



Prepared By: Monica Berg, Tito Fernandez, Ben Lim, Roshanne  
Malekmadani, Chris Miller, Blake Strebel, Huan Weng

Northwestern University  
June 8, 2018

## Table of Contents

<b>Project Definition</b>	<b>5</b>
<b>Sponsor Needs</b>	<b>5</b>
<b>Team and Roles</b>	<b>6</b>
<b>System Diagram</b>	<b>6</b>
<b>Engineering Analysis</b>	<b>7</b>
Shaft Analysis	7
<i>Shaft 1</i>	7
<i>Shaft 2</i>	7
Finite Element Analysis: Link 1	8
Dynamics and Equations of Motion	9
<i>Kinematics</i>	9
<i>Moments of Inertia</i>	9
<i>Energy Equations</i>	10
<i>Equations of Motion</i>	10
Electrical Analysis	11
<i>Power Specifications:</i>	11
<i>Microcontroller Selection - TM4C1294NCPDT</i>	12
<i>Encoder Selection – Renishaw Orbis (BR10SPC14B12CH00) Absolute Encoder</i>	12
<i>Electrically Driven Motor Requirements</i>	13
<i>Motor Controllers – TI DRV8323R Three-Phase Smart Gate Driver</i>	13
<i>UART Communications – FTDI FT230X</i>	13
<i>Other Peripherals</i>	14
<b>Evolution of Design from Winter Quarter Midterm to Present</b>	<b>15</b>
Mechanical Design Evolution	15
<i>Winter Midterm</i>	15
<i>Winter Final</i>	15
<i>Spring Midterm</i>	16
<i>Present</i>	17
Electrical Design Evolution	18
<b>Final Design Description</b>	<b>20</b>
Mechanical Design	20
<i>Arm Links</i>	20
<i>Adjustable Bearings and Motors</i>	21
<i>Onboard Encoders</i>	21
<i>Adjustable Aluminum Housing</i>	22
<i>Belt Guard</i>	22
Electrical Design	23
<i>Changes from First Board to Second</i>	23
<i>On the PCB and Documented:</i>	23
<i>Only in Documentation:</i>	24
<i>Electronics Fabrication and Assembly</i>	24
<i>Test Bench Discussion</i>	24
Software	25
<i>Overview</i>	25
<i>Current and Position Control</i>	25

Menu Interface .....	25
Documentation .....	26
Controls.....	26
Overview .....	26
Motor Commutation.....	27
Current Controller.....	29
Motor Decogging .....	30
<b>Testing Plans .....</b>	<b>30</b>
Mechanical .....	30
Friction Testing: Ball Bearings .....	30
Friction Testing: Timing Belts vs. Posi-Drive.....	30
Motor Characterization: .....	30
Electrical: Benchtop.....	30
Motors + SPI Device.....	30
Encoders .....	31
I2C Device.....	32
Analog Device (Temperature Probe) .....	32
Electrical: PCB .....	32
Input power: .....	32
Microcontroller:.....	33
UART Communications: .....	33
Encoders: .....	34
Motors and Halls: .....	34
<b>Performance Results.....</b>	<b>36</b>
Friction Reduction .....	36
Motor Characterization .....	36
Current Control.....	37
Position Control.....	38
<b>Next Steps.....</b>	<b>38</b>
Mechanical .....	38
Electrical .....	38
Electrical System Incidents.....	39
Software.....	40
<b>List of Other Documentation .....</b>	<b>41</b>
<b>Appendix A: Shaft Analysis.....</b>	<b>42</b>
<b>Appendix B: Link Properties .....</b>	<b>44</b>

Figure 1: Shaft 1 Analysis.....	7
Figure 2: Shaft 2 Analysis.....	7
Figure 3: Deformation in Link 1 .....	8
Figure 4: Stress in Link 1 .....	8
Figure 5: Variables used in Double Pendulum equations. Image courtesy of MIT OCW .....	9
Figure 6: Initial Solidworks conceptual model.....	15
Figure 7: Solidworks rendition of winter final prototype.....	16
Figure 8: Preliminary Housing design, illustrating updated link 2.....	17
Figure 9: Final Production Prototype.....	18
Figure 10: Full Prototype of the R2R Robot.....	20
Figure 11: Close-up of the arm linkages .....	21
Figure 12: Bearing pillow blocks can slide along the 80/20 channel .....	21
Figure 13: The encoders reside below the pillow blocks.....	22
Figure 14: Exterior aluminum frame of the device .....	22
Figure 15: Belt Guard shown in place .....	23
Figure 16 Motor Control Flowchart.....	26
Figure 17: BLDC Motor Current.....	27
Figure 18: Motor Driving Schematic .....	28
Figure 19: Damping of Link due to Friction .....	36
Figure 20: Motor Characterization Results.....	36
Figure 21: Current Control Performance at Stall .....	37
Figure 22: Current Control Performance While Moving .....	37
Figure 23: Position Tracking Performance.....	38
Table 1: Motor Control Frequencies .....	27
Table 2: Commutation Table .....	27
Table 3: Gate Driver Inputs .....	28
Table 4: Gate Driver Outputs.....	29
Table 5: Current Sensing.....	29
Table 6: Motor Testing Sequence .....	35

## Project Definition

The objective of this project is to develop a direct-drivable, low-cost, educational 2R robot arm capable of continuous rotation. Additionally, the robot must have the following characteristics:

- Easily programmable and transparent at all levels (microcontroller to high-level).
- Intuitive user interface for college students.
- Table-top size (linear height ~0.5m); portable
- Easy to configure horizontally or vertically and adjustable for multiple configurations of the arm (2 motors, either motor removed, etc)
- Easily serviceable electromechanical system
- Easily programmable to different control modes (position, force, hybrid, etc)

## Sponsor Needs

Our sponsor, Professor Kevin Lynch, provided the team with the following technical requirements for the robot:

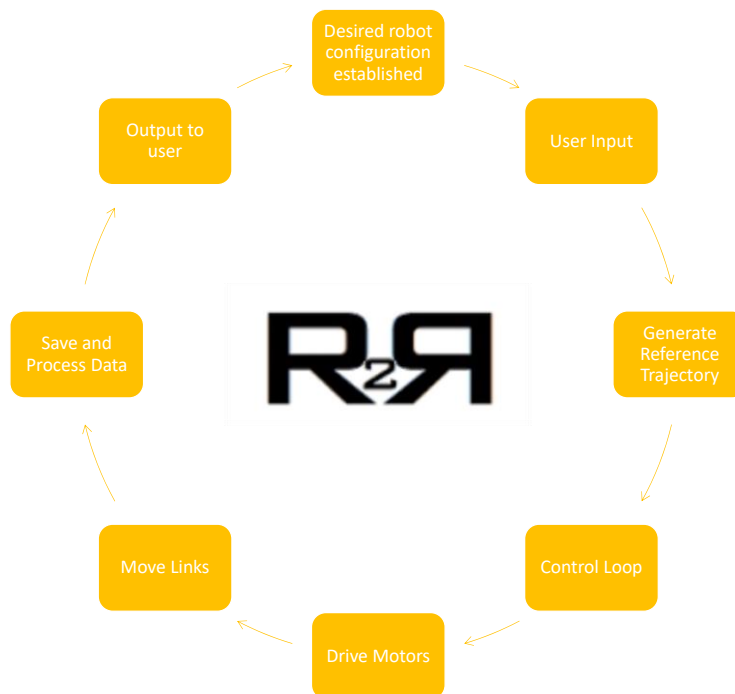
- Two BLDC motors at the base, with one connected to the second joint by a timing belt. Each link has load-mounting points so that users can change the center of mass and inertia of the link.
- Either motor can be removed to make an under actuated robot, e.g., an inverted pendulum (2nd motor removed) or a "pendubot" (1st motor removed).
- Each motor can have no gearing ("direct drive") or gearing (~10:1).
- Each motor has a high-resolution encoder. May use a homing procedure for the robot if necessary.
- A small control box that houses the amplifiers, power supply, microcontroller, etc., with a USB connection to a laptop, 120V to wall, and cables to motors and encoders.
- Current (torque) sensing and control.
- Easy table mount to allow operation in vertical or horizontal orientation.
- Sufficient torque for the motors to continuously hold unloaded outstretched arm against at least 2x gravity.
- Removable mechanical stops for joint limits or continuous rotation.
- Easy control law and filter development and downloading to the microcontroller.
- Interface with MATLAB or Simulink on laptop, with easy visualization of controlled trajectories.
- Simple move functions from the MATLAB interface (e.g., step motions, polynomial trajectories, etc.) that allow easy testing of the control laws.
- Robust design capable of withstanding years of repeated use in a classroom setting (by freshmen)

## Team and Roles

Team R2R's project manager is Roshanne Malekmadani, who delegates tasks, runs meetings, and ensures deadlines are met. The team has been functionally divided into three sub-teams based on individual's skills, experience, and interests. The mechanical team consists of Monica Berg, Tito Fernandez, Roshanne Malekmadani, and Blake Strebel. Tito has been the primary manufacturing lead. The software team includes Ben Lim and Huan Weng focusing on communication protocols and the Graphical User Interface (GUI) and Blake Strebel focusing on controls. Electrical has been primarily led by Chris Miller, with other team members supporting as needed.

## System Diagram

The overall system consists of the 2 revolute joint robot arm and housing, motor driver printed circuit board, and MATLAB script to implement proportional-integral-derivative (PID) control. The user also has the option of programming custom control laws directly onto the TI microcontroller. Below is the overall system diagram



One of the notable features of the system, not shown in the diagram above, is its adjustability and versatility. Several mechanical variations of the robot exist, including mounting orientation, level of actuation, and motor selection. Further, a force sensor can be attached using the mounting holes at the end of link 2 in conjunction with the software and mechanical stops. In terms of adjustability in the software, custom control laws can be implemented by directly programming the microcontroller.

## Engineering Analysis

### Shaft Analysis

The following analysis was performed in order to determine the minimum dimensions of the shafts in order to ensure that they would not fail. **Appendix A: Shaft Analysis** details the lifecycle analysis calculations performed.

#### Shaft 1

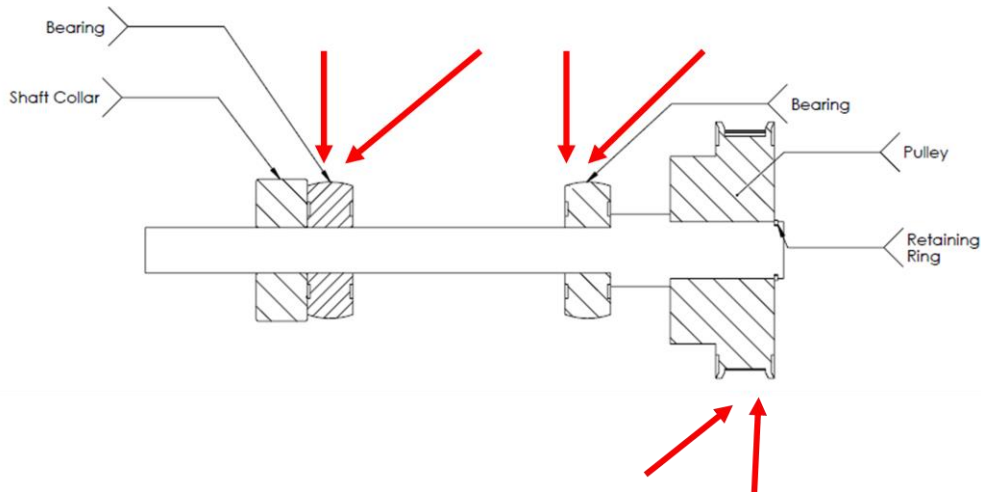


Figure 1: Shaft 1 Analysis

#### Shaft 2

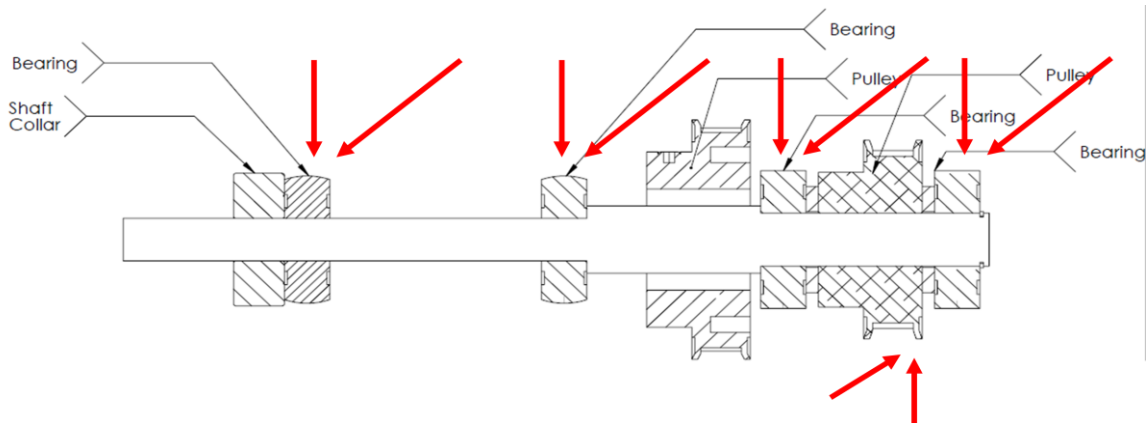


Figure 2: Shaft 2 Analysis

### Finite Element Analysis: Link 1

In order to determine if Link 1 would be strong enough to withstand the forces from the belt tension and torque from the shafts, Finite Element Analysis (FEA) was performed. The results, shown below, indicate that the link is indeed strong enough to be made out of aluminum and ABS, with a maximum deflection of 0.01" and a stress factor of safety of ~50.

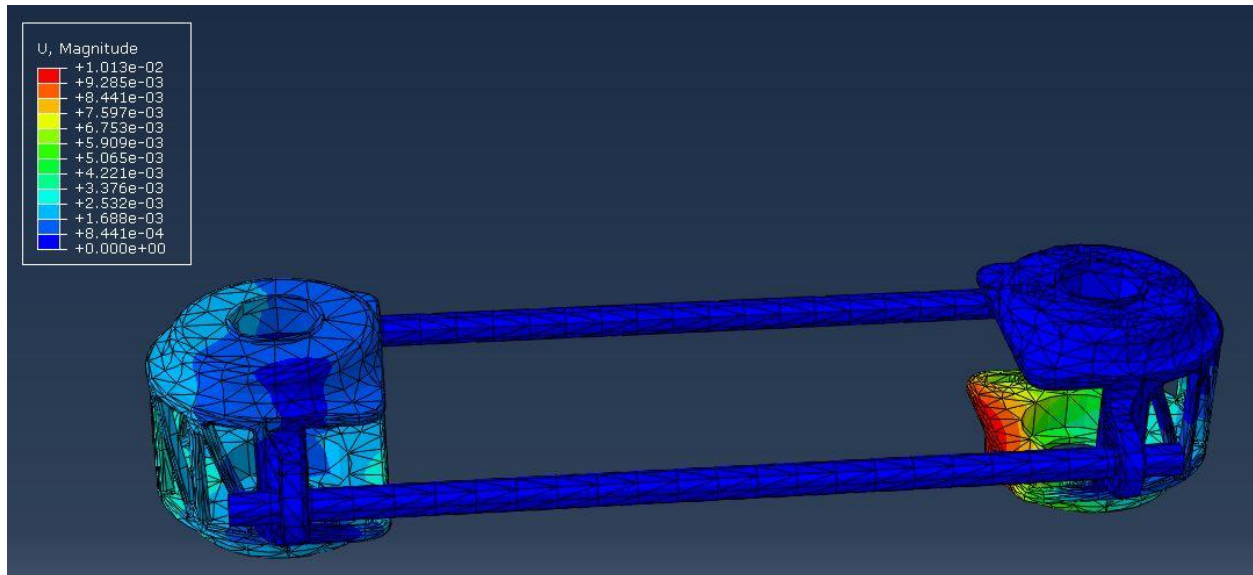


Figure 3: Deformation in Link 1

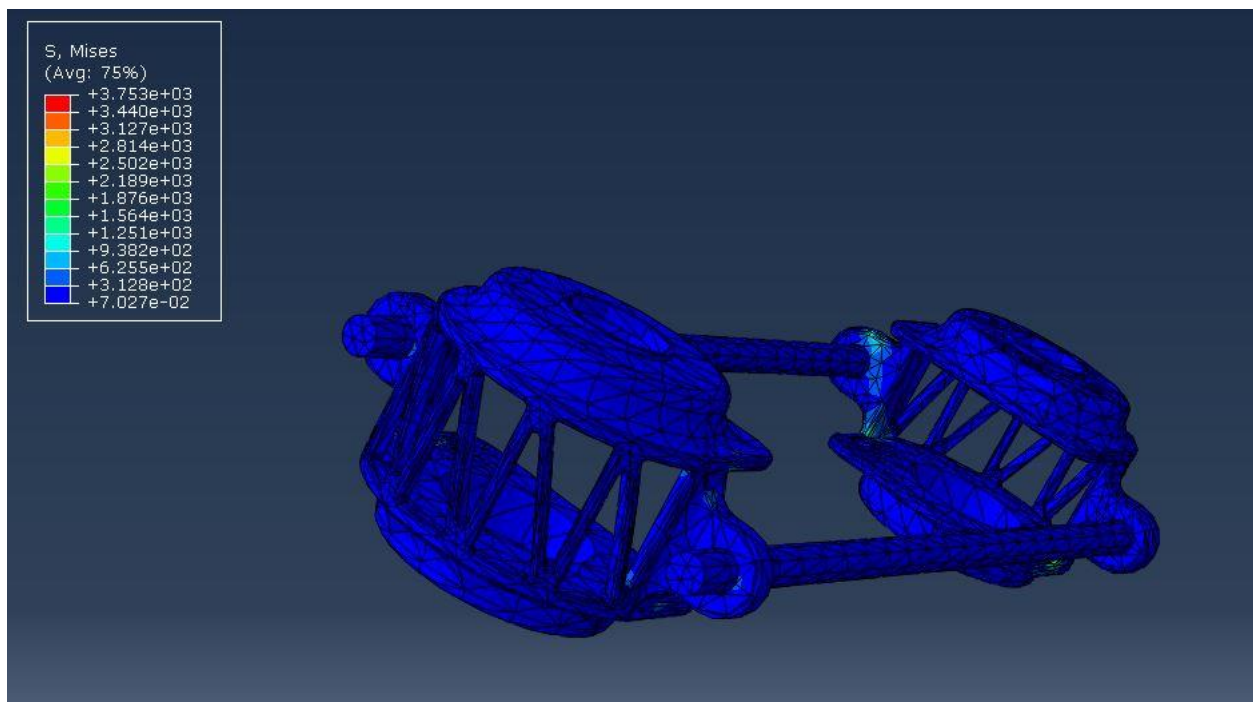
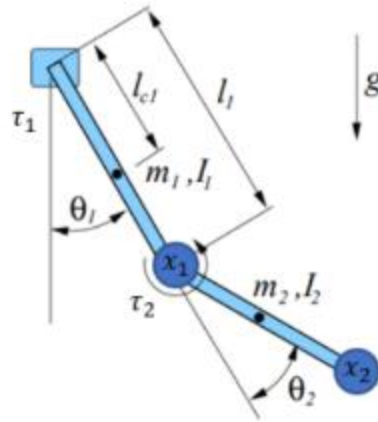


Figure 4: Stress in Link 1



## Dynamics and Equations of Motion

In order to characterize the dynamic behavior of the R2R device, we note that it is essentially a double pendulum system, with optional external torques at the shoulder and elbow joints. Figure 5 explains the variables used in this section. Note that we are simplifying the system dramatically, lumping elements together in ways that will be further explained later.



R2R Physical Parameters	
Link 1 Length	6.00 in.
Link 1 CoM	4.98 in.
Link 2 Length	6.00 in.
Link 2 CoM	3.07 in.
Mass Link 1	125.7 grams
Mass Link 2	47.56 grams

Figure 5: Variables used in Double Pendulum equations. Image courtesy of MIT OCW<sup>1</sup>

### Kinematics

The kinematic equations provide the (x,y) coordinates of the elbow and end effector. Since the system is essentially a double pendulum, these positions rely only on the lengths of the links and the angles.

$$x_1 = \begin{bmatrix} l_1 \sin \theta_1 \\ -l_1 \cos \theta_1 \end{bmatrix} \quad x_2 = \begin{bmatrix} l_2 \sin (\theta_1 + \theta_2) \\ -l_2 \cos (\theta_1 + \theta_2) \end{bmatrix}$$

### Moments of Inertia

The arm of the R2R device is a complex mechanism consisting of a variety of parts. In order to simplify the dynamic calculations, the following assumptions are made:

- 1) The threaded rods, belt, and belt cap of Link 1 are collectively treated as a rigid rod
- 2) The end cap, bearings, pulley, shaft, and nuts in the elbow are collectively treated as a rigid disk centered at the end of the rod in (1).
- 3) Link 2 is similarly treated as a disk on the end of a rod.
- 4) All units are converted to MKS for consistency.

The included MATLAB file (R2R\_dynamics.m) makes use of the standard moment of inertia equations to determine the approximate moment of inertia for each link, by using the above approximations and making use of the parallel axis theorem to account for the fact that the “disks” are rotating about a parallel axis. Moment of inertia for link 1

<sup>1</sup> [https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-832-underactuated-robotics-spring-2009/readings/MIT6\\_832s09\\_read\\_ch03.pdf](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-832-underactuated-robotics-spring-2009/readings/MIT6_832s09_read_ch03.pdf). Most of the mathematical equations and derivations in this section are taken from this text.

was calculated about shaft 2, and that of link 2 was calculated about shaft 3. Using the physical parameters found in Appendix B: Link Properties, the moments of inertia for our system were found to be:

$$I_1 = \begin{bmatrix} 0.0019 & 0 & 0 \\ 0 & 0.0018 & 0 \\ 0 & 0 & 0.0019 \end{bmatrix} \quad I_2 = \begin{bmatrix} 1.0e-03 & 0 & 0 \\ 0.6011 & 0 & 0 \\ 0 & 0.5551 & 0 \\ 0 & 0 & 0.6035 \end{bmatrix}$$

## Energy Equations

Following the classical equations for rotational kinetic energy, we obtain the following for each link:

$$K_1 = \frac{1}{2} I_1 \dot{\theta}_1^2$$

$$K_2 = \frac{1}{2} (m_2 l_1^2 + I_2 + 2m_2 l_1 l_{c2} \cos(\theta_2)) \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + (I_2 + m_2 l_1 l_{c2} \cos(\theta_2)) \dot{\theta}_1 \dot{\theta}_2$$

We then determine the total potential energy of the system:

$$U = -m_1 g l_{c1} \cos(\theta_1) - m_2 g (l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2))$$

The Lagrangian for the system is determined by subtracting the potential energy of the system from the total kinetic energy:

$$L = (K_1 + K_2) - U$$

## Equations of Motion

Once the Lagrangian is obtained, it can be used to generate the equations of motion for this system:

$$(I_1 + I_2 + m_2 l_1^2 + 2m_2 l_1 l_{c2} \cos(\theta_2)) \ddot{\theta}_1 + (I_2 + m_2 l_1 l_{c2} \cos(\theta_2)) \ddot{\theta}_2 - 2m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_2^2 - (m_1 l_{c1} + m_2 l_1) g \sin(\theta_1) + m_2 g l_2 \sin(\theta_1 + \theta_2) = 0$$

$$(I_2 + m_2 l_1 l_{c2} \cos(\theta_2)) \ddot{\theta}_1 + I_2 \ddot{\theta}_2 + m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_1^2 + m_2 g l_2 \sin(\theta_1 + \theta_2) = \tau_2$$

It is interesting to note that the above formulation is for when the robot is in AcroBot mode (forcing on the second joint). However, the equations hold true for the standard and PenduBot modes as well, if one substitutes the appropriate forcing:

$$\text{Default} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad \text{AcroBot} \begin{bmatrix} 0 \\ \tau_2 \end{bmatrix} \quad \text{PenduBot} \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}$$

MATLAB coding is available (R2R\_dynamics.m) to see how the moments of inertia and Lagrangian would change if the physical parameters were modified.

## Electrical Analysis

The engineering specifications for the electrical system were dictated by the needs of both the software and mechanical teams. As a result, the specifications as designed in the final fabrication documentation shall be provided with requirements-driven justification. Any tangential or independent design discussions will be presented as well. As a note: this section reflects the PCB implementations pre-damage and for future fabrications. While the breadboard implementation retains full functionality, its design specifications, either those lacking or secondary to the primary design, are not presented here.

## Power Specifications:

1. Wide input voltage: 22VDC to 56VDC
  - a. Selected to allow for a wide range of motors to be used
2. Fused 25ADC peak current draw
  - a. Selected to allow for up to 11A per motor channel (See 4.a) and 2A for peripheral electronics
  - b. Peripheral electronics allocation was generous to allow for future development should it be deemed necessary
3. Reverse polarity protection
  - a. Common P-MOS/Zener circuit selected for simplicity
  - b. Protects against improper student device handling
  - c. "Tried and true" circuit implemented on thousands of commonly found devices worldwide
4. Over voltage protection
  - a. Zener diode circuit to breakdown at ( $V_z = 56\text{VDC}$ ) 56VDC to compensate for imperfect power supplies
  - b. Protects against improper student handling of device
  - c. Microcontroller monitors primary input rail to allow for motion control in the event of the motor's regenerative effects increasing bus voltage
5. Multi-layer ESD protection
  - a. Fast acting TVS diodes on output of 3.3V, 5V, and primary power rails to short all ESD and transient voltage spikes to ground
  - b. Protects against ESD-related damage
6. 750W 24V (+/- 2V) Power Supply:
  - a. To allow for simultaneous peak current draw by both motors (See 4.a)
  - b. To compensate for non-ideal effects of devices causing voltage drops (24V scaled to 26.4V)
7. TI Simple Switcher 3.3V and 5.0V switching regulators
  - a. Greater than 80% efficiency; far greater efficiency than linear drop-out regulators
  - b. Low thermal footprint
  - c. Ease-of-use, well documented
  - d. Wide voltage input range (up to 60VDC)

### Microcontroller Selection - TM4C1294NCPDT

1. Device requested by software with the following specifications highlighted
2. 120MHz operation
  - a. Selected to allow for operating 1kHz control loop while interfacing with multiple sensors and performing data analysis.
3. Four SPI Buses
  - a. One SPI bus per encoder to prevent any single encoder, should one fail, from preventing the other from failing; safety feature since encoders do not have a fault pin
  - b. One SPI bus to interface with both motor controllers; permissible as both motor controllers have a separate fault pin in the event of a failure event.
  - c. One SPI bus for RAM, should RAM be needed for any future complex programs
4. ADC inputs
  - a. Eight ADC inputs available for use, two broken out for miscellaneous interface
  - b. Other ADC lines used for current sensing, temperature sensing, and optional force sensing
5. Plentiful GPIO pins available
  - a. Allowed for the interface of all devices, both those necessary for operation and those implemented for safety or future expansion
  - b. Of the 90 available, less than 20 remain unused
6. Widely documented and well understood by peers
  - a. Significant documentation available from Texas Instruments
  - b. Organizational-wide use of device allowed for cross-team support

### Encoder Selection – Renishaw Orbis (BR10SPC14B12CH00) Absolute Encoder

1. Brand
  - a. Selected for its reliability, value, legacy within robotics, and corporate support via technical documents and warranty
2. 14-bit Resolution
  - a. Selected as 14 bit yields 16384 steps; when selected, 'back of the envelope' calculations yielded the team required at least 12 bits, therefore, 14 bits provided room for error
3. SPI Interface
  - a. Selected to interface with known TI libraries and large support for encoder SPI interface from Renishaw
  - b. Low-noise environment allowed for this sort of interface to be selected
  - c. SPI interface speed fast enough
4. Magnetic, non-contact
  - a. Magnetic encoders are less prone to improper readings due to age, dirt, and other environmental conditions impacting optical encoders
  - b. Non-contact encoder to prevent friction-related wear-and-tear on the device

5. Absolute
  - a. Selected to prevent need for homing features and to allow for students using the device to know the position of the arm links at all times with precision.

### **Electrically Driven Motor Requirements**

1. See mechanical for full motor selection discussion and motor specifications
2. Motor voltages were limited from 24V to 48V for safety and cost purposes
  - a. US OSHA defines “high voltage” as voltages exceeding 50V and a 50V supply is necessary to drive 48V motors
3. Motor current limit was set at 10A per motor
  - a. Selected to maintain PCB manufacturability and to prevent extreme PCB costs due to heavy weight copper

### **Motor Controllers – TI DRV8323R Three-Phase Smart Gate Driver**

1. Large selection of MOSFETS available for use
  - a. Motor Driver capable of sourcing 1A and sinking 2A to drive MOSFETS
2. Wide voltage range: 6 to 60VDC
  - a. Wider operation than the electronics’ specifications of 22 to 48VDC
3. Three integrated current sense amplifiers for monitoring to allow for coil winding current monitoring
  - a. Adjustable gain and bidirectional capabilities
4. SPI-based device
  - a. Selected over GPIO control to allow access to a richer set of control features
5. Multiple commutation methods available, two of three methods accessible to user with given design to allow students to learn about multiple control modes and for the implementation of both simple and complex controllers
  - a. PWM 1x – trapezoidal commutation; motor controller performs commutation onboard and microcontroller provides direction and speed via a GPIO and PWM signal, respectively
  - b. PWM 3x – sinusoidal commutation; microcontroller performs commutation and provides three PWM signals and a direction signal to command motor
6. Widely documented and well understood by peers
  - a. Significant documentation available from Texas Instruments
  - b. Organizational-wide use of device allowed for cross-team support

### **UART Communications – FTDI FT230X**

1. FTDI UART-to-USB branded devices were selected due to significant, open-sourced documentation and industry-wide legacy
2. Extra header broken out to bypass FTDI chip in the event of device failure or the need for a different, non-USB device to communicate with the electronics; allows for future expansion

## Other Peripherals

**Note:** Justification for some electronics is not provided, as devices are included for “proof of concept” as they were not included in system requirements. This is indicated clearly in the discussion for each of such devices.

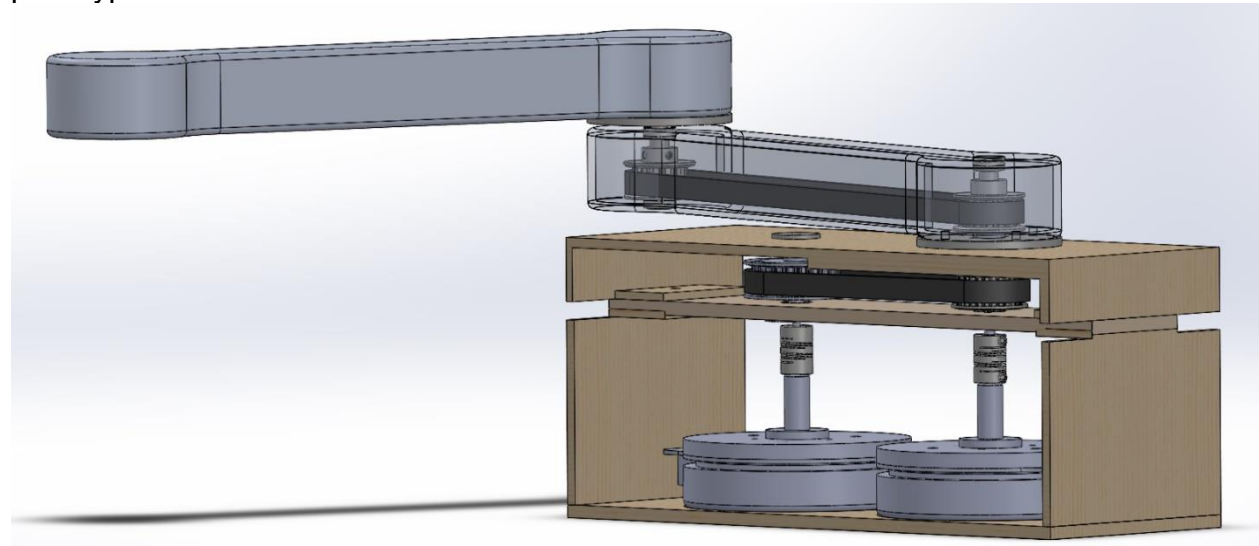
1. Temperature – Analog K-Type thermocouples to monitor motor temperature
  - a. Selected for ease of implementation and wide variety of open-source, well-documented circuit designs
  - b. Selected for wide temperature range and rapid response time
2. ESTOP
  - a. Both motor drivers may be disabled via a software stop command or an external ESTOP button.
  - b. Implemented with via an AND gate that joins the software and hardware commands; selected as discrete gates are more robust and have lower latency than other interfaces
  - c. LED active when system active, LED off when system disengaged, following known practice
3. JTAG Programming Interface
  - a. Selected over SWO for legacy and class-wide familiarity
  - b. JTAG interface broken out to side of electronics box to allow students to easily reprogram the microcontroller
4. Force Sensor
  - a. If needed, analog input for TE Connectivity FC22, with header installed, available for future expansion
  - b. See schematics and software documentation for further details. Device not included, external wiring not installed.
5. IMU
  - a. If needed, I2C-based 9DOF (triple axis acceleration, rotation, and magnetic sensing) SparkFun Electronics MPU6050 with header installed available for future expansion
  - b. See schematics and software documentation for further details. Device not included, external wiring not installed.
6. RAM
  - a. If needed, SPI-based interface for 4MB Fujitsu MB85AS5MY available for future expansion; see schematics and software documentation for further details.
  - b. RAM IC not populated on PCB, support hardware populated

## Evolution of Design from Winter Quarter Midterm to Present

### Mechanical Design Evolution

#### Winter Midterm

At the beginning of Winter Quarter, the team was tasked with developing an educational 2R robotic arm for use as a demonstration tool in controls courses. Unlike typical 2R arms, this device had to allow for continuous rotation of both links, and both motors had to reside in the base, so as to allow the device to be under actuated when desired. After several brainstorming sessions, the team developed the following theoretical prototype for the winter midterm.



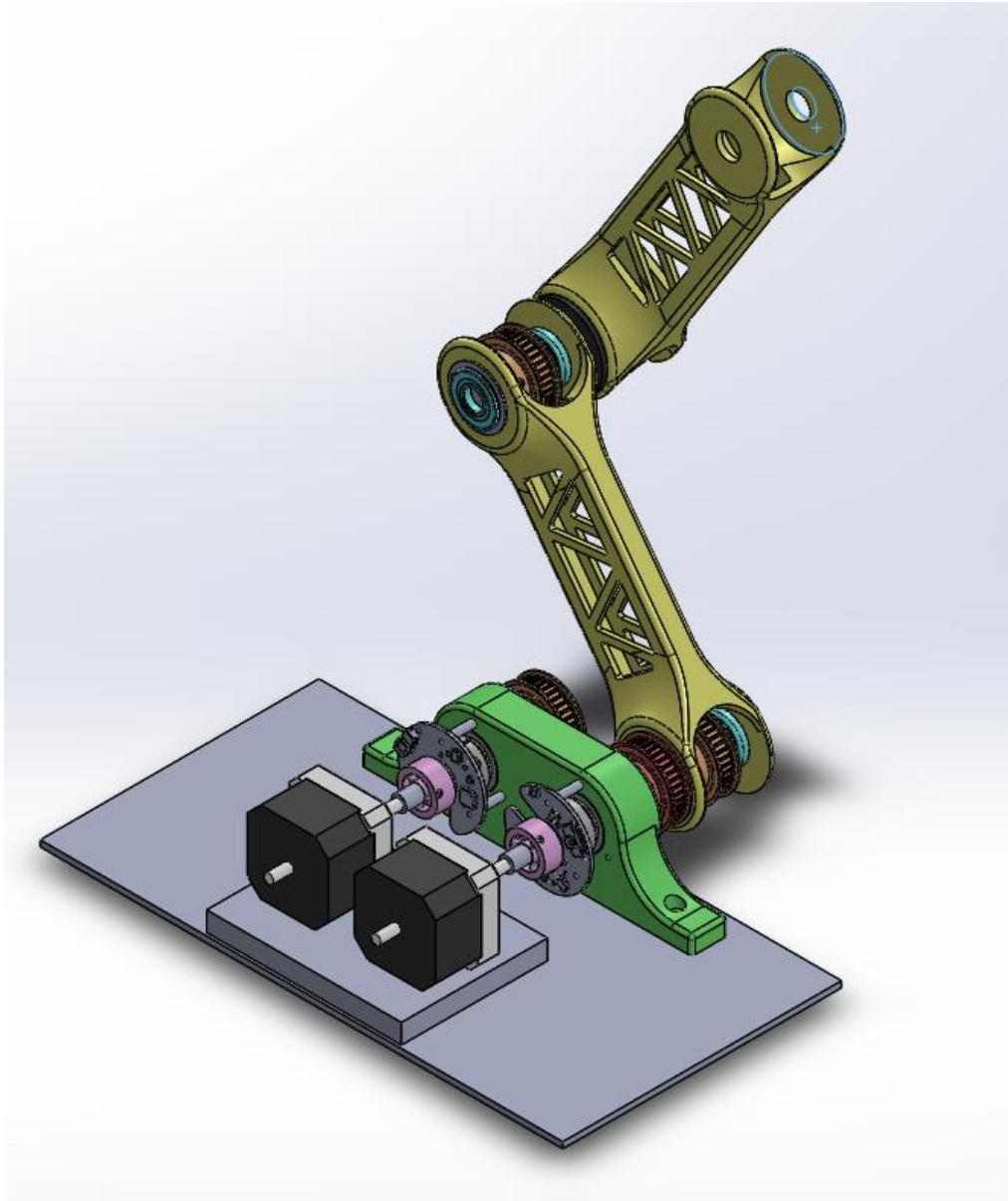
**Figure 6: Initial Solidworks conceptual model**

In this iteration, the leftmost motor drives the first link, and the other motor drives the timing belt in the base link, controlling the second element. The resulting mechanical motion satisfies both the continuous rotation and the under actuation requirements. At this point, arm geometry was not optimized, and housing was purely theoretical.

#### Winter Final

While the previous iteration embodied the core functionality of the robot, strictly speaking it was not mechanically feasible, as it did not satisfy the torque requirements desired by our client. Therefore, the team developed an arm geometry that would be light enough to satisfy the torque constraints. In addition, the physical positioning of parts was determined, and a physical prototype produced. A Solidworks model of the resulting design is shown in Figure 7. Note at this point that the design called for panel bearings to keep the shaft assemblies encased in the arms, and the system is resting in a temporary pillow block since housing had still not been developed.



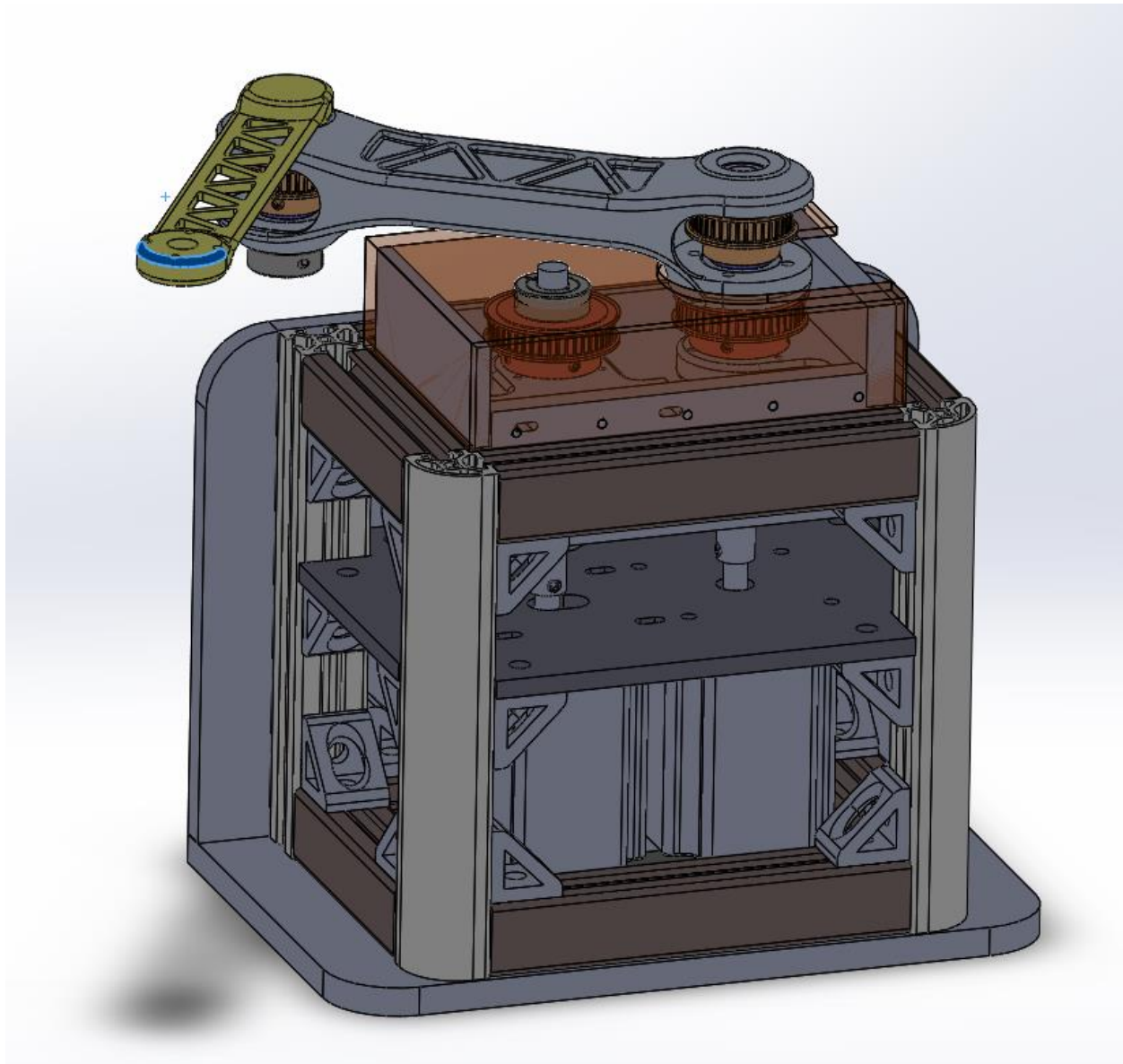


**Figure 7: Solidworks rendition of winter final prototype**

### **Spring Midterm**

By the middle of spring quarter, unnecessary bulk had been removed from the second arm link, and a conceptual design for housing using 80/20 and aluminum had been developed. In addition, a belt guard was designed to protect the lower belt from accidental impacts. Note that there are large aluminum plates on the bottom and backsides of the robot for clamping, and the motors are held on a large continuous aluminum plate, with slots on the left-hand side to allow for tensioning of the lower belt, shown in Figure 8.



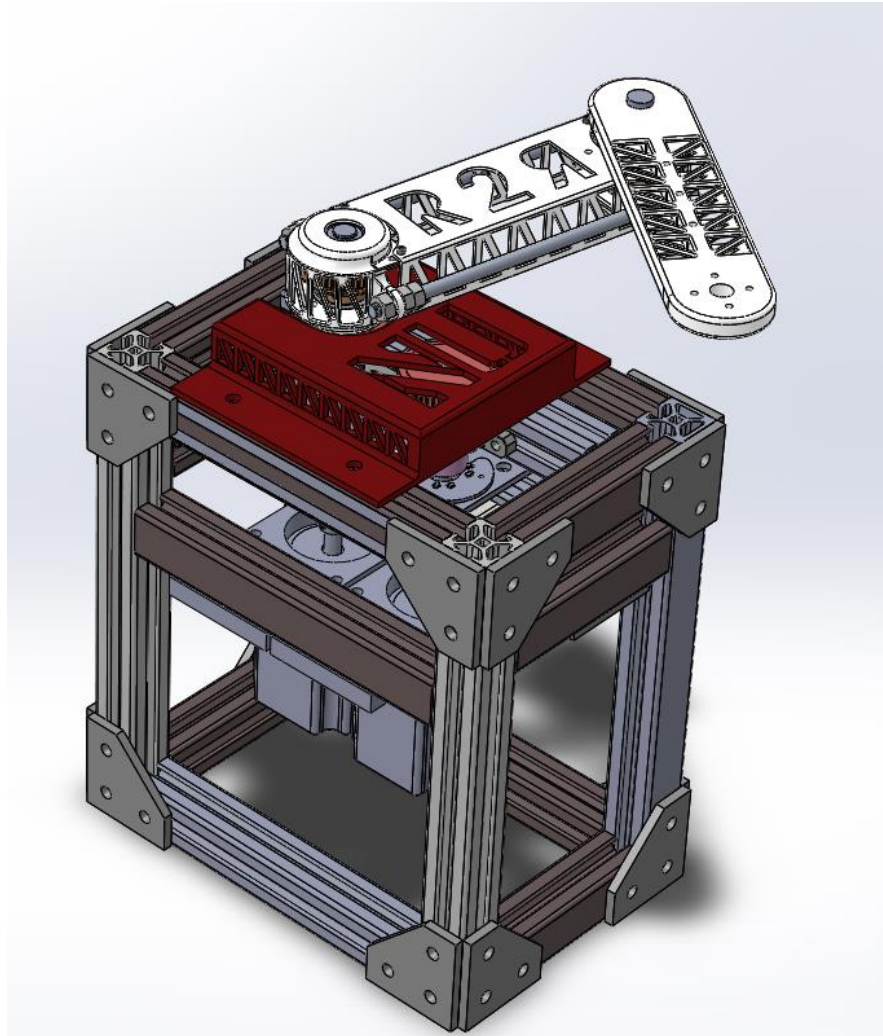


**Figure 8: Preliminary Housing design, illustrating updated link 2**

### **Present**

As in many designs, the final version of the prototype was a major step towards simplicity as compared to the previous model. The most important modification is the first link of the arm, where a fully printed arm was replaced by two end caps joined by threaded rods on either side, in order to resolve the ongoing inability to tension the belt that resided inside the arm. The second link was also redesigned, substituting an I-beam construction for the original flat panel style in order to reduce the vertical resonance vibration that was occurring when the end effector was accidentally bumped. In addition, the lower belt guard was simplified, replacing a complicated box construction with a single printed element. All 3D printing was performed in Northwestern's Rapid Prototyping Lab. The housing itself was improved as well, with the focus on adjustability of the robot. The majority of the L-brackets in the previous design were replaced with corner plates for simple construction. Two cross beams were added to the interior of the

box to allow the bearing pillow blocks to be adjustable in both the vertical and horizontal directions. Finally, the large motor plate was replaced with two independent plates that can slide along the 80/20 independently of one another, greatly simplifying the construction of the device.



**Figure 9: Final Production Prototype**

### **Electrical Design Evolution**

There were few major electrical changes between initial inception and the delivered design. The microcontroller, encoders, temperature sensors, and input power electronics remained unchanged from their selection early in the winter quarter. The voltage regulators selected changed from low-voltage input to wide, high-voltage input regulators to allow for a wide variety of motors to be used with this design. The family of regulators, the TI Simple Switchers, remained in place throughout the project. The motor drivers selected early in the first quarter were improperly spec-ed due to a misunderstanding in the motor selection process. The motors selected after resolving this error required higher-powered drivers, leading to the selection of the TI DRV83xx

series of motor drivers. Two modes of operation were requested by the client, so, both modes were made accessible via 0.1" headers and jumpers.

The requirements for including a force sensor and an IMU were dropped during the course of this project. The support circuitry for these devices remained in the final designs as an optional, but untested, feature for future expansion. The requirement for the use of bulkheads, or panel mounted connectors, was added half way through the spring quarter. This was a simple requirement to meet that further complimented the existing electronic design.

Finally, after the PCBs were fabricated and tested, a series of tests indicated the need for a few changes. A detailed description of these changes is included in the electrical final design description. However, the high-level changes included:

- Correcting the PCB footprints for the RAM and crystal
- A 180-degree rotation of three improperly placed diodes
- Removal of a short that tied the microcontroller's core voltage bus to ground
- Implementation of basic ESD protection on the primary input, 5V, and 3.3V voltage buses
- Improved UART-to-USB circuitry and an optional UART breakout for USB device bypass
- Modification of the motion controller to allow for proper trapezoidal (PWM 1x) and sinusoidal (PWM 3x) control modes
- The "tenting" of all vias and the addition of a conformal coating to provide electronics with environmental, ESD, and arc flash protections

## Final Design Description

### Mechanical Design

The final prototype consists of a simple mechanical linkage system residing in 80/20 aluminum housing, shown in Figure 10.

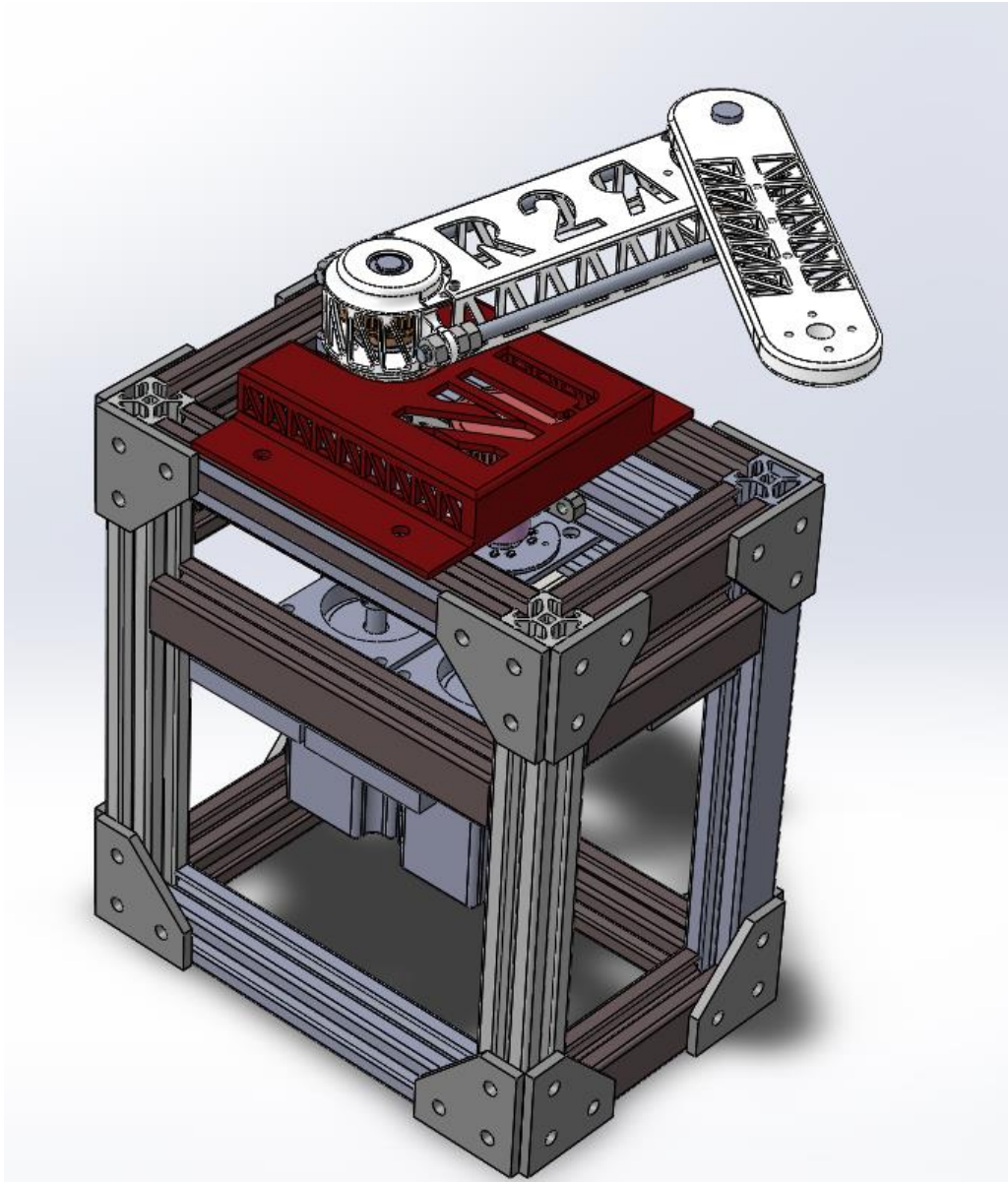


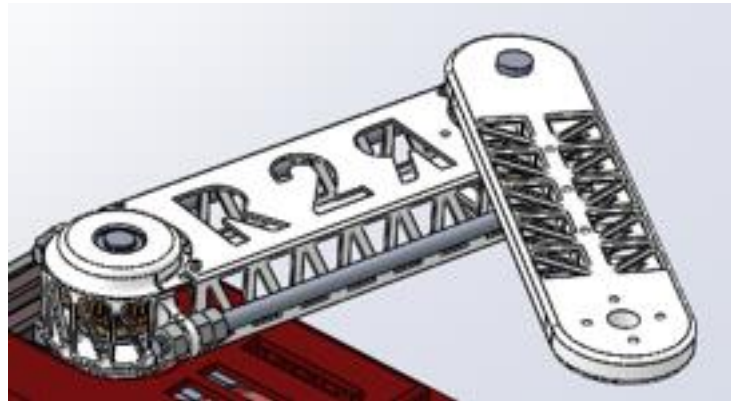
Figure 10: Full Prototype of the R2R Robot

### Arm Links

The arm section of the device consists of two linkages, each designed for form and functionality.



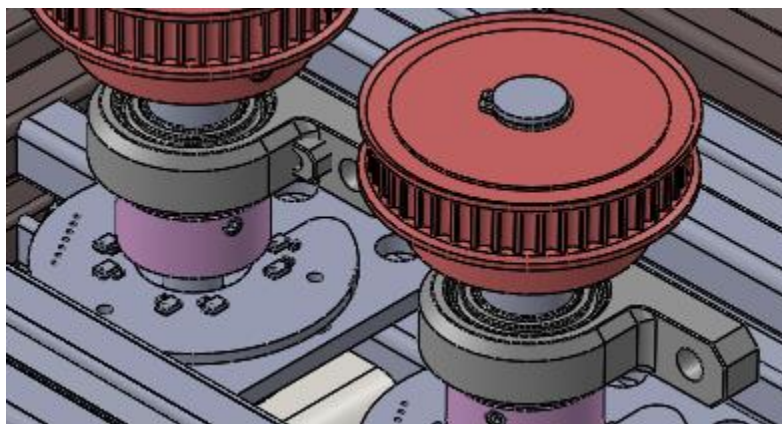
The first link consists of two printed end caps held together with threaded rods. The bearings that allow for the arm's motion are encased in either side of the caps, with the belt pulley sandwiched between them. This design allows for belt tensioning by tightening the nuts on the threaded rods. A cover spans the length of the link to protect the timing belt, and has a hole for a bolt to act as a mechanical stop if desired. The second link is a lightweight printed I-beam with a fishbone pattern on either side to minimize weight. A bolt pattern resides on the end of the link to allow for end effectors, if desired.



**Figure 11: Close-up of the arm linkages**

### **Adjustable Bearings and Motors**

All bearings are mounted directly onto adjustable elements of 80/20 inside the box frame, which allows them to slide horizontally, which in turn allows for the first belt to be tensioned properly. In addition, the motors are attached to independent mounting plates, which also allows for horizontal adjustability.

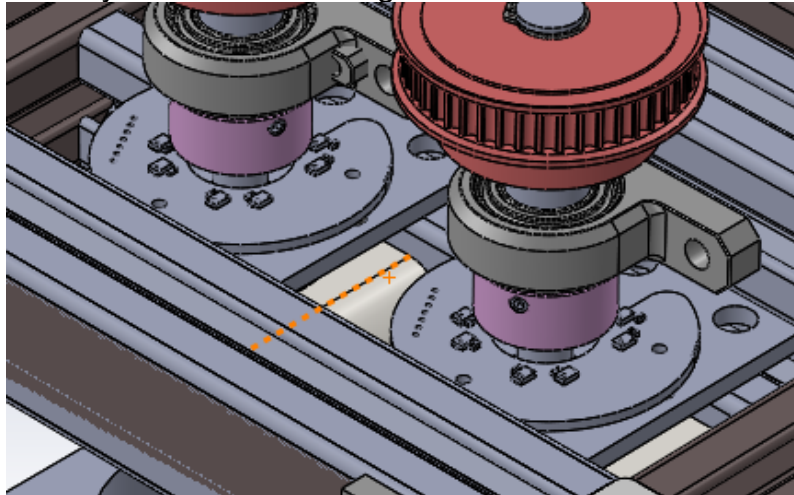


**Figure 12: Bearing pillow blocks can slide along the 80/20 channel**

### **Onboard Encoders**

The device includes two 14-bit absolute encoders that reside permanently on the shaft assemblies, allowing the same encoders to function when different motors are used.

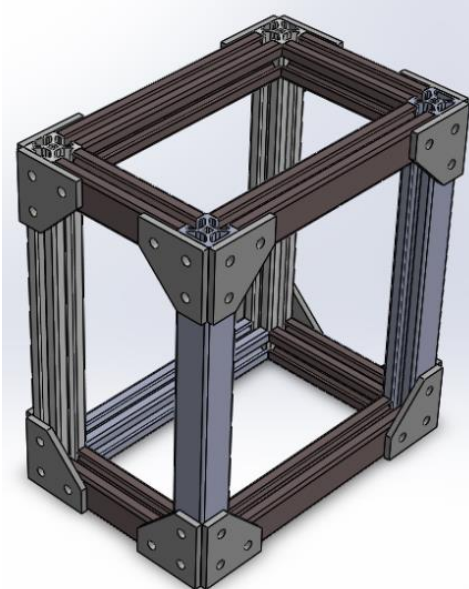
The encoders rest on platforms that can slide along a side of the 80/20 assembly, allowing for adjustability and belt tensioning.



**Figure 13: The encoders reside below the pillow blocks**

### **Adjustable Aluminum Housing**

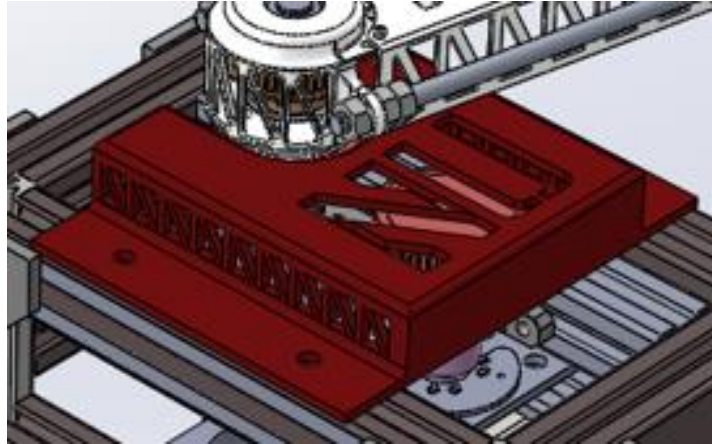
The entire robot assembly rests in an industrial-grade 80/20 casing, held together by flat corner brackets. 80/20 was selected for its ease of assembly and adjustability. The user has the option of inserting acrylic sheeting (not included) into the side slots of the box to fully enclose the internal mechanisms if desired. The user can also clamp directly to the base of the frame to keep the robot from moving during use.



**Figure 14: Exterior aluminum frame of the device**

### **Belt Guard**

A lightweight, printed belt guard rests on top of the frame and encloses the first belt, protecting it from accidental impacts that could damage it or affect its tensioning. The custom design allows the guard to slide easily into place, and attach directly to the 80/20 frame.



**Figure 15: Belt Guard shown in place**

## Electrical Design

### Changes from First Board to Second

The majority of the preproduction prototype electronics system functioned as designed. There were a few changes implemented either on the final PCB, the final design documents only, or both. The list of changes are as follows:

#### On the PCB and Documented:

Note: All device designators reference the prototype hardware. Final device designators may be found by comparing the first production schematics with the final production schematics.

- 1) The microcontroller's core voltage regulator was accidentally shorted to ground instead of being shorted to another core regulator pin. On the prototype hardware, the ground line was cut and the shorted pin (114) was white wired to the proper pin (87) on the microcontroller. On the final designs, the proper routing was implemented.
- 2) The FTDI chip (UART to USB interface) was deemed non-functional on the present PCB. The reasoning for this is unknown as FTDI's datasheets were ambiguous. This chip was bypassed and a drop in SparkFun-branded FTDI-to-USB adaptor was used with success. To implement the aforementioned, resistors R1, R3, R9, and R11 were removed to interface the FTDI chip with existing hardware. On the final designs, the open-source FTDI designs used by SparkFun were copied nearly identically to properly implement the FTDI device.
- 3) The Zener diodes on the 3.3V (D14) and 5V (D12) regulator outputs were wired backwards in the schematic. These devices were flipped on existing hardware to realize 3.3V and 5V rail functionality. The devices were placed with proper directionality on the final documentation.
- 4) The RAM footprint on existing hardware was deemed too small for the IC. The RAM was deemed unnecessary for present operation, so, it is not populated on existing hardware. Final documentation includes the proper footprint should future revisions of the board require RAM.

- 5) The input TVS diode, D10, was rated for 6V instead of 60V. This diode was removed on the prototype hardware. On the final design, the device was replaced with a Zener diode with a Zener voltage of 56V.
- 6) The crystal, Y1, on the prototype hardware used an incorrect footprint. To allay this issue, the crystal was removed on the prototype. In the final documentation, the incorrect footprint was replaced with the correct footprint.
- 7) The motor winding current sensor measurements were deemed too noisy for processing. DSP techniques were deemed too computationally complex and/or intensive for realization on the prototype hardware. An RC filter was placed on the output of both U12 devices (pin 8) to filter some of the noise. In the final documentation, RC filters were placed on each of the three outputs of the motor controller's current sense amplifiers.

#### **Only in Documentation:**

- 1) Extra TVS diodes were placed on the output of the 3.3V and 5V rails to enhance ESD protection for the circuit.
- 2) An improved TVS diode was placed on the primary voltage input to provide further ESD protection.
- 3) A UART breakout header was realized to allow for a user to bypass the onboard FTDI device to allow for quick repairs should the FTDI device fail. This header also allows for the use of other UART devices (e.g. Bluetooth to UART) in conjunction with this design. Four zero-ohm resistors must be removed to allow for this header's proper functionality.
- 4) The primary input power connector was flipped 180 degrees to allow for the MOLEX connector to lock into place. On the prototype design, the connector does not lock completely as the connector is located too close to the fuse holder.

#### **Electronics Fabrication and Assembly**

To assemble the PCB, send the ZIP FILE to any reputable PCB fabrication company. It is recommended that complete assembly be purchased.

To assemble non-PCB electronics, purchase all items listed on the "NON-PCB BOM" and follow the instructions listed in the cable harnessing documentation. Complete electronics box assembly requires review of both cable harnessing documentation and the electronics housing documentation.

#### **Test Bench Discussion**

Unfortunately, an accident near the end of the second quarter resulted in both corrected but improperly designed PCBs to be deemed irreparable and unsafe. As a result, a breadboard realization of the system was used for demonstration purposes. Significant engineering analysis indicates the inevitability of some of the errors contributing to the untimely demise of all fabricated PCBs. Other contributing factors included untidy electrical workbenches and statistical variation in component lifetimes (e.g. MTBFs). The team performed all possible repairs up and until informed to cease fabricated PCB repairs at the judgement of the electrical design lead and the project's sponsors.



## Software

### Overview

The goal in the software design was to accommodate two types of users: a basic user who would use the Arm's provided MATLAB interface, and an advanced user writing their own control laws. To achieve this, the code sacrificed some efficiency for readability, providing a high-level API for users to program the Arm without having to understand the underlying firmware. A MATLAB menu is also provided to demonstrate various functionalities of the robot graphically. Finally, care was taken to fully document the code to enable the Arm's capabilities to be extended and improved upon by future work.

The full software repository is available at <https://github.com/BlakeStrebel/2R-Robots>. The latest version of the documentation is available at [r2rproject.github.io](https://r2rproject.github.io).

### Current and Position Control

The firmware for the Arm has been written to include both current and position control with PID. The Brushless DC (BLDC) motor is controlled in 3X PWM mode with the motor driver. The commutation of the BLDC motor is done in the code and using both current sensing and the hall sensor information, the direction and magnitude of the desired current can be determined. This allows current control to be achieved at a default rate 15kHz.

Position control is achieved with the use of the two 14-bit encoders mounted on the motor shafts. A timer interrupt of 1kHz is used query the position of the motors. Both position and current control are used to control each BLDC motor for the Arm.

A more detailed description of the current control scheme is available in the following section.

### Menu Interface

There are two menu interfaces that were developed for the Arm. These were developed in MATLAB because MATLAB provides several important functionalities such as plotting, data collection, and analysis in a single package, therefore users would only have to install MATLAB to interface with the robot.

The debugging interface is provided in the main branch of the project under *Code>MATLAB*. This interface was used to test and debug the firmware. The main advantage of using this interface is that the code is transparent and most of its functionality can be accessed quickly. Adding functionality for a custom control law is also as easy as adding a few lines to the menu code. The drawbacks to using this code is that it is highly dependent on the user accessing the options in the right order, and a more complicated interface containing multiple options would increase the difficulty of maintaining the menu.

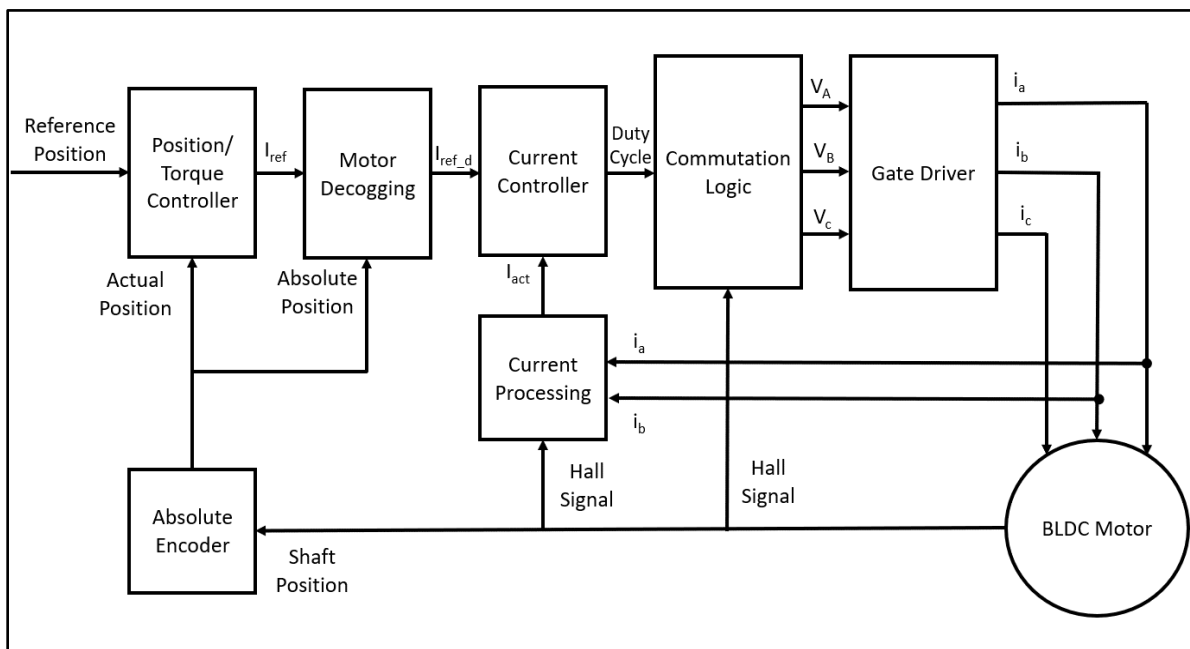
## Documentation

A QuickStart guide is provided to allow new users to get familiar with operating the arm without a need to understand the underlying code. It is intended for a new user to quickly verify that the Arm is operating as expected. The QuickStart guide also addresses potential setup questions and problems.

Further description and documentation of the code is available both as inline comments and a written documentation in Doxygen, available in html and LaTeX. An advanced user setup guide is also available to help users to familiarize themselves with the programming environment and the code. The latest version of the documentation is available at [r2rproject.github.io](http://r2rproject.github.io).

## Controls

### Overview



**Figure 16 Motor Control Flowchart**

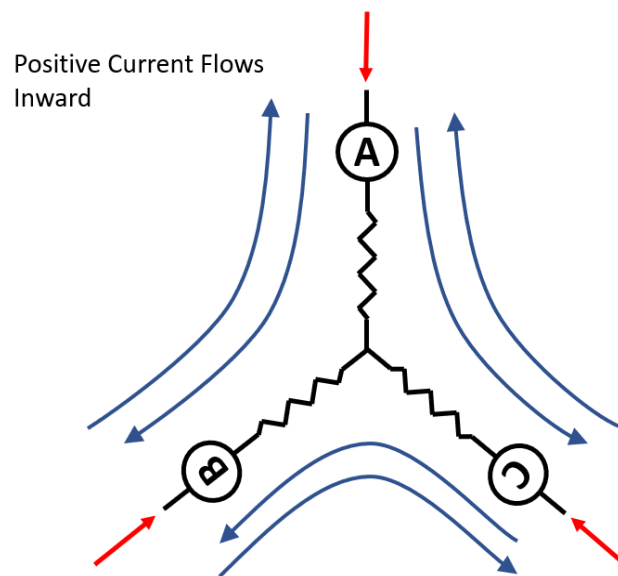
Figure 16 displays the organizational structure for control of the R2R device. In the current implementation, the outer control loop is a PID position controller running at 1KHz. The ISR which processes the data for this controller is in PositionControl.c. This ISR also sends data to the circular buffer located in Utilities.c which communicates with the Client computer over UART. The position controller calculates a reference current based on the desired and actual positions read from the absolute encoders. Depending on the absolute position of the output shaft, a motor decogging feedforward term is calculated and applied to the reference current calculated in the previous step. This is done to reduce the effects of motor torque ripple, and results in less variation in the motor torque constant. The PI current controller located in CurrentControl.c then adjusts the PWM values sent to the motor to match the desired reference current. The relevant frequencies for motor control can be found in Table 1.

**Table 1:Motor Control Frequencies**

Control Frequencies		
Parameter	Frequency	Unit
PWM	30,000	Hz
Current	15,000	Hz
Position	1,000	Hz

### Motor Commutation

The motor commutation implementation can be found in Motor.c. Figure 17 displays the sign convention for the flow of current between motor phases, and Table 2 displays the commutation table used based on hall sensor inputs A, B, and C for each motor.

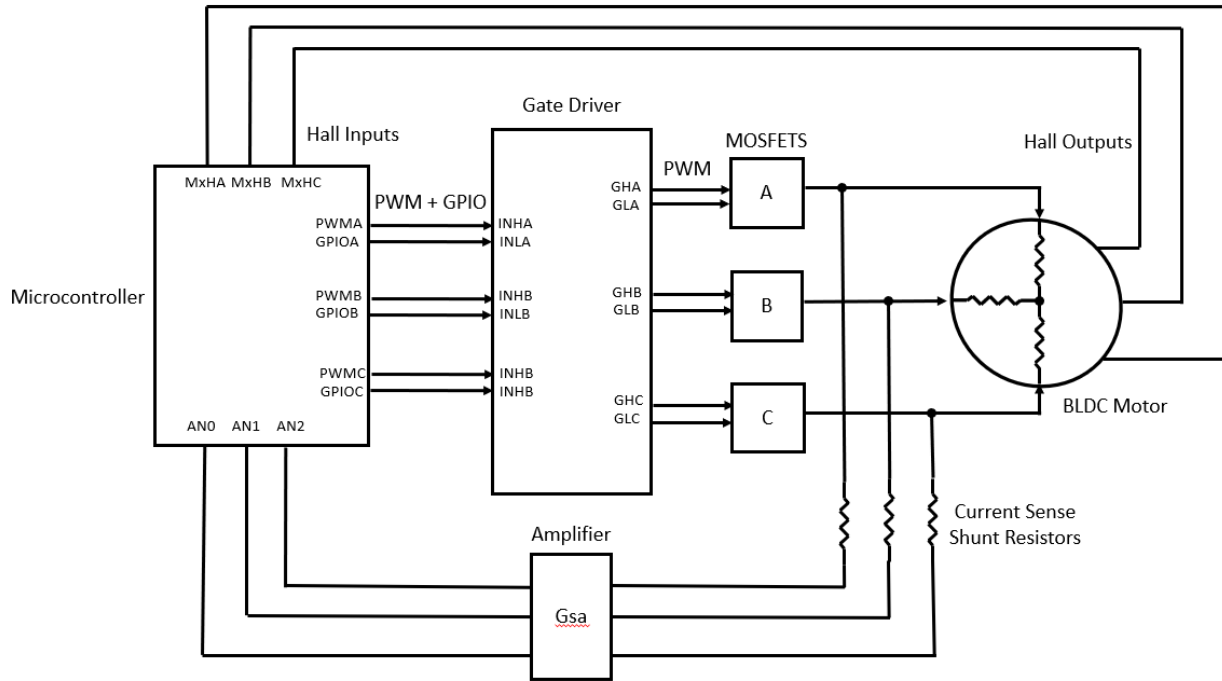


**Figure 17: BLDC Motor Current**

**Table 2: Commutation Table**

Logic and Hall Inputs							
State	CCW (Positive)			CW (Negative)			Description
	MxHA	MxHB	MxHC	MxHA	MxHB	MxHC	
0	0	0	1	1	1	0	B --> C
1	0	1	1	1	0	0	A --> C
2	0	1	0	1	0	1	A --> B
3	1	1	0	0	0	1	C --> B
4	1	0	0	0	1	1	C --> A
5	1	0	1	0	1	0	B --> A

The commutation table defines six states each representing a current flow between two phases in a given direction. MxHy are the yth hall input from the xth motor. An ISR is called on a state change of any of these inputs, and the state is updated depending on the position and desired torque direction for a given motor.



**Figure 18: Motor Driving Schematic**

Figure 18 shows the schematic including the microcontroller and motor driver for this system. The motor drivers are operating in 3x PWM mode. Therefore, each they require a PWM input and a GPIO for all three motor phases. For more information on 3x PWM mode, see the DRV8323 datasheet. Table 3 displays the inputs to the motor driver from the microcontroller, and Table 4 displays the outputs from the motor driver to the FETs for each of the current flow states seen in Figure 17.

**Table 3: Gate Driver Inputs**

State	Gate Driver Inputs						Description
	Phase A		Phase B		Phase C		
	INLA	INHA	INLB	INH B	INLC	INHC	
0	0	x	1	PWM	1	0	B --> C
1	1	PWM	0	x	1	0	A --> C
2	1	PWM	1	0	0	x	A --> B
3	0	x	1	0	1	PWM	C --> B
4	1	0	0	x	1	PWM	C --> A
5	1	0	1	PWM	0	x	B --> A

**Table 4: Gate Driver Outputs**

State	Gate Driver Outputs						Description
	Phase A		Phase B		Phase C		
	GHA	GLA	GHB	GLB	GHC	GLC	
0	L	L	PWM	!PWM	L	H	B --> C
1	PWM	!PWM	L	L	L	H	A --> C
2	PWM	!PWM	L	H	L	L	A --> B
3	L	L	L	H	PWM	!PWM	C --> B
4	L	H	L	L	PWM	!PWM	C --> A
5	L	H	PWM	!PWM	L	L	B --> A

**Current Controller**

As seen in Figure 18, in-line phase currents are sensed using amplified voltage measurements across shunt resistors. These voltages are sensed using ADC inputs on the microcontroller. In this implementation, only two phase currents are sensed for a given motor. These phase currents along with the hall information are used to determine a single magnitude and direction of current for the motor according to the information in Table 5. The current sensing and control implementations can be found in CurrentControl.c.

**Table 5: Current Sensing**

Current Readings						
State	CCW Torque (+)			CW Torque (-)		
	Phase A	Phase B	Phase C	Phase A	Phase B	Phase C
0	0	-	+	0	+	-
1	-	0	+	+	0	-
2	-	+	0	+	-	0
3	0	+	-	0	-	+
4	+	0	-	-	0	+
5	+	-	0	-	+	0

To improve current sensing performance, voltage offsets are calibrated in the startup initializations of the microcontroller. These minimize the voltage variation from zero counts when no current is flowing through the motor. Furthermore, a simple filter is applied whenever a change in the commutation state of the motor is detected. When a change is detected, the motor current is held constant for several control cycles. This is done to minimize spikes in sensed current as a result of dead-time and back-emf.

After determining the magnitude and direction of sensed current, this value is compared to a desired value and a PI controller is used to calculate the resultant PWM duty cycle which is used to control the motors.

## Motor Decogging

Non-ideal commutation and motor imperfections cause motor output torque to vary for a given motor current. This torque ripple complicates control, and therefore was minimized using a motor decogging procedure. This implementation is in PositionControl.c. For more information motor cogging, as well as to calibrate the decogging equations for a new motor, see the supplemental information.

## Testing Plans

### Mechanical

#### Friction Testing: Ball Bearings

In order to minimize friction, multiple options were considered for the bearings and the timing belts. To test these options, a simple test rig was set up, with pillow blocks to hold the bearing options and simple shafts turned to fit into the bearings. The various bearing options (glass vs. metal ball bearings) were tested, and the glass bearings were qualitatively observed to have less friction than the metal bearings, so the glass ball bearings were selected for the final design.

#### Friction Testing: Timing Belts vs. Posi-Drive

After the bearing selection was made, the timing belt options (3mm rubber timing belt, 9mm rubber timing belt, and posi-drive) were tested on the same set up as above. The posi drive belt was determined qualitatively to have slightly less friction than the 3mm timing belt, while the 9mm timing belt had by far the most friction. However, the posi-drive we had ordered (and had been selected by the distributor) was determined to be too weak for the anticipated loads. We selected new posi-drive belts and pulleys that would be sufficient for the loads, however they were backordered. Given that there was only a small frictional difference between the posi-drive and the 3mm timing belts, we decided that it would be best to use the 3mm timing belt for our final prototype as it still reduces friction and we can get the parts.

#### Motor Characterization:

To reduce the cogging effects of the motor, a position control feedback system was implemented. This allowed us to measure how much cogging was happening in the motors, and implement a system to reduce the effects of cogging so that the robot has fewer uncharacterized effects and is more transparent.

## Electrical: Benchtop

### Motors + SPI Device

- 1) Build power supply AC and DC cable bundles. Check the AC input voltage and DC output voltage. Adjust supply voltage to 24.5VDC.
- 2) De-energize all power systems.
- 3) Populate R36, R37, and R38 with 0-ohm resistors (or solder) as device is to be used in PWM -1x mode.

- 4) Connect motor controller to development board.
- 5) Clamp motor to table, gently
  - a. Enough force to hold the motor stable, but not so much to damage it.
- 6) Connect motor-to-motor controller.
- 7) Connect motor halls to motor controller
- 8) Connect development board to computer
- 9) Connect power to motor controller.
- 10) Energize system and check for error indicators
  - a. If error indicator, follow datasheet.
- 11) Assist software team in debugging and setting up interface software.
  - a. Since the controller uses SPI, this test also certifies the functionality of the SPI bus.
- 12) Test motor's motion:
  - a.  $\frac{1}{4}$  peak current
  - b.  $\frac{1}{2}$  peak current
  - c. Peak current
  - d. Spin motor forward and backwards
  - e. Test motor holding position
  - f. Test simple speed control using halls only

## Encoders

- 1) Build encoder test rig with mechanical team. In this phase, follow RLS guidelines to ensure the encoder read head and magnetic ring are aligned with proper tolerances
- 2) Connect wires to encoder board to interface encoder to microcontroller. Label each wire for software to interface.
- 3) Before mounting encoder on test rig, power on the encoder and follow RLS guidelines to ensure proper power up occurs.
  - a. Typically, a success indicator is an LED with a specific flash pattern.
  - b. If the encoder does not pass this test, call RLS. The device may be defective.
- 4) Mount the encoder on the mechanical rig. Power on the encoder. To check alignment, check the RLS guidelines. Disconnect power upon completion.
  - a. Typically, a success indicator is an LED with a specific flash pattern. This pattern will differ from the power on success pattern.
  - b. If the encoder does not pass this test, work with mechanical to resolve tolerance issues.
- 5) Connect encoder to microcontroller's SPI bus for software team. Allow software team to test encoder interface.
- 6) Test encoder spinning forward:
  - a.  $\frac{1}{4}$  turn
  - b.  $\frac{1}{2}$  turn
  - c. Full turn

- d. Test velocity (spin ring freely and observe data, does it start fast and slow down?)
- e. Check position: does the encoder
- 7) Repeat 6 (a) through (e) for backward spins.

### I2C Device

- 1) Solder leads onto the I2C device breakout board
- 2) Connect device to microcontroller not forgetting SDA/SCL pull up resistors.
- 3) Assist software in communication protocols
- 4) IMU: move IMU in 3D space and ensure data is logical.
  - a. Rigorous testing of this sensor is unnecessary as it could be dropped from the design.

### Analog Device (Temperature Probe)

- 1) Using the software team's ADC code, perform a 0V to 3.3V sweep. Review all data for errors or noise.
- 2) Note analog noise and propose methods to fix, if necessary.
- 3) Setup the temperature probe breakout board according to the datasheet. This should involve soldering headers to the board and applying 3.3V power.
- 4) Strip a K-Type thermocouple for use by the probe.
- 5) Connect the sensor to the microcontroller.
- 6) Using the ADC code, test the thermocouple with a series of different temperature objects (ICE, human, laptop, room temp). Convert raw values using the formula found in the data sheet.

### Electrical: PCB

The following describes the methods of testing the critical electronic subsystems. All other electronic subsystems have built-in testing procedures that will alert the user to system failures.

### Input power:

- a. Insert 25A fuse into F1 fuse holder
- b. Locate a power source of at least 22VDC, capable of sourcing at least 22A
- c. Attach to this power supply a 6-pin MOLEX MiniFit Jr female housing. Polarity of the pins are as follows: 1-3 positive, 4-6 negative
- d. Ensure power supply is deenergized
- e. Attach MOLEX connector to P8.
- f. Energize power supply
- g. Observe
  - i. D11: Is it on? If not, stop. The main power input has failed
  - ii. D17: If not on, 3.3V has failed
  - iii. D14: If not, 5V has failed.
  - iv. If all are on, power regulation is functioning as planned. To be certain



1. Probe TP6 and ensure this value is nearly 5V with low ripple
  2. Probe TP7 and ensure this value is nearly 3.3V with low ripple
- v. **If these tests fail, do not proceed to test any further. This is a critical failure and a dangerous situation.** Debug power electronics by looking for hot components, damaged components, or stray wires causing shorts.

#### Microcontroller:

- a. Ensure the electronics are deenergized.
- b. Attach the JTAG connector to either the PCB (P2) or the JTAG bulkhead (clearly labeled).
- c. Attach to the JTAG connector a JTAG programmer of your choice.
- d. Energize electronics. Ensure power LEDs turn on as described in 1.g (i-iii). If not, there is likely a short. Do not continue testing procedures.
- e. Attempt to program the microcontroller using Code Composer Studio or an environment of your choice. You may use any program for this test that does not command the motors (a while loop that does nothing is suggested, but something that toggles D4-D6 may also be interesting).
- f. If programming succeeds, the microcontroller is functioning. If programming fails, the microcontroller is likely faulty. **A faulty microcontroller likely requires the placement of a new device.**

#### UART Communications:

- a. Attach a USB cable to the USB bulkhead on the enclosure.
- b. Using the attached JTAG device from step two, write to the microcontroller a UART test program. This program should echo characters from the terminal.
- c. Using PuTTY or another serial terminal of one's choice, test the echo program
- d. If this test fails, the UART communications have failed. Either:
  - i. The UART on the microcontroller is faulty
  - ii. The onboard UART-to-USB chip is faulty
  - iii. The drop-in device-to-UART dongle is faulty (e.g. Bluetooth to UART)
  - iv. The connections between the device and the PCB is faulty.
- e. To debug the above
  - i. First test continuity of all wires tied to the UART line.
  - ii. Then, test the USB cable(s) functionality.
  - iii. If using external UART-to-device tools, test their functionality and ensure the following resistors were removed from the PCB: R7, R9, R10, R11
  - iv. If using the onboard UART-to-USB device, try using an external UART-to-USB device (e.g. SparkFun Dev-09873). You may use P1

for this test. Ensure resistors R7, R9, R10, and R11 are removed from the PCB for this test.

- v. If all of the aforementioned appear to be functional, assume the microcontroller is faulty. **A faulty microcontroller likely requires the placement of a new device.**

### Encoders:

- a. Ensure the electronics are deenergized.
- b. Ensure encoders are properly installed on motor drive shafts. Follow the Renishaw/RLS guidelines set forth in their datasheet to ensure the redhead and magnetic ring are properly aligned.
- c. Connect the encoder to the electronics via the properly labeled bulkhead on the electronics housing for either motor 1 or motor 2.
- d. Energize electronics. Ensure power LEDs turn on as described in 1.g (i-iii). If not, there is likely a short. Do not continue testing procedures.
- e. Ensure the microcontroller has been loaded with the primarily packaged software or software that tests encoder functionality.
- f. Check the encoders' status LEDs. The LEDs should be solid green shortly after turning on the electronics. If not:
  - i. RED/ORANGE: misalignment of encoders. See mechanical debugging guidelines
  - ii. GREEN BLINKING: no alignment issues, no clock signal from microcontroller. This is a software issue. Try flashing the microcontroller with the packaged code to test.
- g. After verifying encoder LED status, use the software interface (see software guidelines) to ensure proper functionality. Suggested testing:
  - i.  $\frac{1}{4}$  turn – do the encoder values increase by 90 degrees?
  - ii.  $\frac{1}{2}$  turn – do the encoder values increase by 180 degrees?
  - iii. Full turn of encoder – do the encoder values increase by 360 degrees
  - iv. Failure of any of the above indicates an error with either the software or the encoder. If an encoder fails, contact Renishaw to purchase a replacement read head or repair the device if its warranty remains active.
- h. If the encoders do not function, **do not proceed to test the motors.**

### Motors and Halls:

- a. Ensure the electronics are deenergized.
- b. Connect motors and halls to the electronics via the properly labeled motor and halls bulkheads.
- c. Connect encoders to electronics enclosure if removed from step 4.
- d. Energize electronics. Ensure power LEDs turn on as described in 1.g (i-iii). If not, there is likely a short. Do not continue testing procedures.

- e. Monitor electronics and motor temperature (via a temperature probe) to ensure no electronics appear to increase in temperature.
- f. Ensure all hardware is properly clamped to the workbench.
- g. Using the provided software, command each motor, one at a time, to move in both the positive and negative directions using PWM commands as described in Table 6.
  - i. During both the first and second tests, press the emergency stop button located on the side of the electronics housing. Ensure the motor stops spinning in both cases. **If the E-Stop does not stop the motors, do not proceed, a critical failure has occurred and the system is not safe for use.**
  - ii. After a pressing the E-Stop, release it. The electronics motors should move again. If they do not, reset the system via the reset button also located on the electronics enclosure. If the reset button does not work, power cycle the system. If the motors still do not spin, **a critical failure has occurred and the system is not safe for use.**
  - iii. If the procedures pass with success, continue the testing described in Table 6.

**Table 6: Motor Testing Sequence**

M1 Command	M2 Command	M1 Behavior	M2 Behavior
10	0	Spins slowly CW	No motion
0	10	No motion	Spins slowly CW
-10	0	Spins slowly CCW	No motion
0	-10	No motion	Spins slowly CCW
10	10	Spins slowly CW	Spins slowly CW
-10	10	Spins Slowly CCW	Spins slowly CW
50	0	Spins FAST CW	No motion
0	50	No motion	Spins FAST CW
50	50	Spins FAST CW	Spins FAST CW
-50	50	Spins FAST CCW	Spins FAST CW

- h. If all tests behave as described, the system is behaving as designed and is ready for classroom use. If not, a motor controller or motor may be damaged. **Do not continue to commission the device if this scenario occurs. Contact support staff.**

## Performance Results

### Friction Reduction

There has been a huge reduction in friction in the robot as we have switched from using needle bearings to glass ball bearings and improved the fits in our final prototype. The robot can act like a pendulum even when not powered by the motors, which indicates how minimal of friction there is in the final prototype and design.

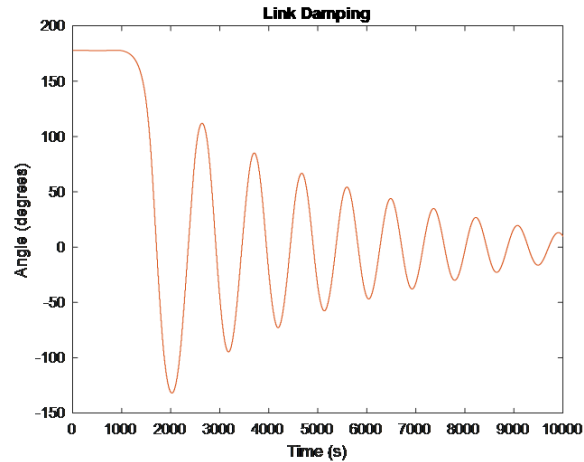


Figure 19: Damping of Link due to Friction

### Motor Characterization

This was very successful, as cogging effects were reduced up to 100%. The error in position between the controlled (AntiCogged) and raw systems can be seen in Figure 20 below.

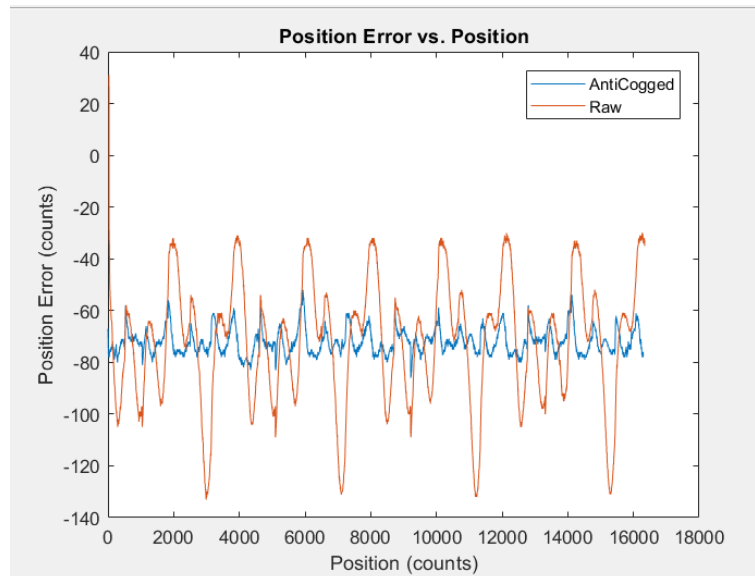
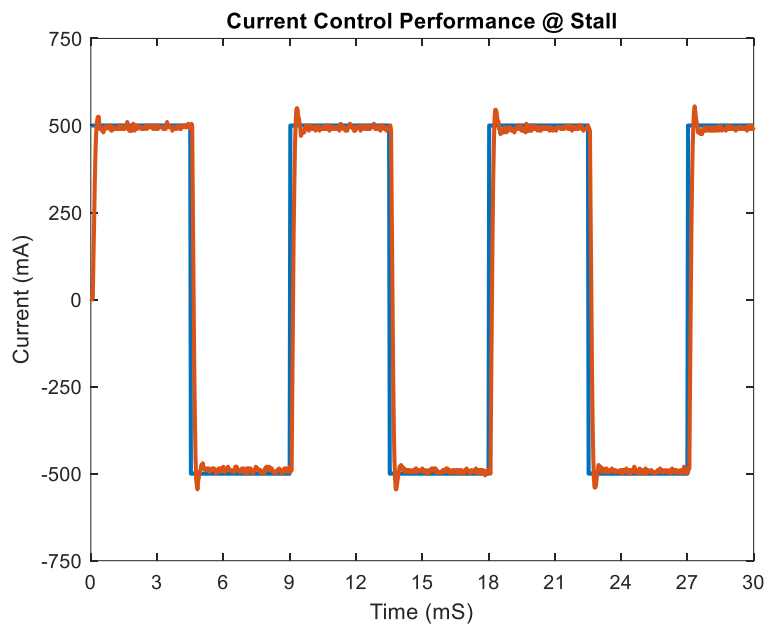


Figure 20: Motor Characterization Results

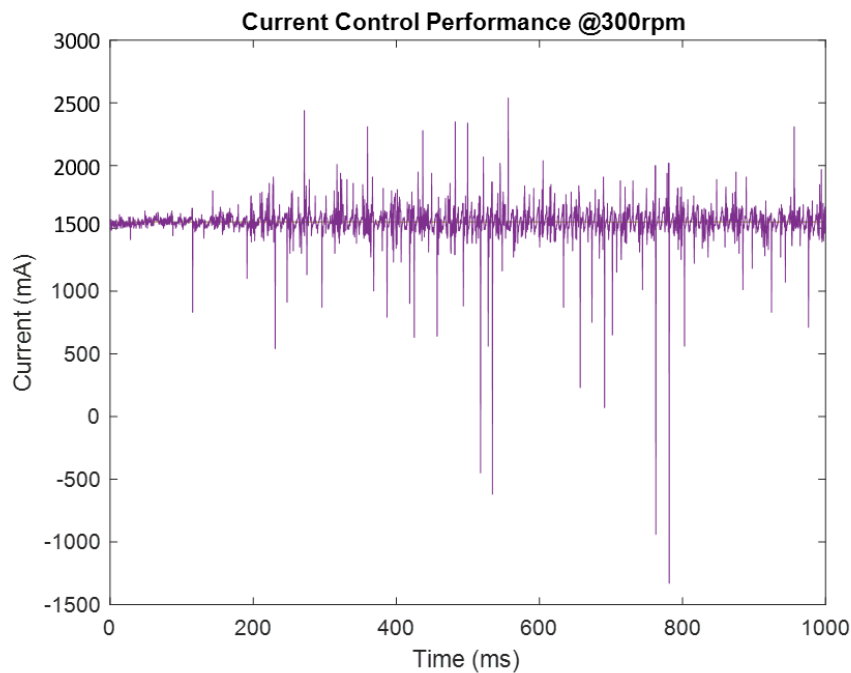
## Current Control

Figure 21 demonstrates that there is very little error in tracking a reference current when the motor is stationary.



**Figure 21: Current Control Performance @ Stall**

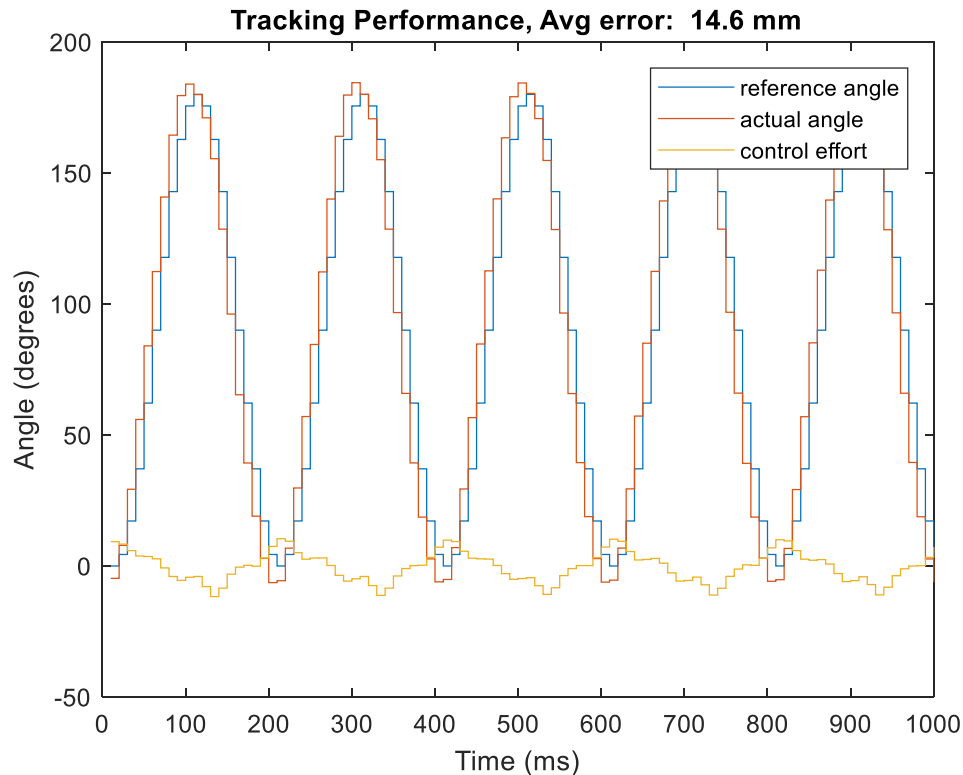
Figure 22 demonstrates increased noise in current tracking performance when the motor rotor is rotating and the commutation state is changing. Regardless, the controller is able to reject the majority of this noise and still track the reference current well.



**Figure 22: Current Control Performance While Moving**

## Position Control

Figure 23 displays that the motors can track a reference trajectory with a high level of accuracy even at relatively high frequencies and magnitudes. The data presented in Figure 23 has been decimated so there are 100 total data points for each curve.



**Figure 23: Position Tracking Performance**

## Next Steps

### Mechanical

There are a few mechanical changes that could be implemented in the future, if desired:

- Posi-drive could be used in place of the timing belts if parts are in stock in order to further reduce friction or examine the control differences between traditional timing belts and posi-drive
- End effectors could be added for running more advanced control laws
- Gear boxes can be added to the motors and/or alternative motors used to test the effects of various motors/gearing

### Electrical

To improve the R2R learning robot's electronics, the following next steps are suggested for implementation. It must be noted that this is not an exhaustive list.

- The two motor controllers may be placed closer together to shrink the size of the control electronics. This will lower manufacturing costs.
- Placing components on both the top and bottom layers of the PCB to further shrink the electronics. This is likely a cost-neutral revision.

- The voltage input range may be shrunk from a range of 22V-56VDC to a range of 22 - 36VDC. This will allow for the following improvements:
  - o The relaxation of the voltage rating requirements for the motor controller's bulk capacitors; this allows for physically smaller and cheaper capacitors to be used.
  - o The use of lower input voltage DC-to-DC regulators; this allows for lower cost and physically smaller 3.3V and 5V DC-to-DC regulators. Further, this minimizes the regulator's support circuitry requirements and will further shrink the 3.3V and 5V regulator's footprints.
  - o The use of more accessible and lower cost power electronics on the input voltage protection lines.
- The removal of testing components to lower cost and bring the PCB closer to a full production device. This also includes "tenting" all exposed vias to protect the board from environmental shorting.
- Implementation of the force sensor and IMU, first via existing electronics, and then via novel designs to implement more advanced sensing hardware.

### Electrical System Incidents

Following PCB fabrication, an unfortunate number of electrical design issues arose. All are documented here with causes and fixes, both implemented and proposed. All are documented in the order of appearance. Detailed solutions to all issues are provided in electrical next steps, electrical changes (from First Board to Second), evolution of electronic design sections. All reference designators mentioned refer to the designators in the final version of the schematics.

**Incident:** System does not energize

**Issue:**  $V_{zeener}$  on input power protection diode (D9) too low; short from input power to ground created.

**Solution:** Remove diode, replace with diode with higher  $V_{zeener}$  rating. Replace part number in design files.

**Incident:** 3.3V, 5V rails did not supply power

**Issue:** D12, D15 facing wrong direction creating a short from 3.3V and 5V to ground

**Solution:** Remove D12 and D15 and flip component on PCB. Rotate component in design files.

**Incident:** Failed JTAG communication/programming

**Issue:** (1) Core voltage of microcontroller (pin 115 on U2D) shorted to ground; (2) mirrored crystal oscillator PCB footprint

**Solution:** (1) Cut trace between pins 114 and 115, short pin 115 to pin 87 as described in microcontroller data sheet. Correct design files with new traces in layout. (2) Remove crystal oscillator and use internal oscillator for prototype, use proper footprint in design files.

**Incident:** 3.3V bus fire, irreparably damaging entire PCB

**Issue:** Metal shavings and loose wires visibly present on workbench, created short on 3.3V regulator output

**Solution:** Prepare new prototype PCB, clean work bench, implement new PCB handling procedures.

**Incident:** Electrostatic discharge event, irreparably damaging microcontroller

**Issue:** Improper grounding of engineer, lack of ESD protection on PCB

**Solution:** implementation of ESD protection for engineers at workbench, addition of fast-acting TVS diodes on all voltage buses (in design files) to short all voltage spike above  $\sim 10\% V_{bus}$  to ground

**Incident:** Twice noticed, visible and audible arc flashes on motor power bus on PCB

**Issue:** Excess energy storage in motor windings (acting as a large inductor) during improper commutation creating a capacitive environment; build-up of charge between high current sourcing traces supplying pins 1 and 2 of header P16 (motor supply), leading to short on lines on PCB as the result of exposed vias (non-tented) and close via placement; arcing possible as the result of  $> \sim 10$  mils spacing between high energy vias.

**Solution:** At manufacturing time, request all vias to be tented. After manufacturing and PCB testing, apply a high-temperature conformal coating to all PCB components.

## Software

There are some software changes that could be implemented in the future, if desired:

- Computational bandwidth should be evaluated. A single microcontroller is being used to do position control, current control, and commutation for two motors as well as to communicate with the Client computer. This may limit the computational complexity of custom controllers written for the device.
- Custom control laws can (and should) be written
- Field-Oriented-Control could be developed and implemented. This would reduce torque ripple and increase efficiency of motors. Utilizing a microcontroller from TI's C2000 series would simplify the implementation process.
- End effectors could be added for advanced learning
- Hierarchical menu can be further tested and released. The second hierarchical interface, available under *BackupAndUI>R2R\_interfaces* addresses the issues of the debugging menu by providing options in a tree format, and it provides a more structured way for users to control the Arm. This makes the arm more accessible to new users and it also provides useful feedback on what part of the process the users are in. The main drawback to this interface is that adding new functions requires more work than the debugging interface.
- A Graphical User Interface can be further developed and tested for the device.



### **List of Other Documentation**

Along with this report, there is also a quickstart guide, an advanced software user manual (d-oxygen), a motor decogging guide, and an assembly guide.

## Appendix A: Shaft Analysis

Overall Parameters					
Material	6061 Aluminum				
Ultimate Tensile Strength (MPa)	310				
Tensile Yield Strength (MPa)	276				
Output Torque (N*m)	0.5				
Failure Analysis					
Shaft 1 & 3					
Dimensions and Parameters			Factor of Safety		
Dimension	Value	Unit	Test	Value	
a	0.03985	m	Goodman's Li	2.862549062	
b	0.01119	m	Peak	13.26698401	
c	0.01119	m			
Pulley Pitch Diameter	0.0344	m			
Pulley Bore	0.00635	m			
Tight/Slack Ratio	5				
Stress Calculations			Stress Concentrators		
Parameter	Value	Unit	Concentration	Value	Notes
Tension Tight	36.3372093	N	Fatigue Stress	3	
Tension Slack	7.26744186	N	Surface Factor	0.9862015778	Machined Shaft
Radial Force	43.60465116	N	Size Factor	1	d = 8mm
Tangential Force	29.06976744	N	Reliability Fac	0.82	99%
Axial Force	0	N	Temperature f	1	
Bx	14.53488372	N	Miscellaneous	1	
Bz	21.80232558	N	Loading	0.29	Cyclical Torsion
Bearing Load	26.20313427	N	Endurance Lim	72.70080791	Mpa
Mz_max	0.1626453488	N*m			
Mx_max	0.2439680233	N*m			
My_max	0.2932130725	N*m			
Bending Stress Amp (No Factor	11.66439609	Mpa			
Bending Stress Amp (Factor Ap	34.99318826	Mpa			
Bending Stress Mean	0	Mpa			
Shear Stress Mean	9.945324051	Mpa			
Shear Stress Amplitude	0	Mpa			
Von Mises Stress Amp	34.99318826	Mpa			
Von Mises Stress Mean	17.22580655	Mpa			
Peak Von Mises	20.80352248	Mpa			

Shaft 2					
Dimensions and Parameters			Factor of Safety		
Parameters	Value	Unit	Result	Value	
a		m	Goodman's Line	1.650360344	
b	0.0195	m	Peak	10.35686646	
c	0.025485	m			
d	0.0127375	m			
Pulley Pitch Dia	0.0344	m			
Pulley Bore	0.01	m			
Idler Pitch Diam	0.0344	m			
Idler Bore	0.008	m			
Thrust Bearing	0.0127	m			
Tight/Slack Rat	5	tight/slack			
Stress Calculations					
Calculations	Value	Unit	Stress Concentrators		
Pulley Tension	36.3372093	N	Concentration #	Value	Details
Pulley Tension	7.26744186	N	Fatigue Stress	1	
Pulley Radial Force	43.60465116	N	Surface Factor	0.8862015778	Machined Shaft
Pulley Tangential Force	29.06976744	N	Size Factor	1	d = 8mm
Pulley Axial Force	0	N	Reliability Factor	0.82	99%
Idler Tension T1	36.3372093	N	Temperature Factor	1	
Idler Tension S1	7.26744186	N	Miscellaneous Factor	1	
Idler Radial Force	43.60465116	N	Loading	0.29	Cyclical Torsion
Idler Tangential Force	29.06976744	N	Endurance Limit	72.70080791	Mpa
Idler Axial Force	0	N			
Bx	29.06976744	N			
Bz	43.60465116	N			
B_Radial_Load	52.40626854	N			
Cx	9.687387345	N			
Cz	14.53108102	N			
C_Radial_Load	69.87045444	N			
Dx	19.3823801	N			
Dz	29.07357015	N			
D_Radial_Load	34.94208264	N			
Mx_B_Offset	0.8502906977	N*m			
Mz_B_Offset	0.5668604651	N*m			
By1	66.95202344	N			
By2	44.63468229	N			
B_Axial_Load	80.46631784	N			
Mx_B	0.8502906977				
Mz_B	0.5668604651				
M_B	1.021922237				
Mx_C	0.3703245997				
Mz_C	0.2468830665				
M_C	0.4450747776				
M_max	1.021922237	N*m			
Bending Stress	20.33049692	Mpa			
Bending Stress	20.33049692	Mpa			
Bending Stress	0	Mpa			
Shear Stress M	9.947183943	Mpa			
Shear Stress A	0	Mpa			
Von Mises Stress	17.22902798	Mpa			
Von Mises Stress	20.33049692	Mpa			
Peak Von Mises	26.64898704	Mpa			

## Appendix B: Link Properties

Element	Weight (g)	Length (if applicable)	Material	Notes			
Link End 1	22.26		ABS				
Link End 2	22.75		ABS				
Belt Cap	17.5		ABS				
Link 2	47.56	6.13 in	ABS	center to center			
Nuts x 12	0.45		Steel	5.4 g total			
Threaded rod x2	14.92	7 in	Aluminum	29.84 g total			
Small Pulley x2	30.91		Aluminum	61.82 g total			
Shaft 3	12.32		Aluminum				
Glass Bearings x	4		Plastic/Glass	16 g total			
		CoM Location (in)					
<b>Relevant Values for PenduBot/AcroBot Equations of Motion</b>							
Link1 (CoM measured from Shaft 2) (Raw data from Solidworks)							
Threaded Rods	29.84	3.5	Aluminum				
Nuts x6	2.7	6.045	Steel				
Link End 2	22.75	6.045	ABS				
Small Pulley	30.91	6.045	Aluminum				
Belt Cap	17.5	3.023	ABS				
Glass Bearings x	8	6.045	Plastic/Glass				
Shaft 3	12.32	6.045	Aluminum				
Long Belt	1.68	3.023	Neoprene				
Link2 (CoM measured from Shaft 3)							
Link 2	47.56	3.065	ABS				
				M*x			
Convert everything to metric MKS system, calculate CoM							
Threaded Rods	0.02984	0.0889	Aluminum	0.002652776			
Nuts x6	0.0027	0.153543	Steel	0.0004145661			
Link End 2	0.02275	0.153543	ABS	0.00349310325			
Small Pulley	0.03091	0.153543	Aluminum	0.00474601413			
Belt Cap	0.0175	0.0767842	ABS	0.0013437235			
Glass Bearings x	0.008	0.153543	Plastic/Glass	0.001228344			
Shaft 3	0.01232	0.153543	Aluminum	0.00189164976			
Long Belt	0.00168	0.0767842	Neoprene	0.000128997456			
Total Mass	125.7	grams	Total M*x	0.0158991742			
CoM	0.000126485077	meters					
	4.979727446	inches					
Link 2	0.04756	0.077851		0.00370259356			
Total Mass	47.56	grams					
CoM	0.077851	meters					
	3.065	inches					