimum speed and torque requirements under these new conditions was selected. This gave a high factor of safety and allowed the linkage design to be modified if necessary due to any unforeseen problems with the design.

The motor bracket and linkages were initially going to be machined from aluminum. Bearings were to be press fit into the links, however the precision needed to properly press-fit the bearings was not possible. Due to time constraints and machining issues, only one of the motor brackets could be machined from aluminum. The two motor links were 3D printed using a Markforged Mark 2 3D Printer, a first-of-its-kind continuous fiber 3D printer, which has the ability to give parts a strength to weight ratio better than aluminum.

After successfully printing the linkages and press fitting the bearings, it was found that the bearings initially purchased had too much play and needed to be replaced with something more precisely made. This provided an opportunity to redesign the linkages for 3D printing and add ribbing for extra strength from out of plane forces. It also was a chance to design the second motor bracket with added ribbing for 3D printing and print it as well.

The revised brackets were thicker - by 0.1 inches - to accommodate the higher precision bearings, which also added strength to out of plane forces. These bearings provided a much higher precision than the old bearings and removed play in the system that was causing the platform to be untestable. Figure 6 shows the final motor and linkage subsystem.

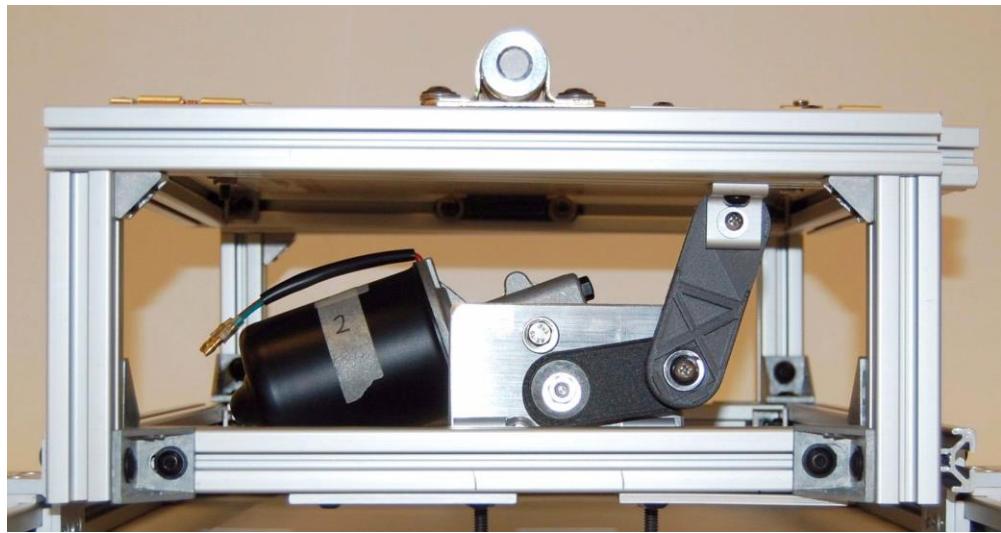


Figure 6 - Motor and linkage subsystem as installed in platform system

### 3.3 – Controls Subsystem – Controller Design

#### Intro to Controls Theory

Designing a controller is always a balancing act of system dynamics; speed may come at the expense of a smooth response while stability may sacrifice dynamic response. The designer must therefore prioritize the desired characteristics in the selection and design of the controller. Important characteristics include percent maximum overshoot (PMO), rise time, steady state error, and settling time. The speed of a system is measured as rise time. Rise time is the time for the system to rise from 10 percent to 90 percent of the input amplitude. Percent Maximum Overshoot (PMO) is the percent amount that the response exceeds the desired amplitude. Steady state error is the offset of the response from the desired amplitude after the system has normalized. Settling time is the amount of time that the system settles after a step input has been applied.

#### Controller Design Goals

The design goals set forth in the concept design phase of project development were to achieve an angular displacement accurate to 0.5 degree (6.7% error) with a smooth response. Additionally, a minimized steady state error, PMO, and rise time were desired attributes of the controller.

#### Bode Design Attempts

The first step in designing the controller was to characterize the dynamics of the system. The dynamics of a system are described mathematically as the transfer function of the system, a single function that describes the system output over the system input. Frequency analysis was first attempted to find the transfer function of the system.

Frequency analysis is a powerful tool to characterize system dynamics by applying oscillatory inputs over a frequency range while observing the response of the system. The ratio of the output amplitude with respect to the input amplitude and phase lag of the system are plotted

with respect to frequency. These common frequency response plots are known as Bode plots, and are useful in extracting important system characteristics such as gain, damping ratio, and natural frequency.

The angular movement of the platform was controlled with position control, as position is directly measured via the encoders. Position control is inherently unstable without feedback therefore the platform could only safely be operated with a controller implemented to prevent damage to the mechanical system. The system with feedback is referred to as closed loop, and the block diagram for the platform system is shown in Figure 7.

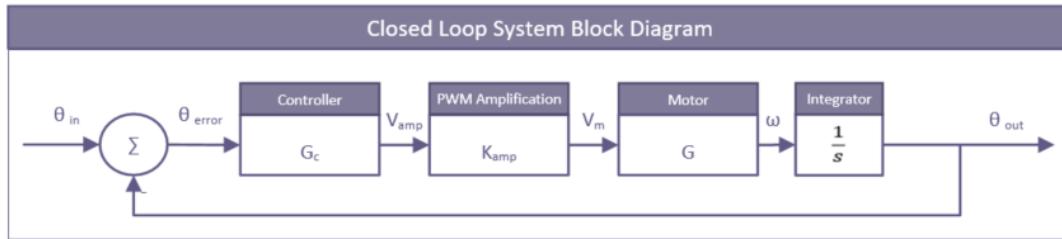


Figure 7 - Closed loop block diagram of the platform system with position feedback from the encoders.

The closed loop transfer function of this system is shown in Equation 1.

$$\frac{\theta_{out}}{\theta_{in}} = \frac{G_c K_{amp} K_t \omega_n^2}{s^3 + 2\zeta\omega_n s^2 + \omega_n^2 s + G_c K_{amp} K_t \omega_n^2}$$

Where  $G_c$  is the control equation  
 $K_t$  is the system gain  
 $K_{amp}$  is the amplification gain via PWM  
 $\omega_n$  is the natural frequency  
and  $\zeta$  is the damping ratio

(1)

A common way of determining the characteristics of a position-controlled system is to collect data over an extended frequency range using the simplest controller possible, frequently a proportional controller, fitting a curve to the frequency response plots, thereby determining the characteristics of the system. In a proportional controller, the control equation  $G_c$  is a gain denoted as  $K_p$ . These characteristics are used to build a theoretical model of the system to design a controller. This method also tested the effectiveness of a proportional controller.

### Motor Only Bode

The platform has a limited amplitude range and is susceptible to damage; therefore, frequency analysis was initially performed on a motor alone. In this way, an errant controller would not cause damage to the platform. This was believed to be a good approximation of the system because friction in the motor was the primary contributor of friction to the system. The bode plot

of this motor only system is shown in Fig. F-2 with the overlay of the fitted transfer function. This analysis provided insight into the characteristics of the motor but controllers designed with the motor transfer function alone were found insufficient when implemented on the platform. They did, however, provide initial gain values to begin working with the platform without causing damage to the system.

### Motor/Platform Bode

A second frequency analysis was performed on the motor-platform system, shown in Fig. F-3. This second frequency analysis also proved to be insufficiently accurate due to the excessive nonlinear Coulomb friction present in such low speed and low amplitude operating conditions. The derived transfer function was useful in designing the initial iteration of the controller.

### System Controller Design

The proportional controller implemented during the closed loop feedback analysis provided valuable insight into the effectiveness of proportional control implementation on the system. By increasing the proportional controller gain,  $K_p$ , the effects of friction were attenuated. This was found to have limits due to the instability effects with higher gain values ( $K_p > 3$ ) which were especially noticeable at higher frequency operation. Additionally, the response was not sufficiently smooth. For these reasons, a proportional integral (PI) controller was designed.

The integrating factor in a PI controller adds phase to the system, which helps to smooth the response while adding stability. The controller equation for a continuous PI controller is shown in Equation 2.

$$G_c = K_p + \frac{K_i}{s} \quad (2)$$

where  $K_p$  is proportional gain and  $K_i$  is integral constant. In digital controls, the sample time is a critical variable involved in the control equation. The digital version, known as the difference equation, of the PI control equation is developed using a Tustin transform for the highest accuracy is shown in Equation 3.

$$\text{Control}_{\text{now}} = \text{Control}_{\text{last}} + \text{error}_{\text{now}} \left( K_p + K_i \frac{T_s}{2} \right) + \text{error}_{\text{last}} \left( K_i \frac{T_s}{2} - K_p \right) \quad (3)$$

where  $\text{Control}_{\text{now}}$  is the control voltage to the motor in the current control cycle,  $\text{Control}_{\text{last}}$  is the stored control voltage from the last cycle.  $\text{Error}_{\text{now}}$  is the position error from the current cycle, meaning the command position minus the actual position.  $\text{Error}_{\text{last}}$  is the stored error value from the last cycle; and  $T_s$  is the sample time, meaning the firing frequency of the control loop. The position error of the encoder reading decreases as the sample time increases and increases as the operating speed decreases, therefore the slowest possible sample times were chosen which would still provide a smooth response. A tool for estimating encoder error at different sample

times and operating speeds was developed in Excel, and is provided in the catalog, with a sample shown in Figure F-X.

The controller was tuned to have the highest proportional gain possible, to aid in overcoming friction, while avoiding instability. Due to especially impactful friction effects at the lowest operating speeds, it was found necessary to separate the operational frequency range into two bands and design two separate controllers.

The first band, designed for platform operational frequencies of 0.1 Hz to 0.3 Hz, was designed with the primary goal of overcoming the higher frictional effects in this lower frequency range. The controller in this range therefore had a larger K<sub>p</sub> value of two. The higher frequency range controller (platform operational frequencies of 0.4 Hz to 1.1 Hz) saw less of an effect from friction, but was increasingly prone to overshoot as operating frequency increased, so the K<sub>p</sub> was tuned lower to one. Both controllers had a very low K<sub>i</sub> of 0.05, as undesirable characteristics were observed at higher values of K<sub>i</sub> in both operating ranges. The low band controller was designed with a sample time of 0.05 seconds (20 Hz) and the high band controller with a sample time of 0.02 seconds (50 Hz), to get the highest performance while minimizing encoder error in each band.

### Microcontroller and sensor system development

Microcontroller programming occurred in parallel with user interface and mechanical design and assembly activities throughout the project. An Inertial Measurement Unit (IMU) was enabled initially, but was not used, as encoders were deemed sufficient for motor control and are more reliable and simpler to use. The fundamental microcontroller design was laid out in the conceptual design phase. Critical components for motor driving were enabled first, followed by motor control, then communication with the Raspberry Pi. Finally, conveniences and round up items such as buttons and sliders were added. Table 3 describes the methods used to enable critical features, roughly in the order in which they were enabled. Figure F-5 shows a graphical layout of Xmega hardware connections and background functionality.

Table 3 - Microcontroller and hardware layout with description of software enablement method; all hardware datasheets and manuals provided in the catalog

Feature	Method	Associated Hardware
System Clock	Default clock is 2MHz, so 32MHz clock was enabled	ATXmega128A1U
USART 2-wire serial communication	USART enabled on Port C, only TX (outward) communications required, critical for debug	ATXmega128A1U
Pulse Width Modulation (PWM) for Pulse Width Modulation (PWM) for Quadrature decode for each encoder	One timer on Port F with 2 compare channels Two timers total, one for each motor, one on Port D and one on Port E, to trigger decode events; Xmega logic is 3.3V, and encoder logic is 5V, so logic level converter was used	Pololu motor driver carrier VNH5019 Adafruit 8-channel Bi-directional Logic Level Converter TXB0108 Encoders AMT102-V
Timed interrupt for management of control loop	Overflow timer enabled on Port C and programmed so sampling frequency of control loop is easily modified	ATXmega128A1U
SPI 4-wire serial communication	SPI Slave enabled on Port C with two Data Ready pins on Port A for communication with Raspberry Pi	Raspberry Pi v1.2 B+
Buttons for manual positioning and slider for moving Xmega program from Manual Positioning Mode to	Programmed 5 inputs on Port A, 4 for each direction of the two motors, one for switching the program mode	SPST buttons and SPDT slider

### Microcontroller software functionality

The final platform system is programmed with software that is compatible with the graphical user interface. One additional program is provided in the catalog for direct programming of user inputs to the Xmega alone in the absence of the UI, which provides the code foundation and manual control capability for the second rotational degree of freedom. Thoroughly documented software is included in the catalog. Xmega software is written in embedded C using Atmel Studio as the development platform. A summary of Xmega code functionality is provided in the catalog.

### Implementation Challenges

The most time consuming challenges in the implementation of the microcontroller occurred at the interfaces with the user interface and mechanical system. Attempts to make the Serial Peripheral Interface (SPI) connection between the Xmega and the Beaglebone Black resulted in the destruction of two Beaglebones and one Xmega, which caused a two-week delay to the scheduled deadline for enabling communication between these systems. The switch was made to the Raspberry Pi (RPi), and 8-bit SPI was enabled very quickly at that time.

The next challenge was enabling 16-bit SPI, as more than 8-bit was required to transmit both the data and necessary control bits to allow effective two-way communication. Until the successful implementation of the 16-bit SPI, functionality on the Xmega could only be tested without the Raspberry Pi, so all the code was thoroughly tested prior to the attempts to interface. After 16-bit SPI, much of the microcontroller code involving the UI could not be tested without the RPi providing SPI inputs, so both UI and Xmega often had to interface untested code, making this work more challenging and time consuming.

In parallel, a number of challenges were faced at the interface between the microcontroller and the mechanical system. Because a position controlled system is not open loop stable due to friction and the platform operating amplitudes are so limited, the platform could not be safely

tested without at least very basic closed loop control. Closed loop control requires accurate position feedback from the encoders.

Encoders are comprised of an inner collet which grips the shaft on which it is installed and an outer frame which is affixed to the device. For the encoder to register a position change there must be relative motion between the collet and frame. Multiple brackets had to be designed to hold the encoder securely enough to provide feedback, and the shafts, which were expected to be stationary, but they were actually subject to nearly imperceptible movement, which had to be fixed. Due to the low operating amplitudes, the problems with encoder and shaft mounting were not readily apparent, so a lot of time was wasted trouble shooting code when the problem was truly related to hardware.

Final construction and testing of the mechanical system was significantly delayed, and was not ready for the scheduled interface with the control system, resulting in extra work and delayed completion and refinement of the controller subsystem. Ultimately, these delays and other challenges prevented the enablement of the second and third degree of freedom specified by the project sponsor.

### 3.4 - Controls Subsystem – User Interface

During the concept design phase, a BeagleBone Black (BBB) was selected as was mentioned in Section 3. During product development, a key issue was discovered with the BBB. Serial peripheral interface (SPI) bus was successfully set up between the BBB and a digital potentiometer. This step was done in parallel with setting up the Xmega SPI communication.

When both the Xmega SPI and the BBB spi were tested independently and working correctly independently, the two systems were integrated. However, when the grounds of the BBB and the Xmega were tied together the BBB stopped working. According to the BBB online wiki forums, the LED pattern indicated that we had a shorted component. A second BBB was purchased and careful precaution was taken but the issue repeated itself.

A Raspberry Pi (RPi) was selected to replace the BBB. The RPi was ruled out during the concept development stage due to its lower clock speed. However, it was later discovered that the operating frequency of the platform was much lower than theorized. This gave more leeway in the clock speed of the computer system to be used.

Development then began with the Raspberry Pi by first enabling a Secure Socket Shell (ssh) server and ssh file transfer protocol. SPI communication was then established with the Xmega in stages. First 8-bit SPI was enabled, followed by 16 bit SPI. A command structure was then developed for the systems to indicate the types of data being transferred. The data structure is depicted in Fig. G-1. The data structure allows for eight types of data to be transmitted from the Xmega to the RPi with values ranging from -8.99 to 8.99.

With communication handled, the user interface began to come together. While there are many web server languages available, the language needed to provide real time communication between client and server. Socket.io is a well-established real time engine for web servers. Node.js was selected as the web runtime language due to its large documentation library. Node.js also has optimized libraries specifically for ARM chip architecture, which the Raspberry Pi uses.

The web user interface allows a user to give the platform step or frequency inputs and provides a real time graph of command position vs actual position. The interface also provides a kill button in case an emergency arises. A screenshot of the user interface during platform operation is provided in Fig. G-2.

### Interface between UI and Microcontroller

For the completed system, the software functionality has three major components: Raspberry Pi: Server Side, Raspberry Pi: Client Side, and the Xmega Microcontroller. Figure 8 highlights the major components of the software communication protocol. Data between the Xmega and the server are handled via SPI transactions, while data between the server and the client are handled over websockets.

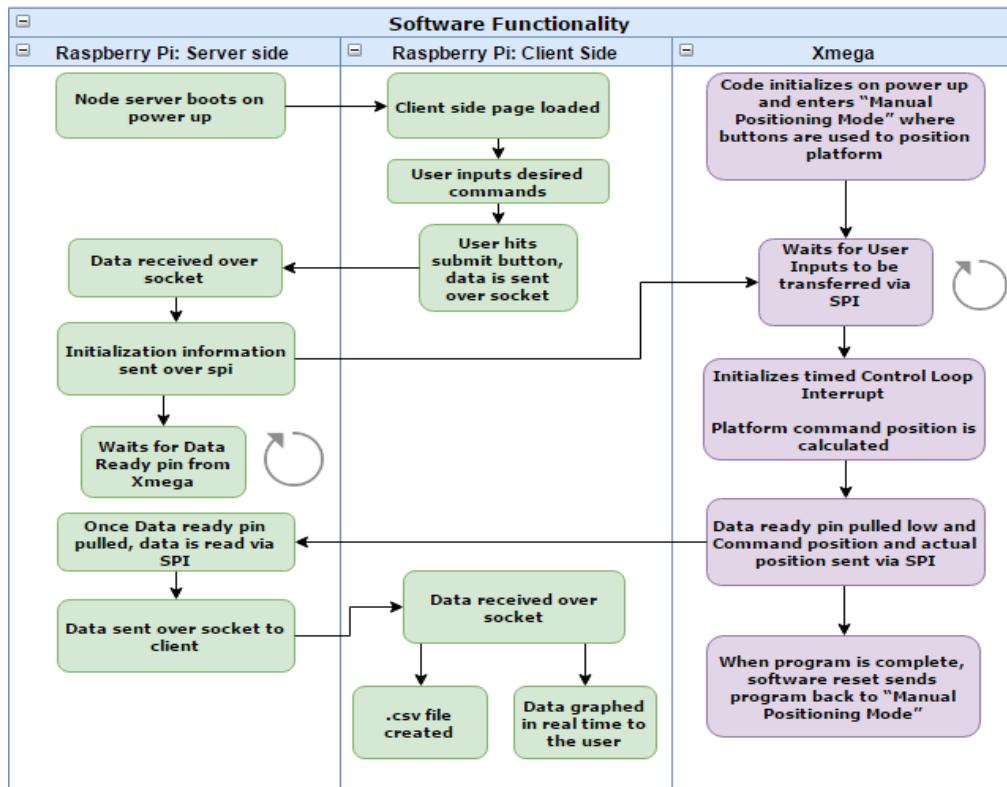


Figure 8 - Software communication protocol. RPi server and RPi client communicate over websockets, while RPi server and Xmega communicate over SPI

## 4.0 – Performance Summary

Performance of the system was analyzed on a subsystem basis but it should be noted that interactions between the subsystems played a key role on subsystem performance. For example, as will be shown below the significant mechanical friction in the system lead to difficulties in control system performance.

### 4.1 - Mechanical System Performance

Table 4 shows the measured performance of the mechanical system. Deflection was measured

using a deflection gauge, with 9.07 kg load applied to the end opposite of the side measuring. The results show that the outer platform performed better than the inner platform, but this is most likely due to the inner platforms reliance on the outer platform. In other words, any deflection the outer platform sees, the inner platform sees as well. Conversely, the outer platform is independent of the inner platform and sees almost no deflection from the application of the weight or the deflection of the inner platform.

User requirements dictated that the platform must be portable, have a reasonable footprint, and be within a safe height, so the size and weight of the platform needed to be measured. The platform dimensions were taken using a measuring tape and the footprint of the platform was then calculated using those dimensions. To ensure portability, the weight of the platform was measured using a digital 'fish scale.'

*Table 4 - Performance Measures for Mechanical Subsystem - Frame*

Performance Metric	Units	Minimum	Ideal	Maximum	Measurement Technique	Measured
Outer platform deflection	mm	0	0	5	The deflection gauge measures the deflection of the outer platform at the furthest most point opposite to the load, which is also at the furthest most point, to determine the maximum amount of deflection for the desired weight applied.	2.6771
Inner platform deflection	mm	0	0	5	The deflection gauge measures the deflection of the outer platform at the furthest most point opposite to the load, which is also at the furthest most point, to determine the maximum amount of deflection for the desired weight applied.	4.3956
Platform weight	Kg	4.5	9.1	15.9	Platform weight taken using a digital fish scale	9.525
Platform Area	m^2	0.1	0.15	0.28	Platform area determined using a measuring tape	0.2597
Platform Height	cm	7.75	15.25	25.25	Platform height determined using a measuring tape	17.78

Table 5 shows the mechanical slip of the bearings used in the linkages, measured using a deflection gauge for both in plane and out of plane forces. Backlash of the motor was also determined using a deflection gauge. While there was no way to correct for backlash from the internal gear reduction, it was an important measurement in determining how accurate the motor position could be.

Friction also played a large part in the ability to position the platform correctly. In order to reduce the amount of friction the motor and linkages had to overcome, oil impregnated bronze washers were used between the contact surfaces of the linkages, where friction would be the highest, and between the bushings where the platform would rotate. The performance metric used to characterize friction of the joints was the coefficient of friction of the thrust washers.

Table 5 - Performance Measures for Mechanical subsystem - Motor & Linkages

Performance Metric	Units	Minimum	Ideal	Maximum	Measurement Technique	Measured
Motor Backlash	mm	0	0	1	The deflection gauge will be used to measure backlash after a load is applied to the motor in either direction. This number is then halved and is shown in plus-minus notation.	+/- 0.1489
Motor Torque	N-m	0.122	3	20	Given from performance spec sheet	11.5
Washer coefficient of Friction	unitless	0	0	0.5	Given from performance spec sheet	0.10 to 0.25
In-plane bearing deflection	mm	0	0	0.5	The deflection gauge measures the deflection of the bearings when a load is applied in-plane	0.0254
Out-of-plane bearing deflection	mm	0	0	0.25	The deflection gauge measures the deflection of the bearings when a load is applied out-of-plane	0.00762

#### 4.2 - Control System Performance

As frictional effects were a concern because of early testing observations, the friction effects of the motor alone, and motor and platform system were compared. Mechanical parts that contributed to friction include bearings, linkages, as well as backlash in the motor. The motor alone and motor and platform system were each tested with a closed loop proportional controller with a gain of one, and the same sample time for direct comparison. For the motor-only test, the input amplitude was 15 degrees, the theoretical amplitude that would correspond to an output amplitude of 5 degrees at the motor shaft, and the data was post-processed for comparison. Fig. 9 shows the frictional effects present in the motor alone as well as the added friction that is attributed to mechanical system.

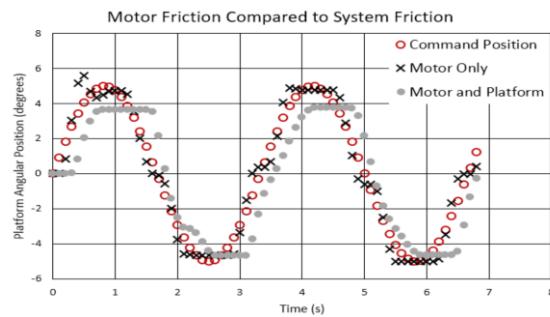


Figure 9 - The motor alone and motor and platform system were each tested with a closed loop proportional controller and  $K_p = 1$  to compare the effects of friction. The motor only was given an input amplitude of 15 degrees, which is the theoretical input amplitude

With frictional effects in mind, the final controller design was split into two separate controllers; one for the lower frequency range (0.1 Hz - 0.3 Hz) and one for the higher frequency range (0.4 Hz- 1.1 Hz). Fig. 10a shows the response of the system to an input of 0.3 Hz with an amplitude of 5 degrees. As expected, the frictional effects are more pronounced at this lower frequency range. This can be observed in the bunching of output data points where the friction build-up

occurs, visible as jerks or delays in the physical system. The increased proportional gain component of the PI controller helped to overcome friction but some of the effect remains. The angular amplitude ratio of 1.03 is an indication of the accuracy of the controller in this range. Also of note is the fast response seen in the short lag period.

The response of the high band controller is shown in Fig. 10b. The effects of friction are less in this range; therefore, the response is smoother though friction is still apparent in the flattening at the maximum and minimums of the response. In this range, the system lag is more pronounced but within the design constraints.

The step response in Fig. 10c shows the system's response to an instantaneous input. The rise time was found to be 0.3 seconds with a steady state error of 6 percent or 0.3 degrees. The steady state error is attributable to the friction in the motor, which the small integral coefficient is too small to overcome at slow speeds. The design requirement of an angular displacement within 0.5 degrees is met in the step response.

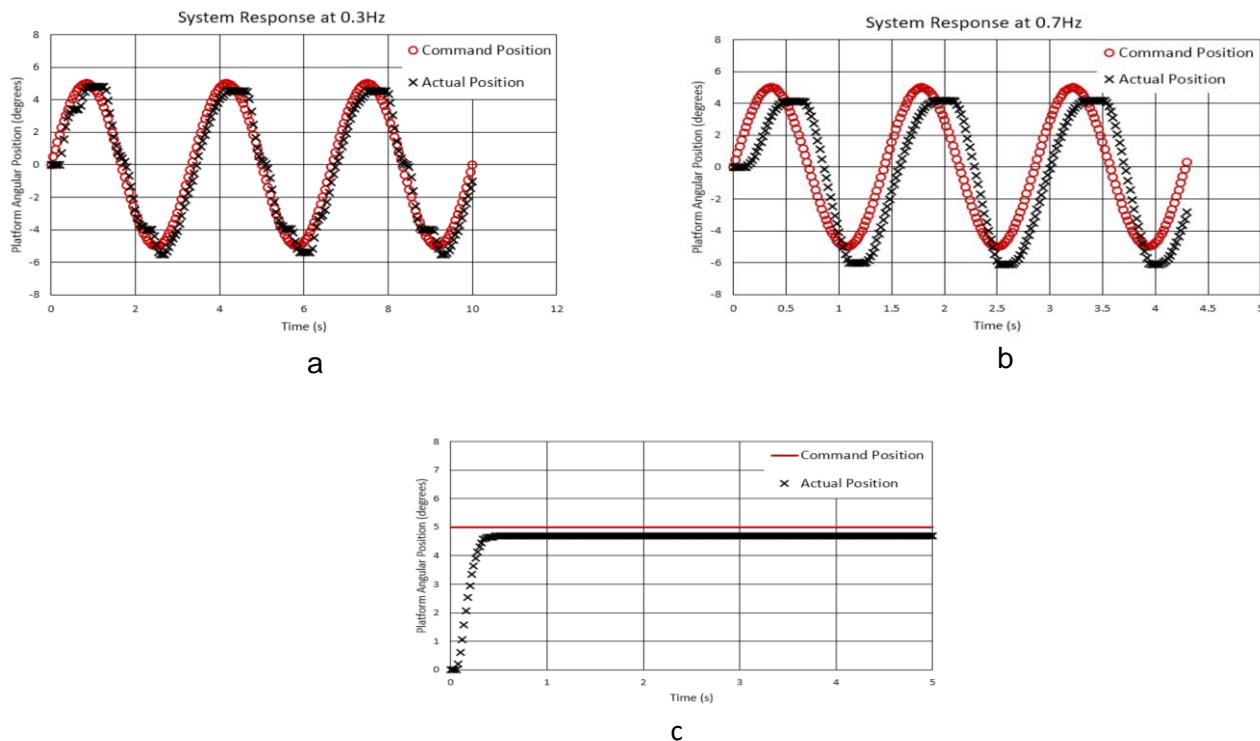
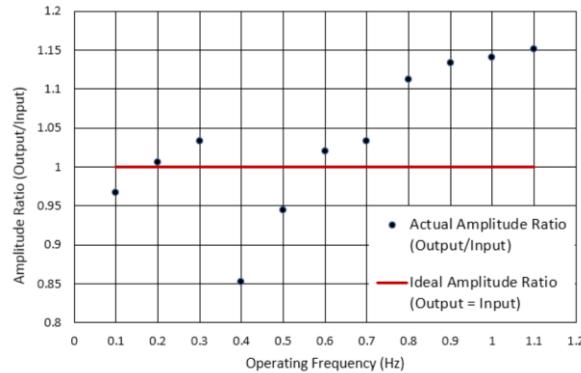


Figure 10 - Samples of low and high band oscillation response and step response with proof load a) 0.3 Hz response with mean standard error of 0.71 and amplitude ratio of 1.03. b) 0.7 Hz response with mean standard error 2.12 and amplitude ratio of 1.03. c) Step Step Response has 6.0% steady state error and rise time of 0.3s

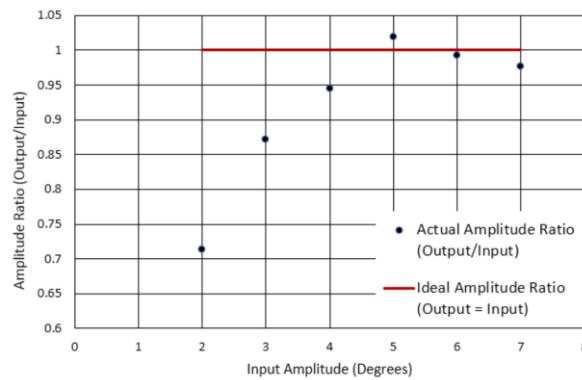
The accuracy of the controller is quantified in Fig. 11(a) across the entire operational frequency range. The average amplitude error was calculated to be 3.6 percent with mean standard error 1.79, which meets the angular displacement requirements. The controller has a robust response through the required operational frequency range.

The dominant factor of the system that the controls system had to overcome was the effect of friction. These effects are more pronounced at lower amplitudes. The frictional effects on the system at various input amplitudes were analyzed and shown in Fig. 11(b). At the lowest achievable amplitude of two degrees, the amplitude was 70 percent of the input due to friction. The response of the system is most accurate operating at an amplitude of 6 degrees.

The sampling rate is at minimum 62.5 times the Nyquist frequency over the operating range, so there was no chance of aliasing effects interfering with interpretation of the data. Despite frictional effects, the controller design goals shown in Table 6 were achieved.



a) Average amplitude error 3.6% over operating range, with mean standard error 1.79



b) Average amplitude error of 8.1% over operating range, with mean standard error 1.67

Figure 11 - Shows the system response over a) the entire range of operating frequencies and b) the entire range of input amplitudes. Note that the system is unable to overcome friction at an input amplitude of 1 degree, so there is no response at that point.

Table 6 - Performance Measures for Controls - Microcontroller subsystem

Performance Metric	Units	Minimum	Ideal	Maximum	Measurement Technique	Measured
Sampling rate	(Hz/Hz)	1	5	100	The Nyquist frequency (2x the operating frequency) will be compared to the sampling rate (sampling rate/Nyquist frequency) for the lowest operating frequency in each band	100 for low band 62.5 for high band
Average Mean Standard Error	degrees	0	0	2	Encoder	1.79
Average Amplitude ratio (Actual/command)	Unitless	0	1	1.2	Encoder	1.00 for low band 1.05 for high band

## User Interface

Performance of the user interface was measured by the maximum speed at which SPI transactions could be conducted between the Raspberry Pi and the Xmega. Both systems were set up with incrementing counters and SPI operations enabled. The Xmega incremented its

counter within a timed interrupt, so its transaction frequency was known. The Raspberry Pi incremented its counter at its system clock speed. The counter was transferred over SPI between the two systems. The Raspberry Pi had a pin set as a listener to detect when the Xmega was ready for a transaction. The datasheet indicated that the theoretical maximum speed for this listener is 1 kHz. The maximum frequency of data communication was found, with an oscilloscope, to be 650 Hz for the Raspberry Pi, when the Xmega transaction frequency was set at 2000 Hz. This corresponds to 1.5 ms maximum data transfer speed, as shown in Table 7.

The performance of the web server was measured using the Linux command ‘top’ while the web server was running and while the web server was pushing data over the socket to client side. Top is a Linux command line program that provides dynamics real-time view of running system processes. Top is analogous to Windows Task manager or OS X Activity Monitor.

A baseline of system usage was recorded by booting the Raspberry Pi and running Top without the web server running. Under the baseline condition, 3% of the CPU was being used and 70% of the available memory was being used. When initializing the node.js server the CPU usage spiked to 60% for approximately a second then returned to roughly 4% and the memory usage rose to 78%. During operation of the platform, the draw on the CPU is 88.6% and the ram used is 30%, shown in Fig. G-3.

*Table 7 - Performance Measures for Controls - GUI subsystem*

Performance Metric	Units	Min	Ideal	Max	Measurement Technique	Measured Value
Minimum data transfer rate between Raspberry Pi and Microcontroller	ms	1	500	1000	Oscilloscope used to measure Chip select pulses	1.5 ms
GUI doesn't consume all of systems resources	%ram % processor	10	40	75	Log system performance data by performing stress test on system	90% CPU 30% RAM

## 5.0 - Final Status

The platform provides one rotational degree of freedom with the desired custom requirements for displacement and velocity. The rotational degree of freedom is offset from the platforms surface through the robots ankle. The user interface allows for frequency and step inputs and provides performance feedback to the user. The microcontroller and controller work to provide control of the motor. The project came in just under budget.

### Future Work

#### Linear Translation

Due to time and budget constraints, the team could not complete the linear translation subsystem. An aluminum T-extrusion track and bearings were installed with the intention of implementing an air-pressured pneumatic cylinder to provide thrust to subject the platform to impulse inputs. The pneumatic cylinder would be controlled via solenoid valve and would be relatively easy to add to the current state, due to the controls being decoupled from the angular module.

## Additional Rotational Degree of Freedom

The team encountered time-consuming issues with the first rotational degree of freedom, the troubleshooting of which critically delayed implementation and refinement of controls. Although a second motor and linkages were installed to enable a second degree of rotational freedom, these setbacks did not leave enough time for controls to be fully implemented for more than one rotational degree of freedom. Code is provided to implement the second motor and sensors in hardware, and future teams will be able to incorporate it into the existing control code to enable the second degree of freedom, using the motor and linkages installed for this purpose. The user interface is setup to accommodate the second degree of freedom.

## Friction Reduction/Control System Improvements

The very low operating frequencies and amplitudes make the current system very difficult to control, as friction is nonlinear and varies depending on heat, speed, and other factors. To better control the system in the future, options should be explored to reduce friction. It was shown that the motors themselves are a significant source of friction, and should be replaced with higher performance motors with lower internal friction. Furthermore, it would be beneficial to maximize the speed and displacement of the motor output shaft by altering the linkage design to one with higher mechanical advantage, which may require increasing the height of the platform or reorganizing the motors and links in the space below the platform.

## Pressure Sensor Installation

In initial discussions with the project sponsor, it was agreed upon that a subsystem allowing the user to get data regarding the center of pressure of the robot during perturbations was not necessary to include in the scope of the project this academic year. The team designed the platform to account for the future addition of a pressure plate to gather this data. Future teams will need to research pressure plates that will accomplish the desired outcome and install them to the platform in a manner that allows the plate to be removed and rotated 90 degrees.

## Full Scale Platform

During the beginning of winter term, our sponsor agreed that the full-scale platform meant to test an 80kg robot, might not be feasible to deliver within a six-month period. The team was instead tasked with providing a scaled down version of the platform, which would be used to test a 20 lb. robot. Using design elements and control systems developed for this capstone project, future teams will be able to more easily develop a platform to test the full-scale robot.

## 6.0 – References

1. NI Low-Cost Data Acquisition Family - National Instruments. (2017). Ni.com. Retrieved 16 March 2017, from <http://www.ni.com/low-cost-daq/>
2. (2017). Cdn-learn.adafruit.com. Retrieved 16 March 2017, from <https://cdn-learn.adafruit.com/downloads/pdf/embedded-linux-board-comparison.pdf>
3. Beaglebone Black, H. (2014). How to Choose Between Raspberry Pi and Beaglebone Black | Make. Make: DIY Projects and Ideas for Makers. Retrieved 16 March 2017, from <http://makezine.com/2014/02/25/how-to-choose-the-right-platform-raspberry-pi-or-beaglebone-black/>
4. Hartenberg, Richard S., and Jacques Denavit. "Algebraic Methods of Synthesis Using Displacement Equations." Kinematic synthesis of linkages. N.Y.: McGraw-Hill, 1964. 295-303. Print.

## 7.0 – Appendix A: Supplemental Images

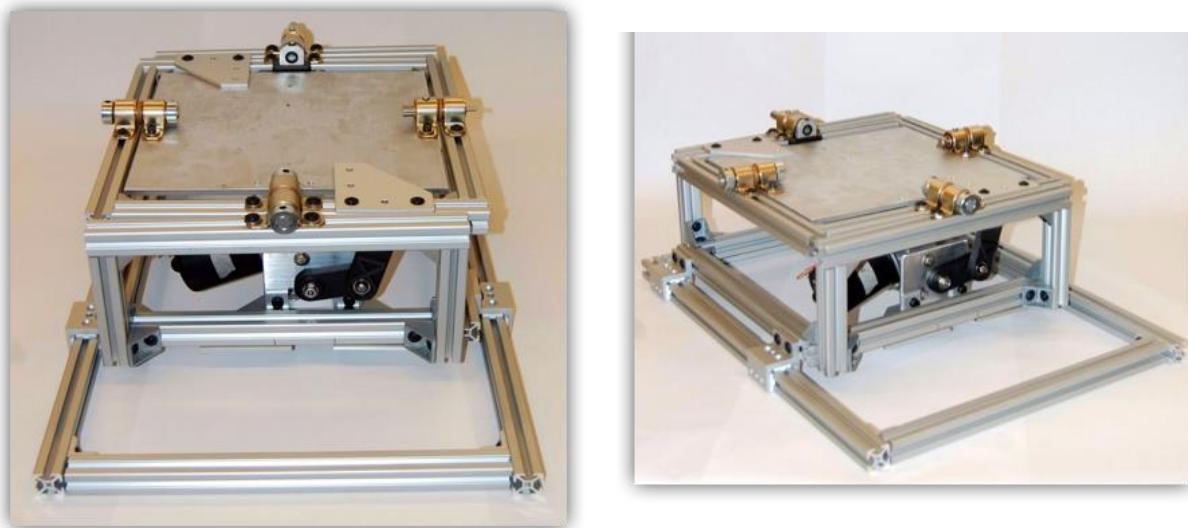


Figure A-1 - Pictures of mechanical frame system

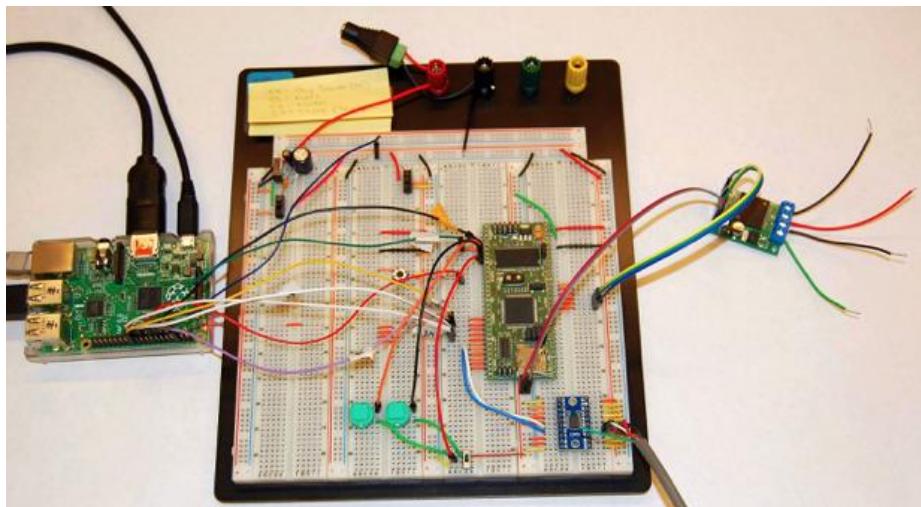


Figure A-2. Pictures of Raspberry Pi and Xmega interface

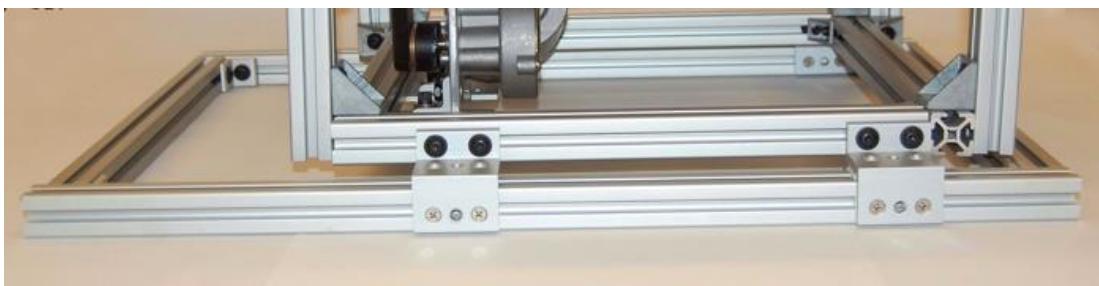


Figure A-3. Picture of linear slide system for future implementation of linear displacement.

## 8.0 – Appendix B: Catalog of Design Artifacts

The following items are provided in the Catalog of Design Artifacts. This list serves effectively as a table of contents for the supplementary documentation

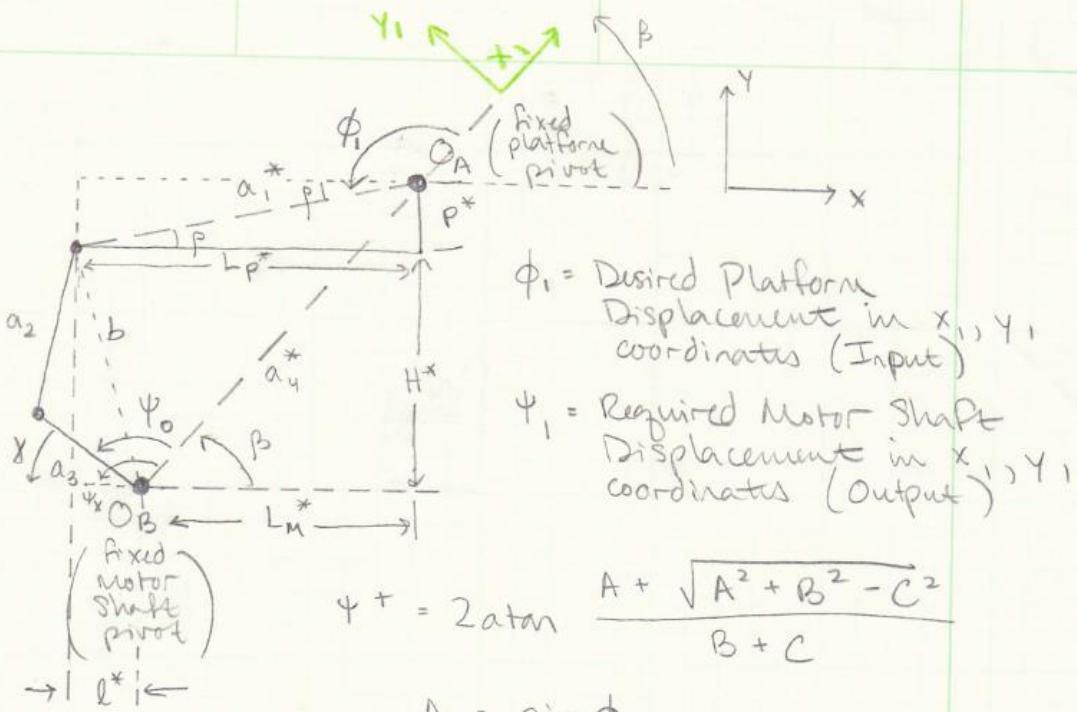
- Final Design - Datasheets, Software, Tools and Instructions, CAD files, and BOM
  - Instructional, Functionality, Useful Tools and Software
    - Xmega Basics Instructional
    - RPi/Xmega Instructional - General System Operation
    - Xmega C code summary of functionality
    - Xmega C code and project file for platform/UI
    - Xmega C code and project file for platform direct programming
    - Xmega hardware connections figure
    - Encoder Error and Sample Time Investigation Tool
    - 4-Bar Linkage Design Tool
    - 4-Bar Linkage Reference: Chapter 10 of “Kinematic Synthesis of Linkages”
    - Encoder Error Estimation Tool
  - Datasheets and Resources for Hardware
    - ATXmega128A1U datasheet
    - ATXmega128A1U manual
    - AVR Guide to C in Xmega Application Note
    - Xmega Event System Application Note
    - Xmega Timer/Counters and PWM Application Note
    - Xmega SPI Application Note
    - 8bit bidirectional logic level converter Datasheet
    - Encoder AMT10-V Datasheet
    - Motor Driver VNH5019 Datasheet
    - Raspberry Pi Datasheet
  - Final System CAD Files
    - Mechanical System CAD file
    - Linkage Mastercam files
    - Encoder holder CAD file
- Client/market requirement
  - Proposal from ME 491
  - Confirmation Email with Dr. Hunt
    - Requirements
    - Budget
    - BOM
  - System R-M Matrix
  - Initial project description
- Conceptual design documents
  - All intermediate CAD files
  - Linear actuator sketches

- o Linkage equations
  - o MathCAD file for sizing motor and linear actuator
- Additional Subsystem Documentation
  - o Subsystem R-M Matrices
  - o Performance Measurements
- Team communications (meetings)
  - o Mentor meeting minutes
  - o Agendas
  - o Gantt Charts
  - o Team contract
- Prior reports
  - o Winter Term Report
  - o Progress presentation
  - o Final Presentation
  - o Poster
  - o Website blurb

## 9.0 – Appendix C: Concept Analysis

Table C-1. Decision matrix used to decide upon frame design

Metric	Weight	Stewart		Concentric		Central Pillar	
		Rating	Total	Rating	Total	Rating	Total
Extra degrees of freedom	1	5	5	0	0	3	3
Mostly pre fab materials	3	3	9	4	12	2	6
Ease of linear translation	3	2	6	5	15	3	9
Less play anticipated	3	3	9	5	15	2	6
Ease of controls	4	1	4	4	16	3	12
Cost	5	0	0	4	20	4	20
Likelihood of success	6	2	12	5	30	3	18
Total			45		108		74



$$\rho = \tan^{-1} \left( \frac{P^*}{L_p^*} \right)$$

$$\beta = \tan^{-1} \left( \frac{H^* + P^*}{L_m^*} \right)$$

↳ transformation angle between  $x_1, y_1$  &  $x_1, y_1$  reference

$$\Psi_x = \phi_x = 180^\circ - \beta$$

$$\Psi_0 = \Psi_x - \gamma$$

$$\gamma = \Psi_x - \Psi_0$$

Figure C-1. Equations and reference diagram for 4-bar linkage which correspond to variables used in 4-bar linkage dynamics tool in the catalog

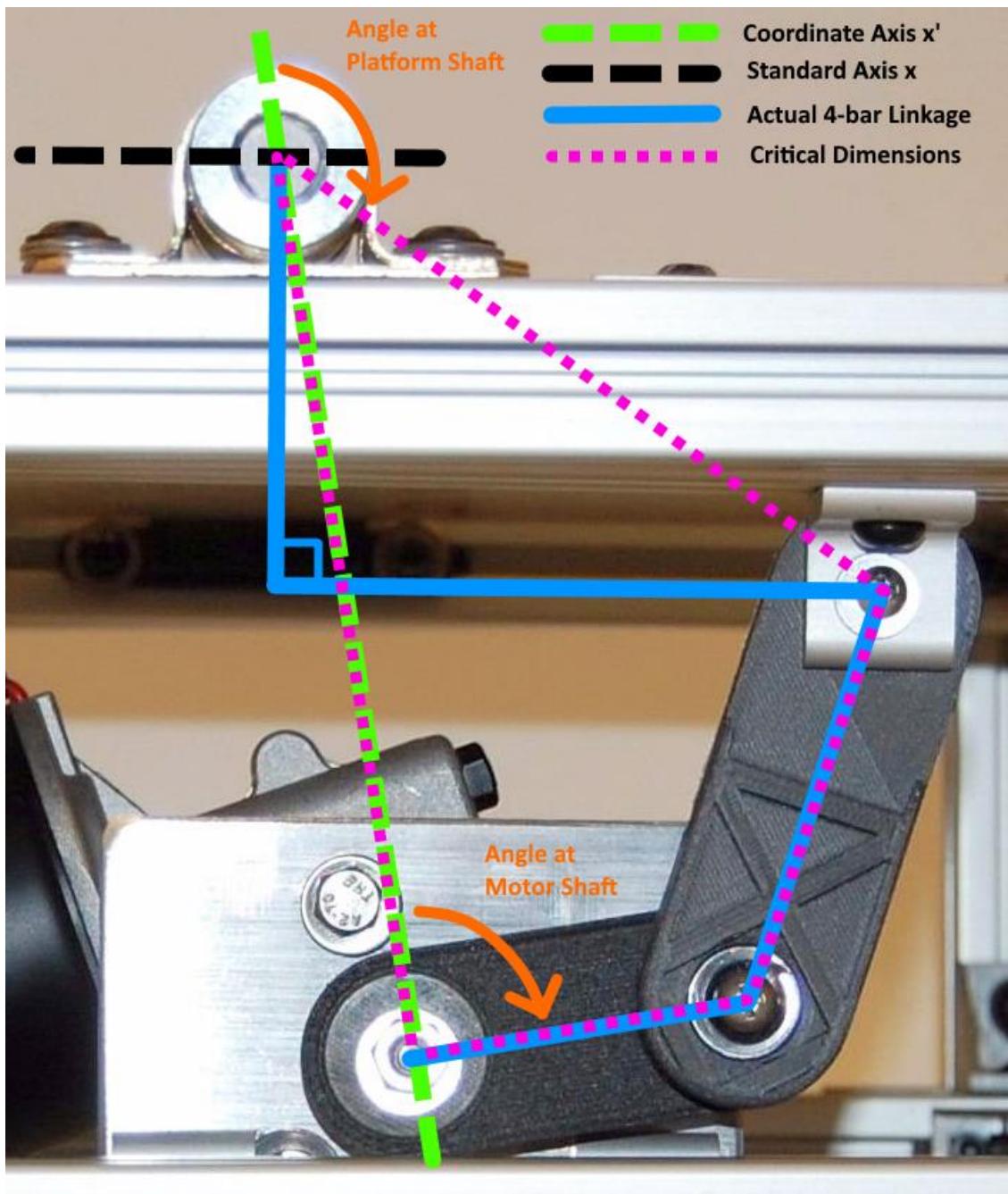


Figure C-2. Picture showing relationship of system linkage relates to four-bar linkage analysis

## 10.0 – Appendix D: System-level Design & Performance Matrices

Requirements Matrix - Bi-pedal Robot Testing Platform (BRUTE)															
Requirement ID	Requirement Description	Performance Metrics								Design Constraints					
		1	2	3	4	5	6	7	8	m	s	kg	m	deg/s	samples/s
1	The platform footprint must be sufficient for the robot	*	*	*	*	*	*	*	*	1	Minimum footprint dimension	Units	m*	°/s	?
2	The platform provides sufficient degrees of freedom with appropriate acceleration, velocity, and displacement values	*	*	*	*	*	*	*	*	2	Minimum mass to be supported	kg	m	deg/s	?
3	The platform enables oscillation in rotational degrees of freedom over the full range of angular velocity	*	*	*	*	*	*	*	*	3	Maximum height	m	m	deg/s	deg/s
4	Dynamics data of system must be recorded and stored	*	*	*	*	*	*	*	*	4	Maximum linear acceleration	m/s^2	m/s	deg/s	deg/s
5	The system allows for user inputs to control the motions	*	*	*	*	*	*	*	*	5	Maximum linear velocity	m/s	m	deg/s	deg/s
6	The system accurately measures the dynamics data	*	*	*	*	*	*	*	*	6	Maximum displacement	m	m	deg/s	deg/s
7	The platform must support the mass of the robot	*	*	*	*	*	*	*	*	7	Maximum degree of rotation	m	m	deg/s	deg/s
8	The platform must be made of mostly off-the-shelf parts	*	*	*	*	*	*	*	*	8	Maximum angular velocity	m/s	m	deg/s	deg/s
Target design requirements															
Lower Acceptable	0.15 m x 0.1 m	10	0.3	0.5	0.5	0.5	0.5	0.5	0.5	10	1000	1	1	10	?
Ideal	0.2m x 0.15m	20	0.4	1	1.5	0.75	10	30	60	100	1200	1	2	11	N/A
Upper Acceptable	0.75 m x 0.75m	40	1	5	4.5	1.5	1.5	1.5	1.5	60	1000	1	2	12	N/A

*Figure D-1 System level Requirements-Measurements Matrix developed during Opportunity Development Phase*

*Table D-1. System-level performance measures were updated to include updates of measurement techniques and measured values*

<b>Performance Metric</b>	<b>Units</b>	<b>Minimum</b>	<b>Ideal</b>	<b>Maximum</b>	<b>Measurement Technique</b>	<b>Measured</b>
Minimum footprint dimension	m <sup>2</sup>	0.15m x 0.1m	0.2m x 0.15m	0.75m x 0.75m	Use tape measure to calculate footprint	0.2597
Minimum mass supported	kg	5	9	15	Test apparatus with proof load (9kg)	9
Maximum Platform Height	cm	7.75	15.25	25.25	Use tape measure to measure height	17.78
Maximum degree of rotation	degrees	5	10	20	Measure with angle level	7.2
Maximum angular velocity	degree/s	10	100	150	Take data at maximum operating frequency and calculate max speed from encoder data	48.35
Minimum sensor resolution	degree/loop	0.04	0.20	1	Run system at lowest frequency of each control band and calculate maximum position change in single control loop	0.65 at 0.1Hz 0.39 at 0.4Hz
Maximum project cost	\$	1000	1500	2000	Maintain detailed budget to measure against budgetary goals	\$1567.32
Response frequency	Hz	0.1	1	5	Compare period of output oscillation to that of theoretical/input oscillation	1.1

## 11.0 Appendix E: Mechanical Subsystem

Subsystem Requirements-Measurements Matrix for Frame

	Target design requirements							< dwl	Performance measures	Units
	Upper Acceptable	Ideal	Lower Acceptable	Imp	✓	9	6	3	6	6
1	0	0			•	9		1	Deflection	mm
2	2	0			•	9		2	Vibration	Hz
3					•	3		3	Platform weight	Kg
4					•	3		4	Time	Minutes
5						3				
6						6			5	Torque
7						9			6	Area
									7	Height
										cm

Figure E-1 Requirements-Measurements Matrix for frame subsystem

Subsystem Requirements-Measurements Matrix for Actuation

	Target design requirements					Imp ->	Upper Acceptable	Lower Acceptable	Performance measures	Units
	1	2	3	4	5		9	8	9	Nm
1	Motors supply sufficient torque to move robot and platform						*	1	Static torque	Nm
2	Motors supply sufficient holding torque to prevent backlash						*	2	Torque	Nm
3	Motors have sufficient rpm to move platform at sufficient speed						*	3	Speed	m/s
4	Motor height does not exceed space available under platform						*	4	Size	m
5	Power supply to motor does not exceed feasible amperage						*	5	Current	A

Figure E-2. Requirements-Measurements Matrix for actuation subsystem

## 12.0 Appendix F: Controls Subsystem

Controls Subsystem Requirements Matrix

	Target design requirements				< due	Performance measures	Units
	Ideal	Lower Acceptable	Upper Acceptable			(Sampling rate)/(Nyquist frequency)	(Hz/Hz)
1	Controls sampling rate prevents aliasing	9	9	10	•	1	
2	Sensor feedback is accurate	9	9	10	•	2	Displacement deviation
3	Control system is accurate	9	9	0.5	•	3	Angular displacement deviation
4	Sensors deliver clear signal	6	6	40	•	5	Signal to noise ratio (signal/noise)
	Imp ->	9	6	20	10	9	

Figure F-1 Requirements-Measurements matrix for controls subsystem

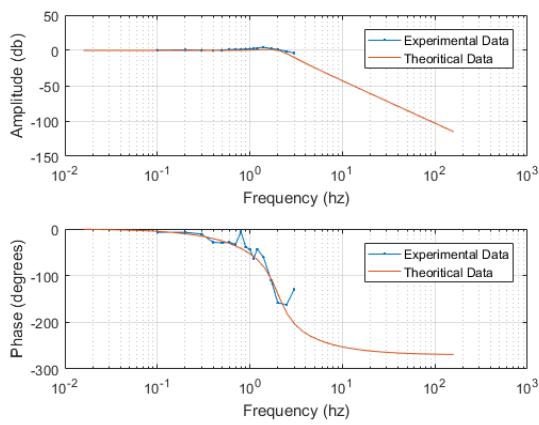


Figure F-2: Bode plot of motor only system

Damping Ratio,  $\zeta = 0.6\%$   
 Natural Frequency,  $\omega_n = 2.5\%$   
 $\omega = 2\pi\omega_n$

$$K_{tot} = 7\%^* \\ G_p = \frac{1727}{s^3 + 18.85s^2 + 246.7s}$$

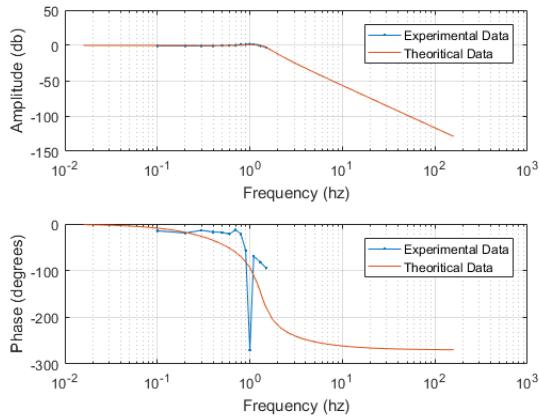


Figure F-3: Bode plot of motor and platform system

Damping Ratio,  $\zeta = 0.5\%$   
 Natural Frequency,  $\omega_n = 1.5\%$   
 $\omega = 2\pi\omega_n$

$$K_{tot} = 4 \\ sys = \frac{355.3}{s^3 + 9.425 s^2 + 88.83 s + 355.3}$$

Fixed Sampling Frequency Bands, Varied Operating Frequency (Encoders are Direct Mounted at Platform Level)					Encoder Hardware Resolution (2048) = 8192					
Operating Frequency (Hz)	2*pi*(Op Freq)	Maximum Angular Velocity (dps)	Average Angular Velocity (dps)	Minimum Sampling Frequency (Hz)	Actual Sampling Frequency (Hz)	Average Position Change per Control Loop (deg)	# Encoder Counts per Control Loop	% Position Error at Average Velocity	Velocity Resolution at Sampling Frequency (dps)	% Velocity Error
0.1	0.6283	4.7124	3.1	1	20	0.1571	3.6	28.0%	0.8789	28.0%
0.2	1.2566	9.4248	6.3	2	20	0.3142	7.1	14.0%	0.8789	14.0%
0.3	1.8850	14.1372	9.4	3	20	0.4712	10.7	9.3%	0.8789	9.3%
0.4	2.5133	18.8496	12.6	4	50	0.2513	5.7	17.5%	2.1973	17.5%
0.5	3.1416	23.5619	15.7	5	50	0.3142	7.1	14.0%	2.1973	14.0%
0.6	3.7699	28.2743	18.8	6	50	0.3770	8.6	11.7%	2.1973	11.7%
0.7	4.3982	32.9867	22.0	7	50	0.4398	10.0	10.0%	2.1973	10.0%
0.8	5.0265	37.6991	25.1	8	50	0.5027	11.4	8.7%	2.1973	8.7%
0.9	5.6549	42.4115	28.3	9	50	0.5655	12.9	7.8%	2.1973	7.8%
1	6.2832	47.1239	31.4	10	50	0.6283	14.3	7.0%	2.1973	7.0%
1.1	6.9115	51.8363	34.6	11	50	0.6912	15.7	6.4%	2.1973	6.4%

Figure F-4. Sample of tool developed in Excel and provided in the catalog to estimate encoder error at varied sampling and operating frequency. Encoder error increases as sample time increases, and increases as operating frequency decreases. The low operating speeds and encoder resolution of 2048 pulses per revolution necessitate very low sample times. The minimum sampling frequency set at 10 times the operating frequency, which is five times the Nyquist frequency.

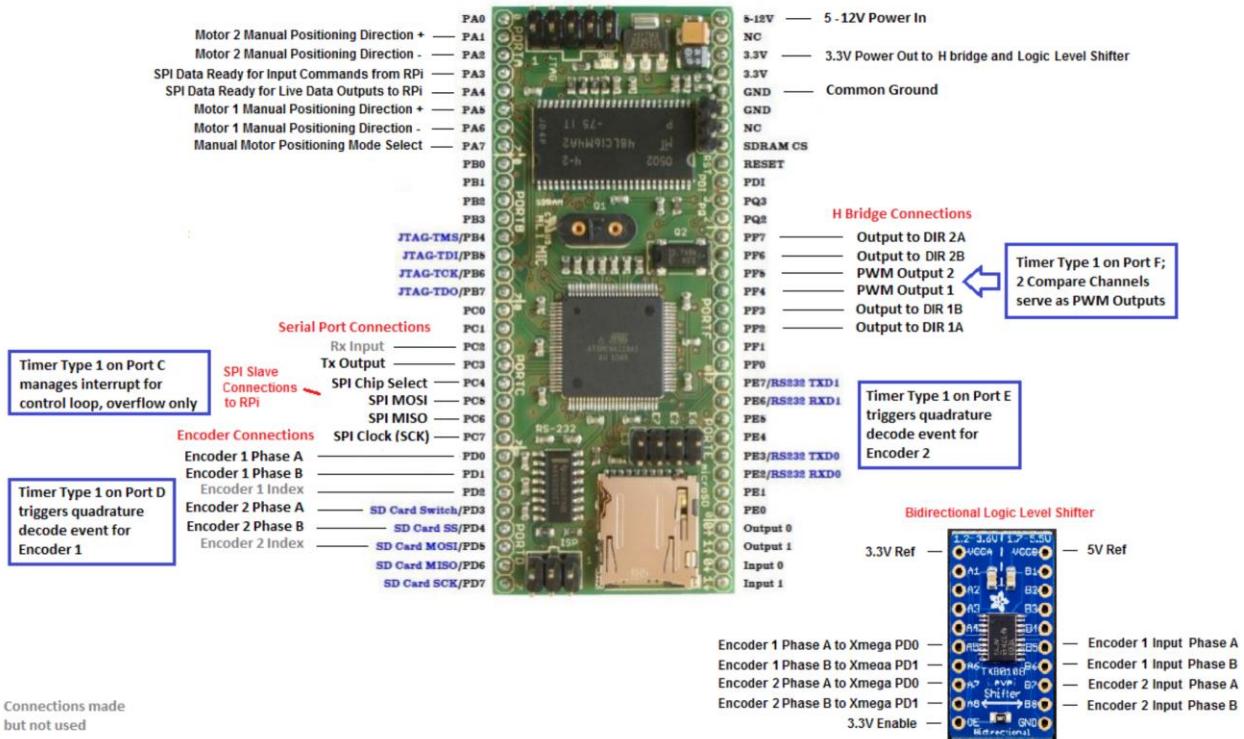


Figure F-5. Figure F-5. Hardware connections and enabled software features on Atmel Xmega microcontroller and logic level converter are shown. Quadrature encoders are used for dynamic control and amplification to the motors is provided by PWM through motor driver carriers.

## 13.0 Appendix G: User Interface

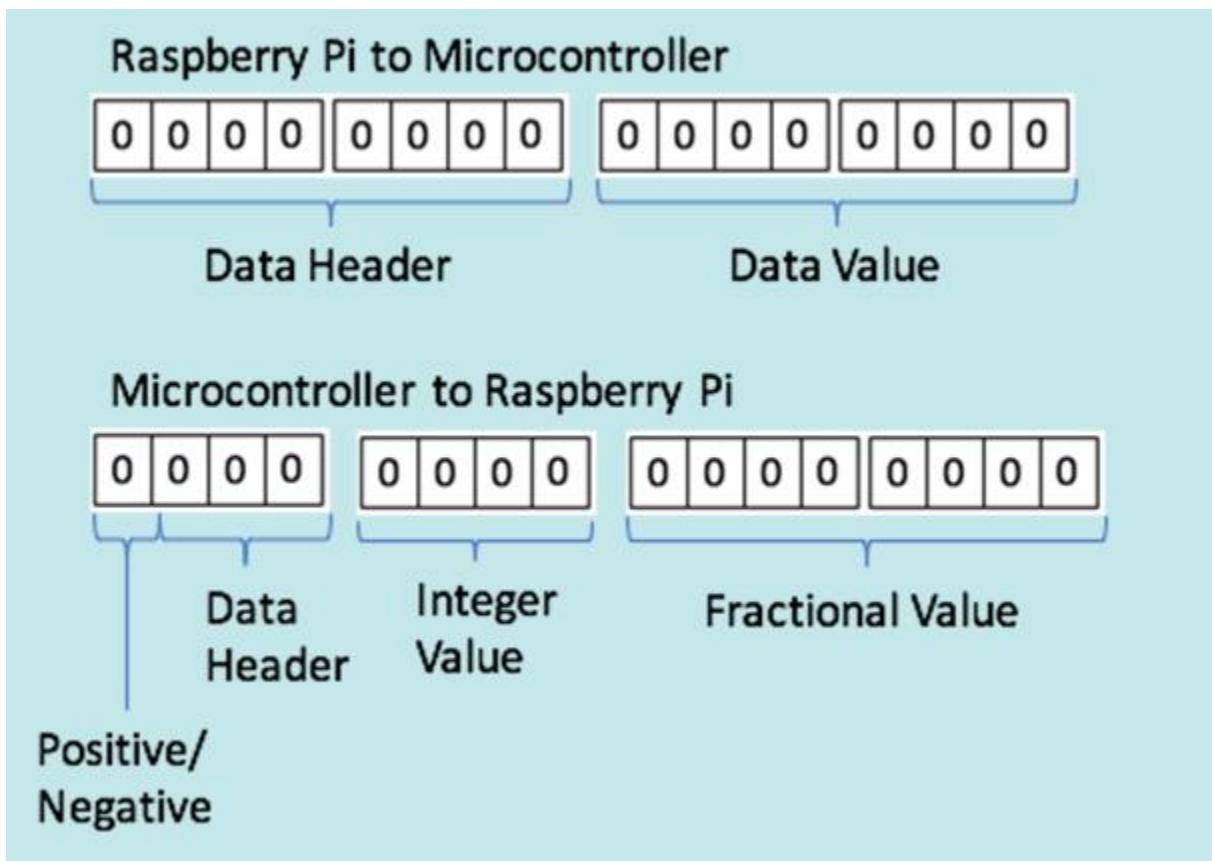


Figure G-1. Data structure for SPI communication between Xmega and Raspberry Pi to transmit coded discrete commands such as input amplitude and frequency as well as floating-point numbers.

6/3/2017      [Brute Launch Pad](#)

[Emergency Stop / Reset](#) [Reset/Stop](#)

### Frequency Input

Amplitude 1 (deg)

Frequency 1 (Hz)

Amplitude 2 (deg)

Frequency 2 (Hz)

Duration(Num Periods)

Send to Xmega:

### Step Input

Amplitude 1 (deg)

Amplitude 2 (deg)

Send to Xmega:

[Download Txt](#)

Legend: Motor 1 Command (Red Solid), Motor 1 Actual (Yellow Dashed), Motor 2 Command (Blue Solid), Motor 2 Actual (Pink Dashed)

*Figure G-2. Snapshot of user interface with live input and output signals*

```

top - 21:29:02 up 3:01, 2 users, load average: 0.27, 0.13, 0.04
Tasks: 92 total, 2 running, 90 sleeping, 0 stopped, 0 zombie
%Cpu(s): 88.6 us, 4.9 sy, 0.0 ni, 1.0 id, 0.0 wa, 0.0 hi, 5.6 si, 0.0 st
KiB Mem: 444540 total, 137716 used, 306824 free, 16696 buffers
KiB Swap: 102396 total, 0 used, 102396 free. 65948 cached Mem

 412 nobody    20   0   2280   1460   1340 S  0.3  0.3  0:00.22 thd
419 root    20   0 150824 35200 19584 R 96.0  7.9  0:41.62 node
10635 root  20   0  5100  2424  2016 R  1.3  0.5  0:04.31 top
  3 root       20   0     0     0     0 S  0.3  0.0  0:03.84 ksoftirqd/0
10273 pi       20   0   12280   4048   3332 S  0.3  0.9  0:01.08 sshd
10837 root     20   0     0     0     0 S  0.3  0.0  0:00.06 kworker/0:2
10838 root     20   0     0     0     0 S  0.3  0.0  0:00.01 kworker/u2:2
  1 root       20   0   5428   3836   2720 S  0.0  0.9  0:05.25 systemd

```

Figure G-3. Raspberry Pi System performance log during platform operation at maximum sampling frequency

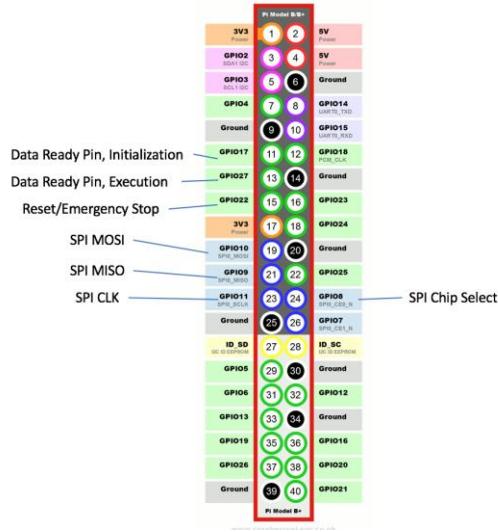


Figure G-4. Raspberry Pi Pin layout with labels for the pins used and their location

GUI Subsystem Requirements Matrix - Bi-pedal Robot Testing Platform (BRUTE)

		Performance measures													
		<- dual		1 Simultaneous Step Input for each DOF		2 Simultaneous Frequency input for each DOF		3 Time delay for performance data to be displayed to user		4 Percent of system processor used		5 percent of system memory used		6 Maximum size of input file	
		Units	Each	Each	Each	ms	%	%	kb						
	<b>Target design requirements</b>														
1	Allows for user to input step input in each DOF	9	•												
2	Allow user to input sinusoidal input in each rotational DOF at a given frequency	9		•											
3	Track command position versus real time position	3				•									
4	Provides near real time feedback to the user of the platform's position	6				•									
5	Doesn't consume all the systems resources	9				•	•								
6	Allows for parametric equation or point cloud input in a text file	6	•	•											
	Imp ->	9	9	3	6	6	6	3	3	25	25	1k			
		Ideal	Lower	Acceptable	1	1	1	25	25	100	100	1k			
		Upper	Acceptable		3	3	500	40	50	100	100	50K			

Figure G-5. Requirements-Measurements Matrix for GUI subsystem

## 14.0 Appendix H: Bill of Materials

Bill of Materials					Total Cost	\$1,434.80
Qty	Unit of Measure	Vendor	Part Number	Description	Unit Price	Cost
30	ea,	80/20	3204	1/4"-20 Standard T-Nut	\$0.79	\$23.70
30	Inches	80/20	3204	10 Series 1/4-20 Standard Slide-in T-Nut	\$0.79	\$23.70
12	ea,	80/20	3204	10 Series 1/4-20 Standard Slide-in T-Nut	\$0.79	\$9.48
60	ea,	80/20	3342	1/4"-20 FBHSCS (screw)	\$0.30	\$18.00
12	ea,	80/20	3342	Flanged Button Head Socket Cap Screw	\$0.30	\$3.60
10	ea,	80/20	3393	Black BHSCS and Slide-In Economy T-Nut	\$0.40	\$4.00
5	ea,	80/20	4265	10 Series 2 Hole - Slotted Inside Corner Bracket	\$3.55	\$17.75
4	ea,	80/20	6423	Double flange standard linear bearing	\$55.95	\$223.80
20	ea,	80/20	14058	Corner Bracket for T-slot extrusions	\$1.80	\$36.00
240	Inches	80/20	1010-S	1" T-slotted aluminum extrusions	\$0.23	\$55.20
1	ea,	Alvidi	AL-XAVRB V2.0	ATXmega128A1U Microcontroller	\$42.50	\$42.50
1	ea,	Amazon	7805 5V TO-220	Voltage Regulator to 5-12V	\$7.68	\$7.68
1	ea,	Amazon	FTDI TTL 232R	3.3V USB to Serial Cable	\$26.99	\$26.99
1	ea,	Amazon	na	12V 2A power adapter and barrel jack	\$8.99	\$8.99
2	ea,	Amazon	PN01007	Maker Motor 12V DC Reversible Electric Gear Motor 50 RPM	\$68.99	\$137.98
2	ea,	Amazon	Ruland PCMR25-10-8-A	10mm to 8mm zero backlash coupler	\$32.31	\$64.62
1	ea,	Amazon	TXS0108E	Gikfun 8 channel logic level converter	\$8.58	\$8.58
2	ea,	Digikey	AMT102-V	Encoder with Index for 2-8mm shaft	\$23.63	\$47.26
2	ea,	Digikey	CUI-3131	Encoder cable	\$12.10	\$24.20
1	ea,	LID	1.2 B+	Raspberry Pi	\$50.00	\$50.00
1	ea,	LID	na	8 gb SD card	\$10.00	\$10.00
1	ea,	McMaster-Carr	1327K53	3/16" diameter 12" carbon steel rotary shaft	\$6.83	\$6.83
1	ea,	McMaster-Carr	1346K17	1/2" 1566 Carbon Steel Shaft	\$8.06	\$8.06
1	ea,	McMaster-Carr	1439K211	Keyed rotary shaft carbon steel 8mm	\$20.27	\$20.27
11	ea,	McMaster-Carr	5906K513	1/2" dia oil-embedded thrust washer	\$1.14	\$12.54
1	ea,	McMaster-Carr	60645K16	Ball Joint Rod End, 55 deg range 1/2" RH Shank	\$6.58	\$6.58
8	ea,	McMaster-Carr	6383K11	Ball bearing for 3/16" shaft	\$3.57	\$28.56
4	ea,	McMaster-Carr	6432K16	1/2" dia shaft collar	\$1.26	\$5.04
10	ea,	McMaster-Carr	7930K13	Bearing Block for a 1/2" shaft	\$13.00	\$130.00
2	ea,	McMaster-Carr	8600N2	3/16" mounted ball bearing	\$15.14	\$30.28
1	ea,	McMaster-Carr	8975K142	1/4" thick Aluminum Plate 12"X24"	\$45.56	\$45.56
1	ea,	McMaster-Carr	90295A414	0.188" nylon washer	\$7.39	\$7.39
4	ea,	McMaster-Carr	92401A574	Clevis pin 3/16" with 9/16 usable length	\$4.71	\$18.84
1	ea,	Newark	45X9866	Atmel Ice Basic Programmer	\$94.68	\$94.68
1	ea,	Online Metals	na	12"x12" aluminum bare sheet T6 0.125" thick	\$16.44	\$16.44
2	ea,	Pololu	G2 24v13	Hi Power Motor Drivers (13A)	\$29.95	\$59.90
3	ea,	Pololu	VNH-5019	Motor Driver Carrier	\$24.95	\$74.85
1	ea,	Sparkfun	LSM9DS1	IMU	\$24.95	\$24.95

### Vendors

Name	URL	Location
80/20	<a href="http://www.8020.net">www.8020.net</a>	Columbia City, IN
Alvidi	<a href="http://www.alvidi.de">www.alvidi.de</a>	Oulu, Finland
Amazon	<a href="http://www.amazon.com">www.amazon.com</a>	Seattle, WA
Digikey	<a href="http://www.digikey.com">www.digikey.com</a>	Thief River Falls, Minnesota
LID	<a href="http://psu-epl.github.io/">http://psu-epl.github.io/</a>	Portland, OR
McMaster-Carr	<a href="http://www.mcmaster.com">www.mcmaster.com</a>	Elmhurst, IL
Newark	<a href="http://www.newark.com">www.newark.com</a>	Chicago, IL
Online Metals	<a href="http://www.onlinemetals.com">www.onlinemetals.com</a>	Seattle, WA
Pololu	<a href="https://www.pololu.com/">https://www.pololu.com/</a>	Las Vegas, NV
Sparkfun	<a href="http://www.sparkfun.com">www.sparkfun.com</a>	Boulder, CO

Notes: Miscellaneous parts such as wiring, small screws, covers are not included