

Novice Fishing Pole Design

A Major Qualifying Project Report:  
JMS0701

submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

---

**Cale C. Putnam**

---

**Patrick J. Quinn**

Date: April 26, 2007

Approved:

---

**Professor John M. Sullivan, Jr., Major Advisor**

## Table of Contents

Abstract.....	5
Introduction.....	6
Areas of investigation.....	7
Concept 1: Carbon dioxide (CO <sub>2</sub> ) powered launcher.....	7
Concept 2: Crank-back fishing catapult.....	8
Concept 3: Cast-training arm brace/training rod.....	8
Concept 4: Cock-back fishing rod.....	9
Concept 5: Casting tunnel.....	10
Concept 6: Casting motion robot.....	11
Concept 7: Novice Fishing Pole.....	11
Experiments needed for design.....	12
Casting Experiment.....	12
Explanation of Experiment.....	12
Experiment.....	12
Conclusion of Casting Experiment.....	14
Constant Diameter Rod Equation Experiments.....	15
Force Equation experiment.....	20
Pole Measurement Experiment.....	22
Strain Gage Experiment.....	23
Proposed Strain Gage Configuration.....	23
Strain Gage Placement.....	24
Explanation of Experiment.....	25
Hypothesis.....	25
Results.....	27
Conclusion.....	27
Computer Model.....	31
LabView Experiment.....	35
Single Accelerometer Experiment.....	38
Experiment.....	39
Results.....	39

Results.....	40
Fishing pole design.....	40
Accelerometer setup.....	40
Accelerometer Placement on Pole and Arm.....	41
Wiring Configuration of the Pole and Arm.....	44
Mathematics.....	46
Two Dimensional Space Equations.....	46
Three Dimensional Analysis.....	52
Final LabView VIs.....	57
Formula Node Calculations.....	66
First VI.....	67
Moving Average.....	67
Zero Out.....	68
Conversion to Acceleration.....	68
Second VI.....	69
Trigonometric Functions.....	69
Acceleration to Position.....	69
Conclusions.....	72
Bibliography.....	73

## **Acknowledgements**

We would like to thank Professor John M. Sullivan Jr. for his continued support, assistance, guidance, and patience. We would also like to thank Professor Cosme Furlong for equipment and advice as we moved through the project. We would also like to thank Worcester Polytechnic Institute's Mechanical Engineering department for the use of their equipment and facilities.

## **Abstract**

Novice fishermen commonly have difficulties executing a proper cast. Therefore we created a fishing pole which is able to map the motion of the cast, thus enabling problems with the motion to be identified. This was accomplished by means of 14 accelerometers nested in triads to calculate the absolute acceleration vectors at multiple locations on the pole as well as the arm. These sensors were attached at the tip, middle, and butt end of the pole and on the upper arm and forearm of the caster. Using LabView, the accelerations were translated from their local coordinate system to the world coordinate system, and then integrated to track the change in position of each sensor as a function of time. This data is graphed and saved in a spreadsheet file for further analysis.

## Introduction

The goal of our MQP was to design a fishing pole to assist the cast of either a novice or a handicapped fisherman. After doing some background research we found quite a few designs of a fishing pole that assists a handicapped person and not as many that assist a novice fisherman. Therefore we decided to pursue the design of a fishing pole that would assist a novice fisherman. After observing our own casts we realized that there are many variables that have an effect on a cast which allow for many instances in which a novice fisherman's cast can go awry. Some of the factors that play a role in making a good cast are such things as certain speeds that need to be obtained by each joint, certain angles that need to be met at different points in the fisherman's arm and pole and also a certain smoothness that needs to be carried throughout the cast, and these are just a few. Seeing as there are so many variables that factor into a good cast we decided that we would need to create a way to quantify the cast at numerous locations both spatially and temporally. We decided the best way to do this would be to create a fishing pole and arm bands that could track the casters motion. This way we could record each motion made by the caster and compare them to casts performed by a qualified caster. This would allow for the fisherman to focus on their particular problem area(s) to perfect their cast. We placed accelerometers strategically on two arm bands, just above the elbow and wrist, and instrumented the tip, base and mid sections of the pole. Readings from these sensors would allow us to translate the given voltages into accelerations and later into spatial positions at each of the accelerometers, allowing us to track each cast through time. We read these readings through a software program, National Instruments Measurements and Automation, and translated them into positions via the software program LabView.

## ***Areas of investigation***

We brainstormed several ideas of potential devices for investigation, with the intent of narrowing them down to one specific concept to develop. Along with creating summaries of the ideas, we also estimated where the potential market for each, a preliminary materials list, and assumed pros and cons that would be used to evaluate the concepts.

### **Concept 1: Carbon dioxide (CO<sub>2</sub>) powered launcher**

This concept is based on the principles behind paintball guns. It is a fishing "cannon" which launches a circular, weighted bob out of a barrel with compressed carbon dioxide. The line is attached to the bob and leads to the lure on one end, and back to the reel on the other, both located outside the barrel. Reeling could be performed manually, but we have considered an electric reeling system, to make the entire device mechanical. As such, we feel such a machine would be best marketed to paraplegics and quadriplegics, although ballistics and paintball enthusiasts would probably also be interested.

#### **-Materials:**

- Paintball gun or PVC barrel
- CO<sub>2</sub> cartridges or tank
- Paintball gun valving and trigger mechanism
- Weighted bob
- (optional) Electric motor for radio controlled car
- (optional) Speed controller for radio controlled car
- (optional) Battery pack

#### **-Pros:**

- Device would be based off of existing technology
- Device could be made from currently produced parts
- Device could achieve better distances than required, and would have the ability to be adjusted for different distances.

### **-Cons**

- Device could potentially be nothing more than a minor modification to existing designs
- Device would not allow for training of casting motion
- Device could potentially be dangerous for those with little or no experience with CO2 powered devices

## **Concept 2: Crank-back fishing catapult**

This concept is a device which uses the elastic properties of the fishing rod itself to launch the line in a manner similar to a catapult. The fishing rod would be held stationary at its base, while the top of the rod is pulled back and locked down by means of a winch, forming an arc shape. When a trigger is pulled, the rod is released, and the rod straightens itself out, acting as a spring to launch the lure, which pulls the line as it is fired. This device could be either ground mounted or hand held, and would use a traditional hand reel. The potential market would focus on the handicapped, although again ballistics enthusiasts would probably have an interest in the device.

### **-Materials**

- Fishing rod or similar thin rod with elastic properties
- Hand-cranked winch mechanism
- spring-release trigger mechanism
- Fishing reel
- Aluminum or steel for base, either handheld or positioned on the ground

### **-Pros**

- Device would be based off of existing technology.
- Device could be made from easily acquired and inexpensive materials
- Device could be hand held
- Device would provide adequate distance and could be fine tuned for specific distances

### **-Cons**

- Device does not allow for training of casting motion
- Device could potentially be dangerous if misused

## **Concept 3: Cast-training arm brace/training rod**

This concept is a device which would strap around the fisher's arm to assist in proper fishing technique. The device would have an elbow and a wrist joint, and would have springs, stops, or other methods to direct the arm along a path which is determined to be the proper casting motion. The fishing rod and reel used would both be standard fishing devices, and could be interchangeable, although they could also be integrated into the device if this proves beneficial. This device would have a wide market range, including handicapped and recovering fishers, as well as novice fishers who are trying to learn the proper technique.

**-Materials**

- Metal (aluminum or steel) and plastic for arm brace
- Velcro
- Springs
- Fishing rod and reel
- Pin joints

**-Pros**

- Device would teach proper casting motion
- Device would assist in adding distances to casts
- Device would be based off of existing technology

**-Cons**

- Device may injure arm if motion limiting mechanisms are too violent
- Device may be clunky
- Device may be expensive to create, and mass produce

#### **Concept 4: Cock-back fishing rod**

This concept is a fishing rod which would use a traditional, unassisted casting motion. However, a trigger on the rod would launch the lure and line with an added mechanical assist. One potential means of achieving this assist would be a spring-loaded launcher which would need to be cocked-back before firing, hence the name, but there are a variety of methods which could also be used, including smaller variations of the CO2

powered launcher and crank-back catapult listed above. This device would use a hand-crank reel, and would be hand held. The potential market includes the handicapped, fishers recovering from an injury, and novice fishers who are trying to learn the proper technique.

**-Materials**

- Fishing rod and reel
- Spring-loaded trigger mechanism
- Further materials based on the method of mechanical assist, and would be investigated further if the concept is chosen

**-Pros**

- Device would teach proper casting motion
- Device would assist in adding distances to casts
- Device would be based off of existing technology
- Device would appeal to a wide market

**-Cons**

- Device may be dangerous if misused
- Device forms no extra casting guide for fisher

## **Concept 5: Casting tunnel**

This concept is a guide which would direct the casting motion along a single plain, thereby eliminating any side-to-side variation in casts. This device could be either worn around the fisher or placed on the ground in front of the fisher. It would take the form of two plates which would be aligned in a "V" shape. These plates would direct the motion of the cast forward while limiting side-to-side motion. The potential market would be novice fishers who may not have the proper direction, and fishers with shaky arm movements, primarily those with Parkinson's disease.

**-Materials**

- Plywood or Plexiglas for guide
- PVC or metal for base or harness

**-Pros**

- Device would teach proper motion
- Device would be based off of existing technology

- Device would be inexpensive to construct.
- Cons**
  - Limited engineering in design of device
  - Device would be clunky
  - Device would not assist in additional distance

## **Concept 6: Casting motion robot**

This concept is a robotic arm which would simulate a casting motion, without actually casting. It would be used to determine the proper angles and release points for fishers of different sizes and strengths. It would therefore need adjustable arm lengths and a means of adjusting the amount of power. An entire structure and control means would need to be constructed. The device would have a limited potential market of fishers who want to see a demonstration and want to fine tune their technique.

### **-Materials**

- Steel or aluminum frame and arm construction
- Springs or pneumatics for arm movement
- Accelerometers
- Control system

### **-Pros**

- Device would appeal to veteran fishers

### **-Cons**

- Device does not actually cast or assist in motion
- Device would be expensive to construct
- Device would be unreasonable for the time frame of this project

## **Concept 7: Novice Fishing Pole**

In reviewing these ideas, it was determined that, while some of them would provide extra distance and others would give a model of a cast, none would be able to identify and diagnose problems in a novice cast. Because of that, we decided to instead work on a device which could read the accelerations of the fisher's arm and be able to track the

motion. This device would provide for more specialized assistance for each individual and their difficulties.

**-Materials**

- Fishing Pole
- Accelerometers
- Possible strain gages
- Computer with LabView

**-Pros**

- Device would appeal to all beginners
- Device would be simple to construct and use
- Device would allow for personalized diagnostics

**-Cons**

- Device does not physically cast or assist in motion

## **Experiments needed for design**

### ***Casting Experiment***

#### **Explanation of Experiment**

The purpose of this experiment is to determine average values found during the average fisherman's cast. Some of the things in which we were interested in measuring was the distance produced by a person casting, the amount of time taken to make a cast, the amount of time taken for the object to land and the change in angle of the individuals cast.

#### **Experiment**

##### **Distance created from cast**

The distance created from a cast was measured using the following method. First the MQP group created a set starting point. At this starting point a line was drawn in which the caster was not allowed to move beyond. The caster then made their cast. The opposite MQP partner then measured the distance between the caster and the fallen object. The fallen

object used in this experiment was a set amount of weight 30g (1oz). Each of the two MQP partners did a total of 5 casts each and these distances were measured. The results for the distances produced by the casters can be seen in the graph below. The blue line represents the casts made by MQP partner 1 and pink represents the casts made by MQP partner 2.

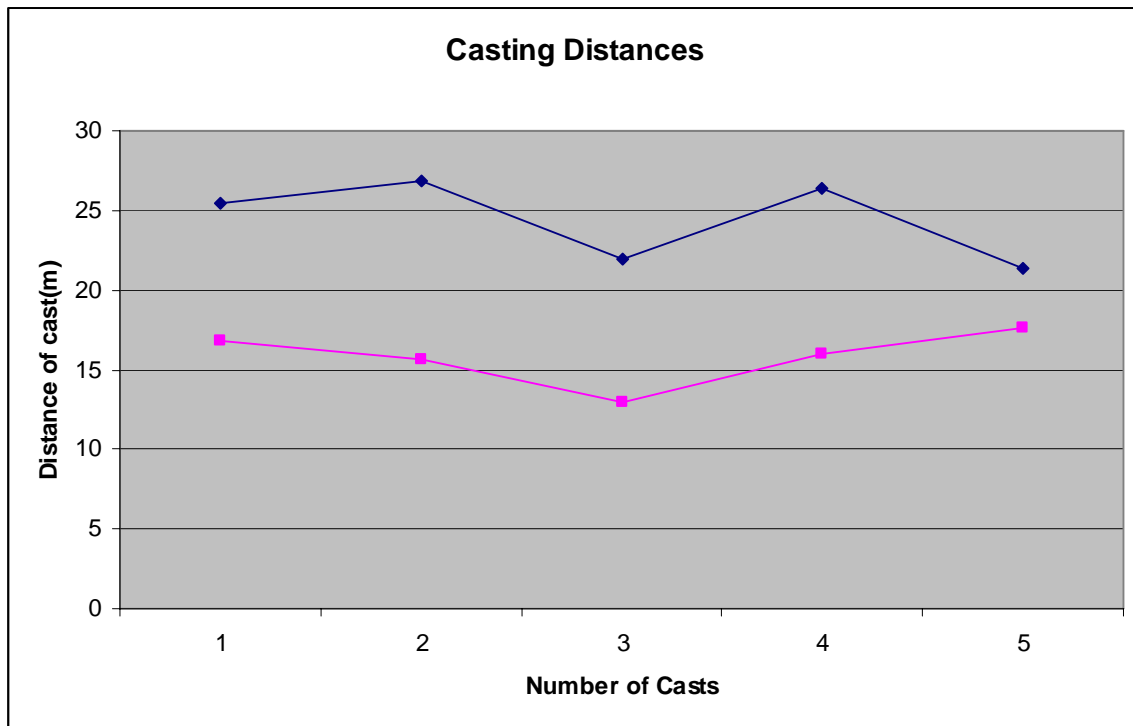


Figure 1: Chart of distance of casts from initial tests

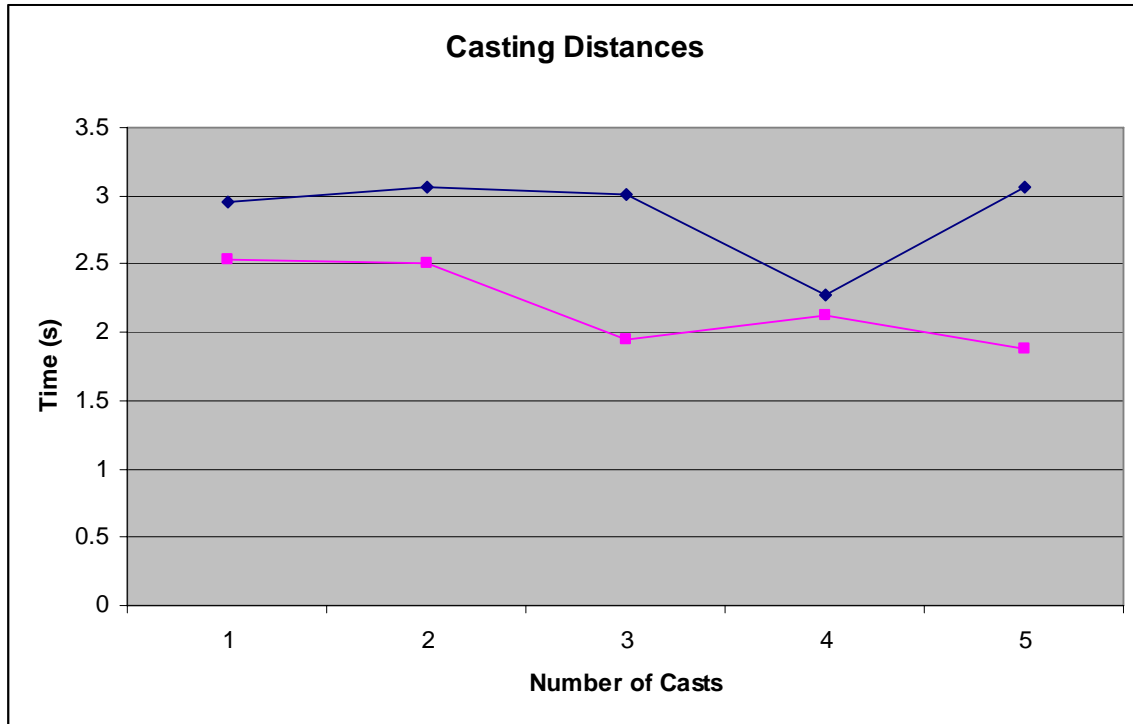
### Amount of time to make a cast

The amount of time to make a cast was found using a stop watch. This was done by starting the stop watch at the beginning of the cast and stopping the stop watch when the caster releases the line. It was found that the average time to make a cast was found to be .4 seconds.

### Amount of time for cast to land

The amount of time it took for the cast to land was measured using the same stop watch as used in cast timing. The stop watch was started at the point of release and stopped when the

object landed. All of these times recorded were taken off of the same casts used in the distance measurements. The times are represented in the graph below. As in the distance graph the blue line represents MQP partner 1 and pink represents MQP partner 2.



**Figure 2: Time of casts from initial tests**

### **The change in angle**

The change in angle was visually observed during the cast. The change of angle was found to be approximately 45 degrees.

### **Conclusion of Casting Experiment**

The results found from the casting experiment provided a rough idea of values seen during an average fisherman's cast. This allowed the MQP group to do calculations to find out velocities and accelerations produced during a cast. These were calculated using the basic stick equation calculations shown below.

## ***Constant Diameter Rod Equation Experiments***

The following calculations are based off of an experiment which was performed by the members of this MQP group. The values found are time taken to do the casting motion, time for a set 30 gram weight to go from release to landing and, and the distance in which this weight went.

Givens:

$$\theta := \frac{\pi}{4} \quad \text{rad}$$

$$L1 := 5.8 \quad \text{ft}$$

We approximated that the angle covered by the fishing pole was approximately 90 degrees or  $\pi/4$  radians. L1 represents the length of the fishing pole from the tip to the pivot point which was just above the fishers hands.

$$t1 := .28 \quad \text{sec}$$

$$t2 := 3 \quad \text{sec}$$

$$t3 := .45 \quad \text{sec}$$

t1 represents the amount of time it took for person one to travel the 90 degrees ( $\pi/4$  radians). t3 is the amount of time it took person two to travel the 90 degrees ( $\pi/4$  radians). t2 represents the average amount of time it took from release to landing of the weight.

Solving:

The left hand column represents calculations performed on person 1 of the MQP group. The right hand column represents the calculations performed on person 2 of the MQP group.

Angular acceleration is represented by  $\alpha$ . First the angular acceleration was found using the governing equation that the angle covered is equal to 1/2 angular acceleration multiplied by the time squared. After some manipulation of the equation we obtain:

**Person1**

$$\alpha := \frac{2 \cdot \theta}{t1^2}$$

$$\alpha = 20.036 \frac{\text{rad}}{\text{sec}^2}$$

**Person2**

$$\alpha1 := \frac{2 \cdot \theta}{t3^2}$$

$$\alpha1 = 7.757 \frac{\text{rad}}{\text{sec}^2}$$

From the angular acceleration the angular velocity can be found. The angular velocity is represented by  $\omega$ . This was found from the governing equation which states that angular acceleration is equal to the change in angular velocity divided by the change in time. After some algebraic manipulation  $\omega$  is obtained.

$$\omega := \alpha \cdot t1$$

$$\omega = 5.61 \frac{\text{rad}}{\text{sec}}$$

$$\omega1 := \alpha1 \cdot t3$$

$$\omega1 = 3.491 \frac{\text{rad}}{\text{sec}}$$

From the angular velocity the linear velocity at the top of the fishing pole could be found. The linear velocity is represented by  $v$ . This was found from the governing equation which states that linear velocity is equal to the angular velocity multiplied by the radius under which it is being angled.

$$v1 := \omega \cdot L1$$

$$v2 := \omega1 \cdot L1$$

$$v1 = 32.538 \frac{\text{feet}}{\text{s}}$$

$$v2 = 20.246 \frac{\text{feet}}{\text{s}}$$

The velocity found is in feet per second therefore it will be converted to meters per second using the conversion 1 foot = .3048 meters

$$V1m := v1 \cdot .3$$

$$V1m = 9.761 \frac{m}{s}$$

$$V2m := v2 \cdot .3$$

$$V2m = 6.074 \frac{m}{s}$$

This shows that the linear velocity off the top of the fishing pole of the weight. From this the distance obtained by the weight can be found.

We used the assumption that the deceleration caused by drag of the line and air resistance is ignored and the approximate flight length is 3 seconds. The distance traveled in the x direction is simply the linear velocity multiplied by the time it takes to land.

$$x := V1m t^2 \cdot \cos(\theta)$$

$$x = 20.707 \text{ m}$$

$$x1 := V2m t^2 \cdot \cos(\theta)$$

$$x1 = 12.884 \text{ m}$$

The previous equations used a flight time based off of experimental results. The following results are based of calculations not results found in the previous experiment.

The maximum trajectory height was found. This was done using the governing equation that max height reached would be equal to 1/2 acceleration (gravity) \* time<sup>2</sup> + initial velocity \* time + initial height. First the time would have to be calculated. This was done using the governing equation acceleration is the change in velocity over time. Vm is the linear velocity the velocity in both x and y directions. The y velocity was found by multiplying by the sin of the angle. Estimated angle was  $\pi/5$  radians.

$$V1m = 9.761 \frac{m}{s}$$

$$V2m = 6.074 \frac{m}{s}$$

$$\delta := \frac{\pi}{4}$$

The x and y components of velocity are shown below.

$$V1x := V1m \cos(\delta)$$

$$V2x := V2m \cos(\delta)$$

$$V1x = 6.902 \frac{m}{s}$$

$$V2x = 4.295 \frac{m}{s}$$

$$V1y := V1m \sin(\delta)$$

$$V2y := V2m \sin(\delta)$$

$$V1y = 6.902 \frac{m}{s}$$

$$V2y = 4.295 \frac{m}{s}$$

Acceleration is equal to the change in velocity over time.

gravity

$$a := 9.8 \frac{m}{s^2}$$

The time to reach max height was calculated. time at max height is represented by tup.

$$tup := \frac{V1y}{a}$$

$$tup2 := \frac{V2y}{a}$$

$$tup = 0.704 \text{ s}$$

$$tup2 = 0.438 \text{ s}$$

max height = 1/2 acceleration (gravity) \* time<sup>2</sup> + initial velocity \* time + initial height

$$y := \left(\frac{-1}{2}\right)(a)(tup^2) + V1y \cdot tup + 3.5$$

$$y2 := \left(\frac{-1}{2}\right)(a)(tup2^2) + V2y \cdot tup2 + 3.5$$

$$y = 5.931 \text{ meters from ground}$$

$$y2 = 4.441 \text{ meters from ground}$$

Next the time it took to travel from the max height to the ground was calculated.

This was found by manipulating the same governing equation max height = 1/2 acceleration

(gravity) \* time^2 + initial velocity \* time + initial height. The time taken to reach the ground is represented by tdown.

$$t_{down} := \sqrt{\frac{2 \cdot y}{a}}$$

$$t_{down} = 1.1 \text{ s}$$

$$t_{down2} := \sqrt{\frac{2 \cdot y2}{a}}$$

$$t_{down2} = 0.952 \text{ s}$$

From this the total time was calculated by adding together the time to reach peak with the time to come down from peak height.

$$t_{total} := t_{up} + t_{down}$$

$$t_{total} = 1.804 \text{ sec}$$

$$t_{total2} := t_{up2} + t_{down2}$$

$$t_{total2} = 1.39 \text{ sec}$$

From the total time and found the distance could be calculated using the governing equation velocity = change in distance over time. The total distance traveled is represented by xcalc.

$$x_{calc} := V1 \times t_{total}$$

$$x_{calc} = 12.455 \text{ m}$$

$$x_{calc2} := V2 \times t_{total2}$$

$$x_{calc2} = 5.971 \text{ m}$$

These tests determined that the time spent in the air was almost twice that calculated with a commensurate lineal span greater than that predicted.

attempts	Person 1		Person 2	
	distance(m)	time(s)	distance(m)	time(s)
0	23.35	2.95	16.8	2.53
1	15.45	3.07	15.6	2.51
2	22.1	3.01	12.9	1.95
3	24.65	2.28	16	2.12
4	0	0	17.5	1.88
5				

This experiment was done with a constant weight, same measuring instruments and on the same day. From the calculations performed above force calculations we were then able to calculate an average motor power that would be needed. These calculations can be seen below. We performed these calculations as we were still unsure of which design we would be going forward with. The motor power was not needed for the design which we ultimately chose.

## ***Force Equation experiment***

### **Force Calculations**

Givens:

Mass of the weights

$$m := .03 \quad \text{kg}$$

Radius of fishing pole from tip to reel

$$r := 1.9812 \quad \text{m}$$

To find the torque, the moment of inertia was calculated from the above values, where moment of inertia "I" equals mass times the square of the radius. This result was then multiplied by the rotational acceleration found for Person 1's casts for the final value of torque.

$$I := m \cdot r^2$$

$$I = 0.118 \quad \text{Kg} \cdot \text{m}^2$$

$$\tau := I \cdot \alpha$$

$$\tau = 2.359 \quad \text{N} \cdot \text{m}$$

Force is calculated by the torque divided by the radius times the sine of the traversed angle.

$$F := \frac{\tau}{r \cdot \sin(\theta)}$$

$$F = 1.684 \text{ N}$$

We made the assumption that work would be found by multiplying the above force by the length of the arc of the cast. The arc is calculated as the radius of the arm multiplied by the angle.

$$W := F \cdot (r \cdot \theta)$$

$$W = 2.621 \text{ Joules}$$

Power is then defined as work divided by the time the work is applied, which in this case would be the time taken for the casting motion.

$$P := \frac{W}{t1}$$

$$P = 9.359 \text{ Watts}$$

For comparison purposes, we also calculated the power in mechanical horsepower, which is approximately 1 HP per 746 W.

$$HP := \frac{P}{746}$$

$$HP = 0.013 \text{ HP}$$

$$V_{\text{initialcalc}} := \frac{V1m}{\cos(\theta)}$$

$$V_{\text{initialcalc}} = 13.805$$

$$V_{\text{initialcalcy}} := V_{\text{initialcalc}} \sin(\delta)$$

$$V_{\text{initialcalcy}} = 9.761$$

$$t_{\text{upnew}} := \frac{V_{\text{initialcalcy}}}{a}$$

$$t_{upnew} = 0.996$$

$$y_{new} := \left(\frac{-1}{2}\right)(a)(t_{upnew}^2) + V_{initial}c_{y-tupnew} + 3.5$$

$$y_{new} = 8.361$$

$$t_{downnew} := \sqrt{\frac{2 \cdot y_{new}}{a}}$$

$$t_{downnew} = 1.306$$

$$t_{new} := t_{upnew} + t_{downnew}$$

$$t_{new} = 2.302$$

$$x_{new} := V_{1m}t_{new}$$

$$x_{new} = 22.474$$

$$diffx := x_{new} - x_1$$

$$diffx = 9.59$$

## ***Pole Measurement Experiment***

The preliminary tests used a typical, novice spin caster fishing pole. The measurements of the fishing pole can be seen in the graph below. This graph represents the relationship between the diameter of the pole with the length of the pole. The equation for the relationship between the length of the pole and the diameter of the pole is written in the corner of the graph. The length of the pole is in feet and the diameter of the pole is in inches.

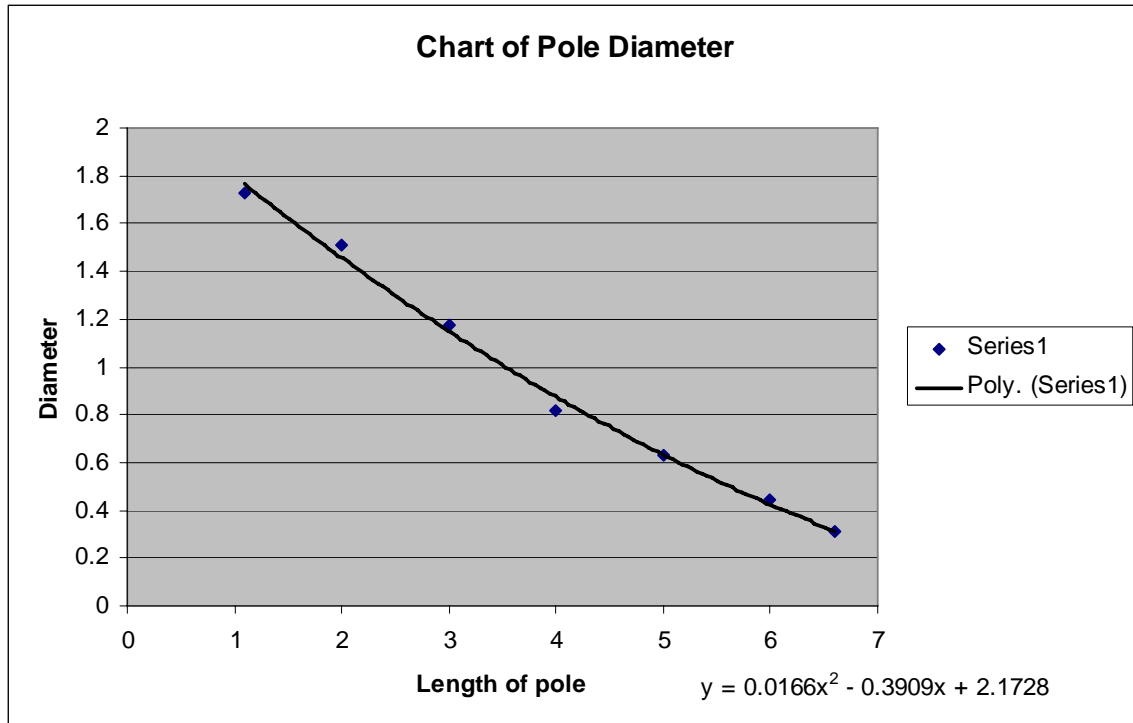


Figure 3: Pole diameter to length

## Strain Gage Experiment

### Proposed Strain Gage Configuration

In the proposed configuration for the strain gages on the pole, the positive wire of the strain gage goes to the 5V source as well but the negative wire is connected to a dummy gage. In our case the dummy gage is a resistor that provides equal resistance that another strain gage would (120 Ohms). The remaining wire which is connected to the negative terminal of the strain gage goes to the positive channel output. The difference in this wiring from the wiring of accelerometer is that the strain gage is part of a bridge configuration. The bridge that is utilized in this configuration is called a quarter bridge. This quarter bridge configuration can be seen below. R1 and R2 are resistors of equal resistance. The wires that point toward the V in the middle are the channel outputs. The channel output on the left is negative and the channel output on the right is positive. The bridge was created

on a bread board. As there are 4 strain gages being used on the fishing pole there are 4 separate bridges created for each of the channels. The MQP group ran an experiment that proves that each of the strain gages requires its own separate bridge. As in the case of the accelerometers the strain gages are able to share a single 5V source and a common ground. There for the bread board has separate terminals which provide only positive voltage and only grounds.

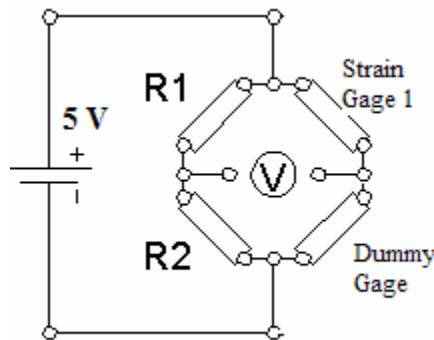


Figure 4: Strain gage schematic

## Strain Gage Placement

Initially, we planned to use strain gages to track the flex of the pole during the cast. Experiments were performed to determine the feasibility of this idea. For the first experiment, the MQP group took measurement of the angles under which the strain gages would cover on the pole at the different diameters of the pole. This was calculated using the equation  $\text{Angle} = (.15 / (\text{Thickness} / 2)) * (180 / \text{Pi})$ , where .15 is the distance between the positive and negative terminals on the strain gage. The graph below shows the angle of coverage of the strain gage to the pole length.

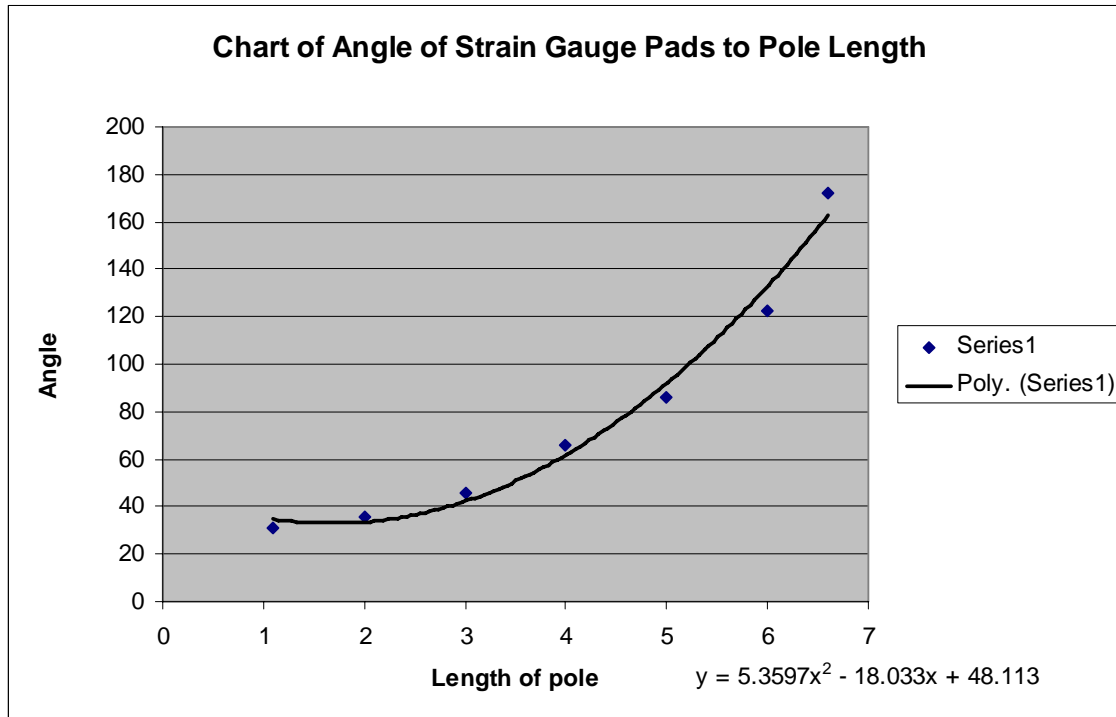


Figure 5: Acceptable angle of strain gages to the length of the pole

## Explanation of Experiment

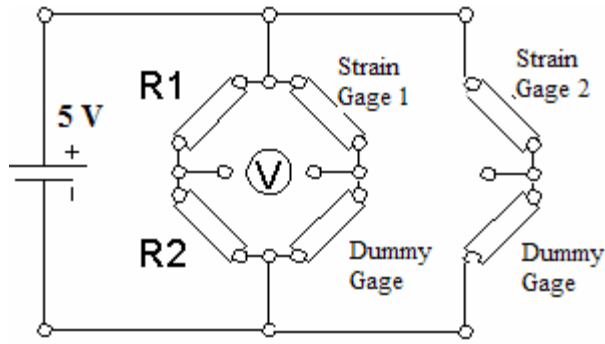
Two strain gages were wired up to one bridge and measurements were taken to see if this configuration had any impact on the results received. The purpose of this experiment was to allow us to use one bridge that would power all the strain gages necessary for our fishing pole design.

## Hypothesis

We hypothesized that both strain gages could be hooked up to the same quarter bridge setup and still function independently of one another.

## Experiment

We first wired the strain gages as shown below:



**Figure 6: Multiple strain gage experiment schematic**

In this diagram R1 and R2 are both of equal resistance. For our experiment we used 100 Ohm Resistors. The dummy gages as well were represented by 120 Ohm resistors. 120 Ohms was chosen because the strain gages hold a 120 Ohm resistance respectively. This bridge configuration was supplied a constant voltage of 5 Volts throughout the experiment and that was confirmed using a volt meter.

In order to test that the strain gages were able to work independently of one another we conducted several experiments. We started first by setting Control Conditions. For this we allowed LabView to take readings with both strain gage hooked up but under no strain. The results of these readings can be seen in graph 1 (control conditions). Blue is strain gage 1 readings and Pink is strain gage 2 readings.

Second readings were taken for both strain gages while just one of each of the strain gages experienced a set amount of strain. This was done by placing a set weight on the end of the cantilever beam under which each strain gage was attached. The results for this experiment can be seen in both graph 2 and 3(test 2 channel 0 and test 2 channel 1).

Thirdly, readings were taken from both the strain gages while each strain gage experienced changing strains individually. The results for this can be seen in graphs 4 and 5(isolated vibrations channel 0 and 1).

Lastly, readings were taken from both strain gages which both were undergoing strain at the same time. These readings were both taken in, in phase and out of phase motion. The results for this can be seen in graphs 6 and 7(in phase and out of phase motion).

## **Results**

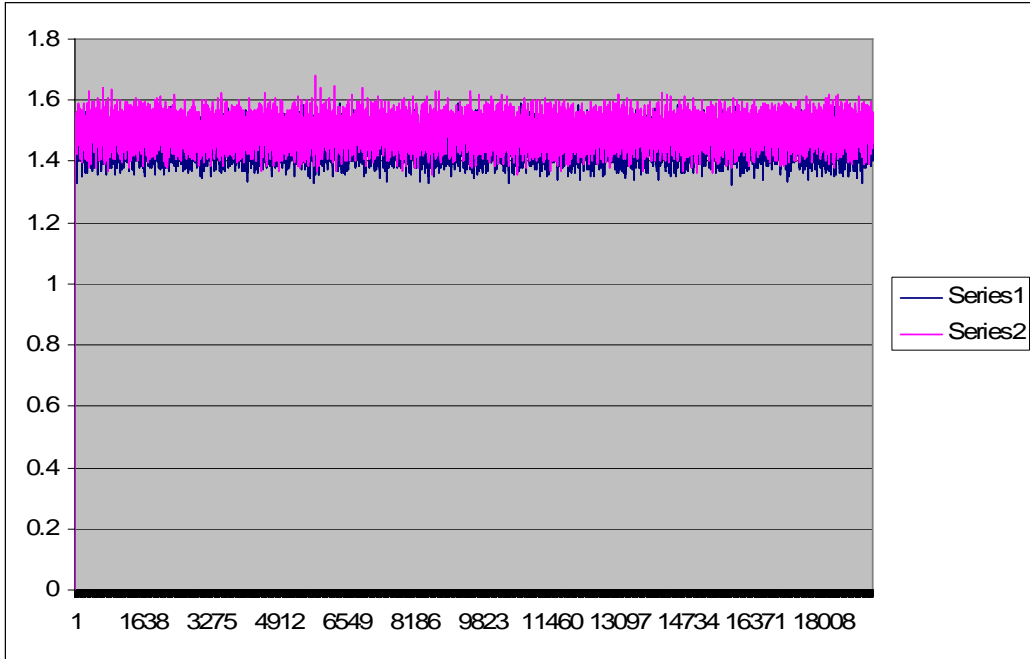
Readings taken from the control portion of the experiment had shown that both strain gages were able to take readings while hooked up in this configuration. The second experiment under which each strain gage experienced a set portion of constant strain did not provide adequate variance to make any conclusions. The third test however under which the strain gages experienced a change in strain individually provided the best results. From the third test it was shown that the strain given to the second strain gage impacts the first strain gage readings drastically. Although the first strain gage is able to take readings independently of the second strain gage. The In phase out of phase experiment showed us that the strain gages are able to take readings in sink but the first strain gage is still impacted by the second strain gage.

## **Conclusion**

The strain gages will not be able to work off of the same bridge for fishing pole configuration. Each strain gage will need it own bridge to act independently of one another.

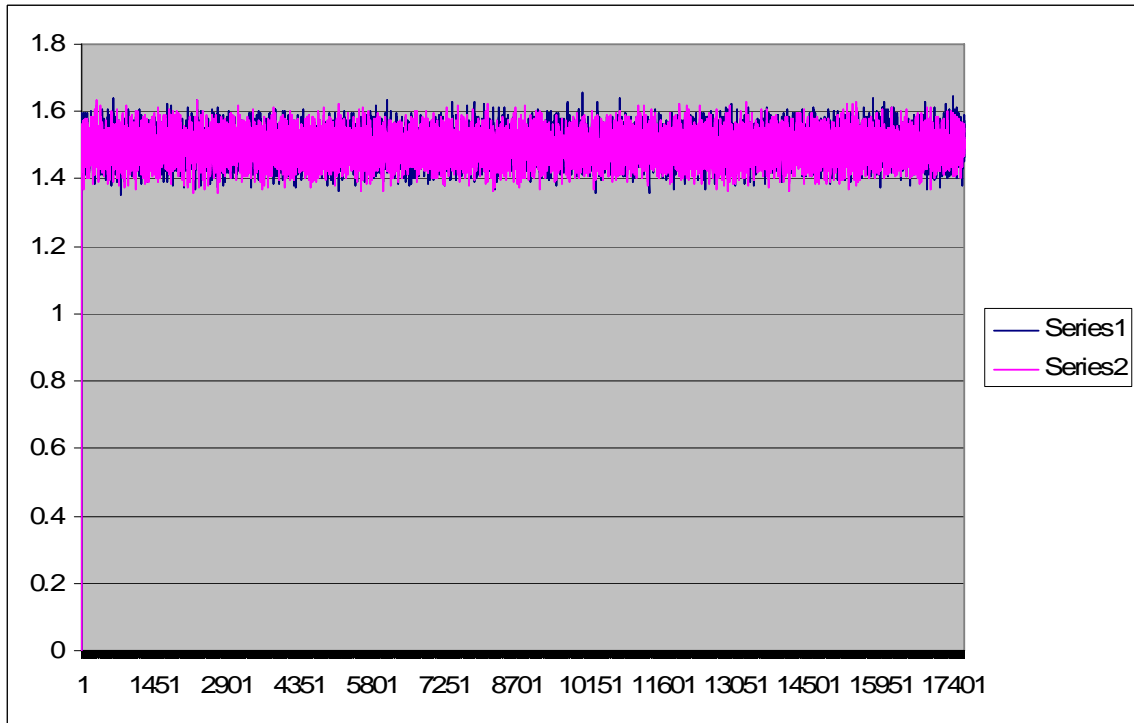
## **Graphs:**

Series 1 (blue) represents strain gage 1, Series 2(pink) represents strain gage 2



**Figure 7: Control**

Graph 2 (Test 2 channel 0, strain gage 1)



**Figure 8: Constant strain applied to strain gage 1**

Graph 3 (test 2 channel 1, strain gage2)

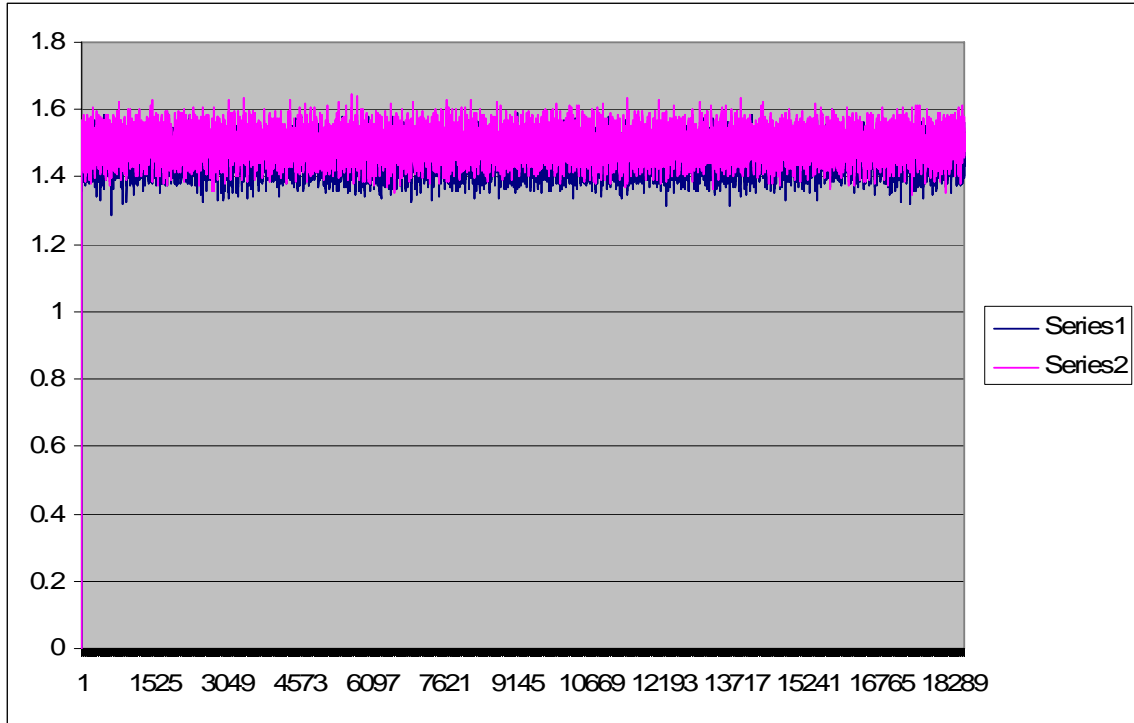


Figure 9: Constant strain applied to strain gage 2

Graph 4 (Isolated Vibration channel 0, strain gage 1)

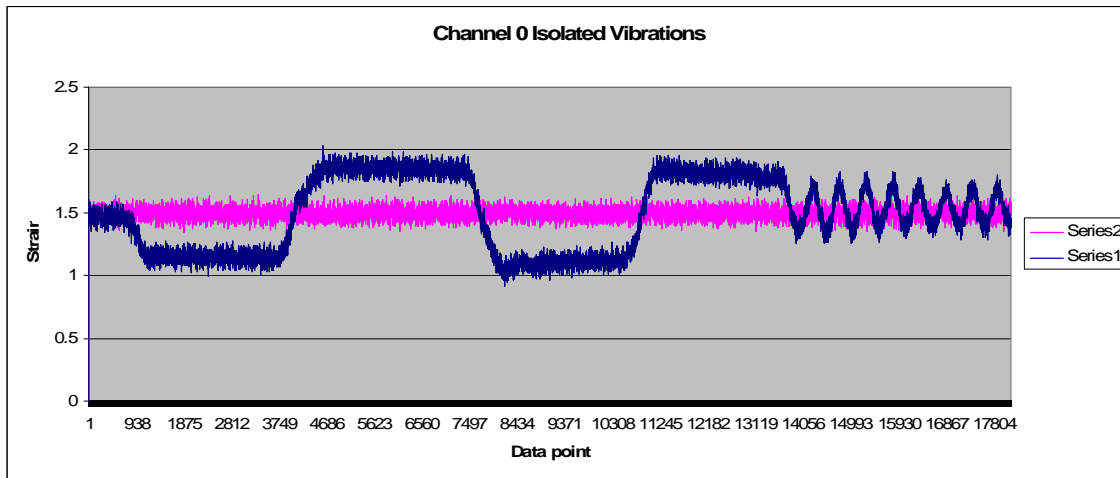


Figure 10: Changing strain applied to strain gage 1

Graph 5 (Isolated Vibration channel 1, strain gage 2)

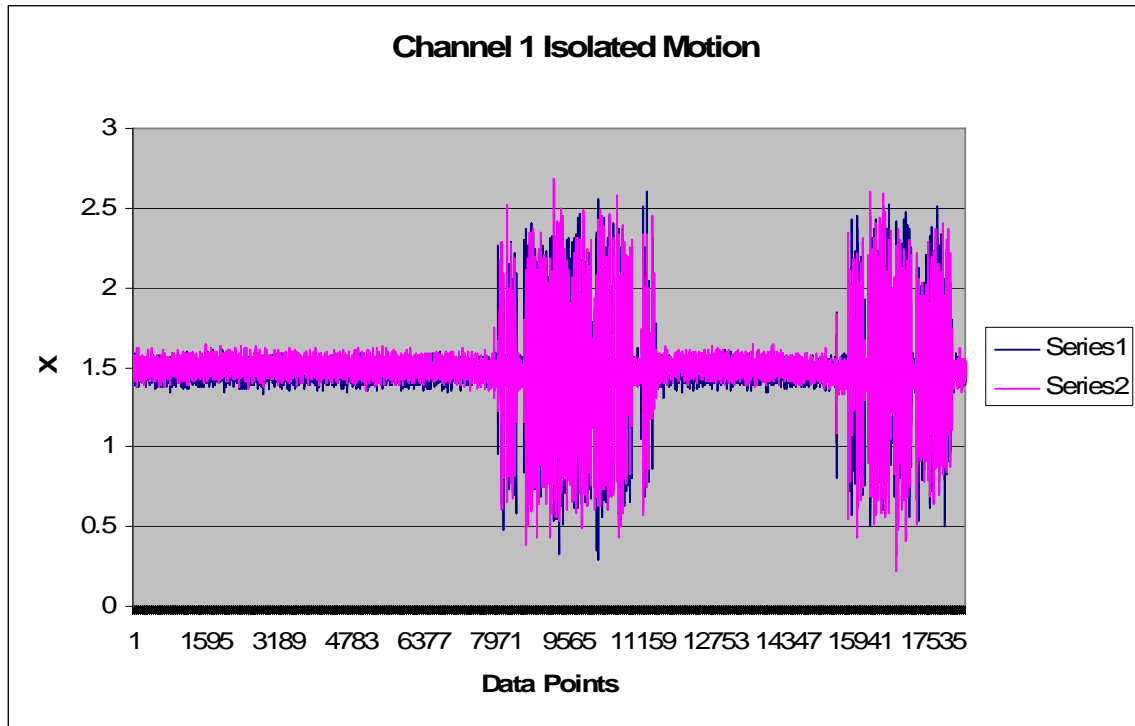


Figure 11: Changing strain applied to strain gage 2

Graph 6(In Phase Motion)

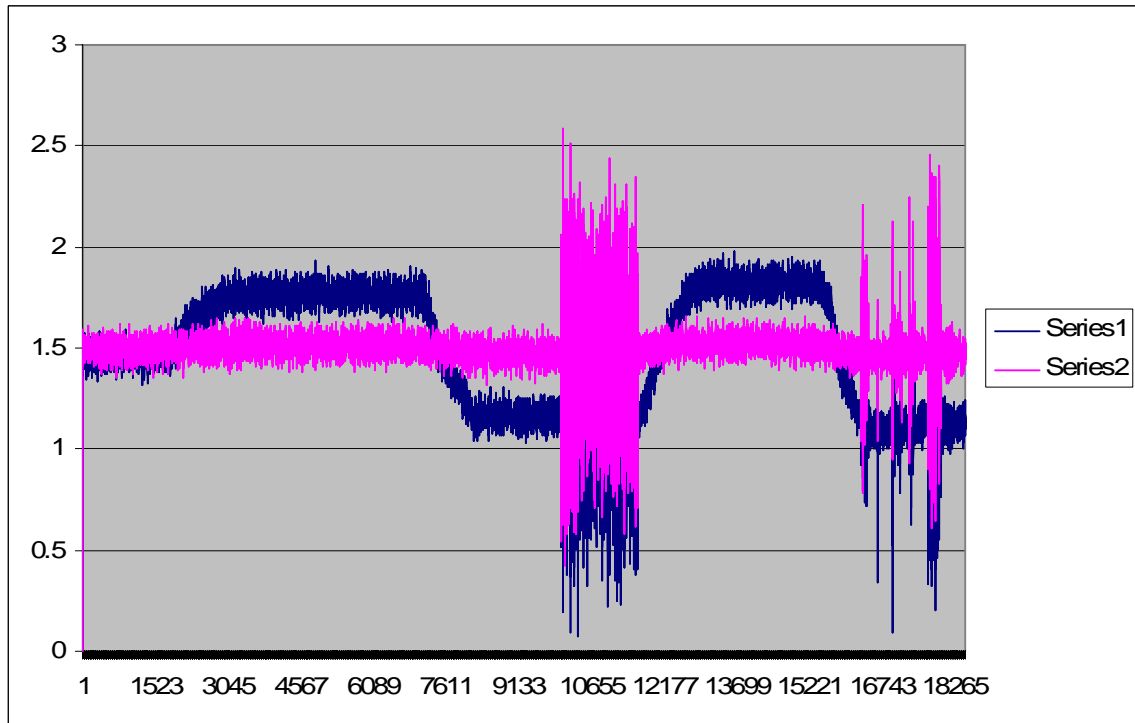


Figure 12: Changing strain applied to both strain gages (In phase motion)

Graph 7 (Out of Phase Motion)

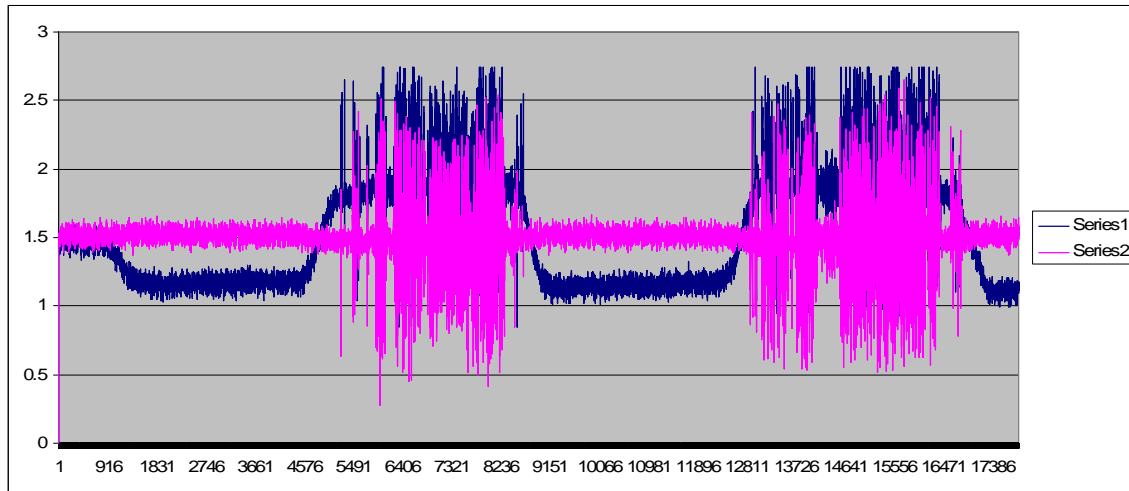


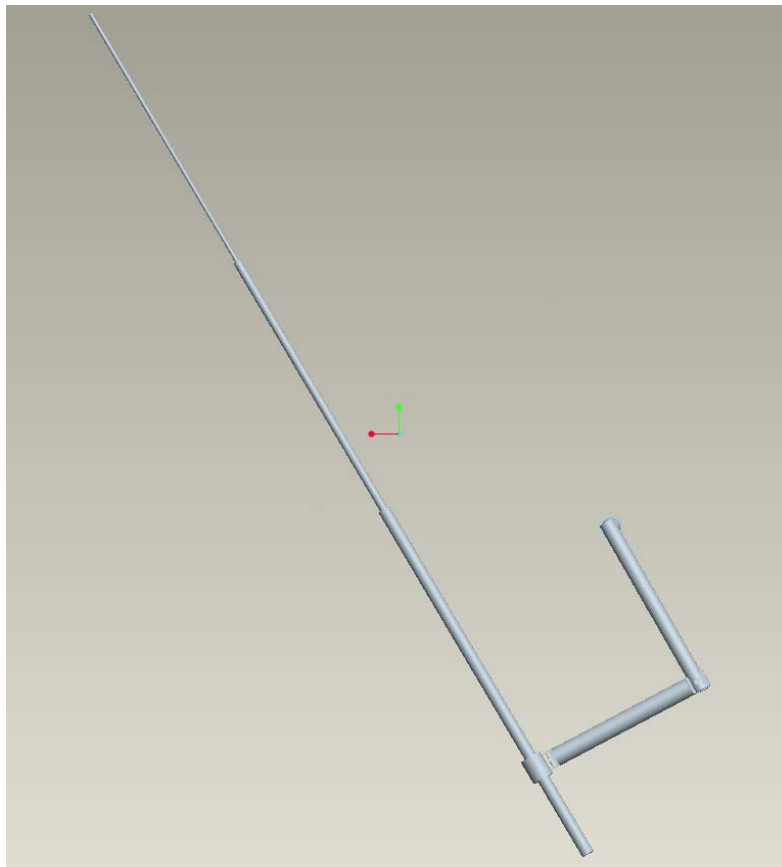
Figure 13: Changing strain applied to both strain gages (Out of phase motion)

After finding the issues with the strain gages we decided to focus on the accelerometer equations that we would need to use to track the accelerations due solely to the joints in which we are interested. We started this with first doing the equations for the first two portions of our arm. We first did the equations out in two dimensional space and then expanded them to three dimensional space. We first decided to see if using our stick and force calculations if we could reproduce the motion of a cast in Pro E using individual motors for joints and estimates of power produced by each of those joints.

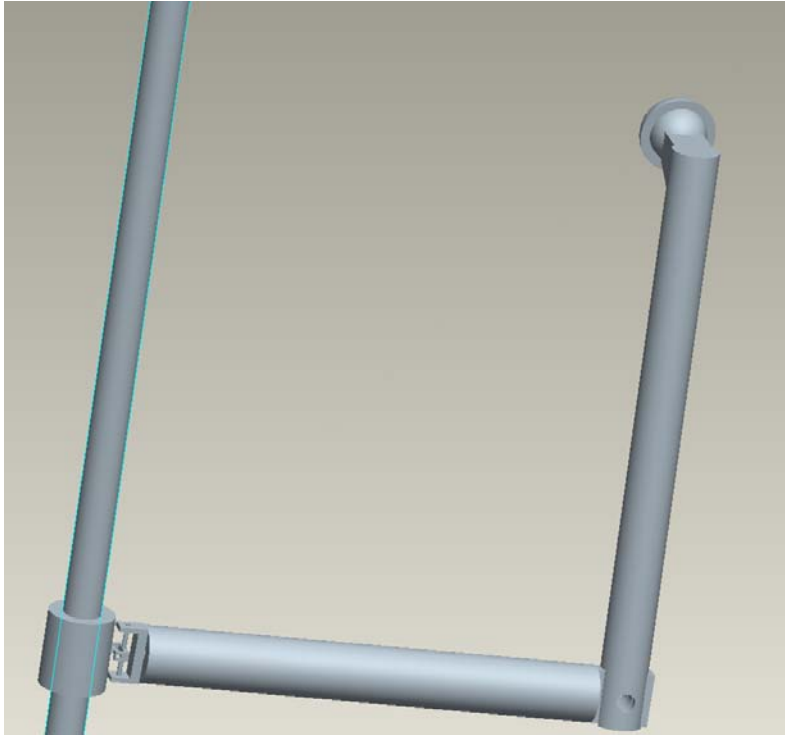
### ***Computer Model***

In order to get a better idea of the kinematics involved in a cast, we modeled a human arm and fishing pole in ProEngineer Wildfire 2.0. The model is a simplified version of an actual arm, with dimensions accurate to that of an actual human, as they were

measured by us before building the model. The upper arm is connected to ground with a ball-and-socket joint, the elbow is a pin joint, and the wrist is a series of 3 pin joints to achieve a realistic and constrainable movement. Each joint is constrained within a certain range of angles matching that of an actual human's arm. For simplicity, the fishing pole is modeled as a solid link, although its shape tapers according to our actual pole.



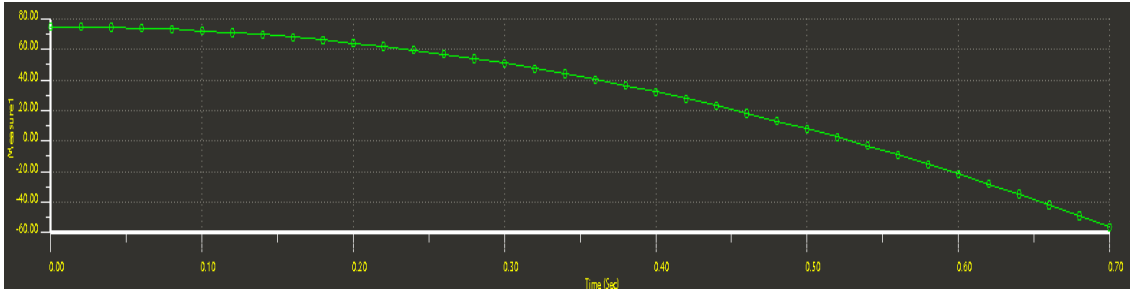
**Figure 14: Arm/Pole ProEngineer assembly**



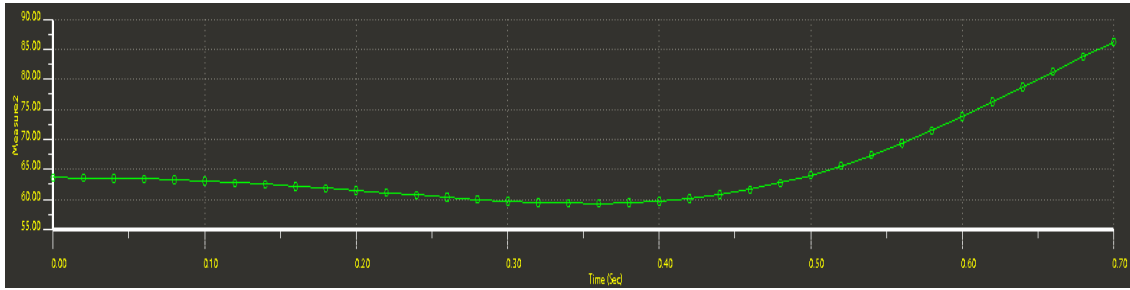
**Figure 15: Detail of arm within assembly**

Using the Mechanism program within Wildfire, we attached stepper motors to each joint. The given motion was determined by the approximate time we calculated an actual cast to occur in based on our initial experiments. Also, initial and final positions of the arm segments were approximated based on observation of the casts. This then allowed us to find the distance from ground, velocity, and acceleration of the end points of each segment of the arm. At the time which we performed this analysis, we were focusing on the endpoint of the arm, trying to grasp what magnitudes of velocity and acceleration we were getting.

For the position of the endpoint of the arm, we made two graphs. The first is the angle from horizontal, and the second is the linear distance from ground. A pole will go through approximately a 110 degree rotation throughout its cast. The linear distance shrinks for a time as the arm simultaneously pulls the tip of the pole down and across by the elbow, but grows with the follow-through.

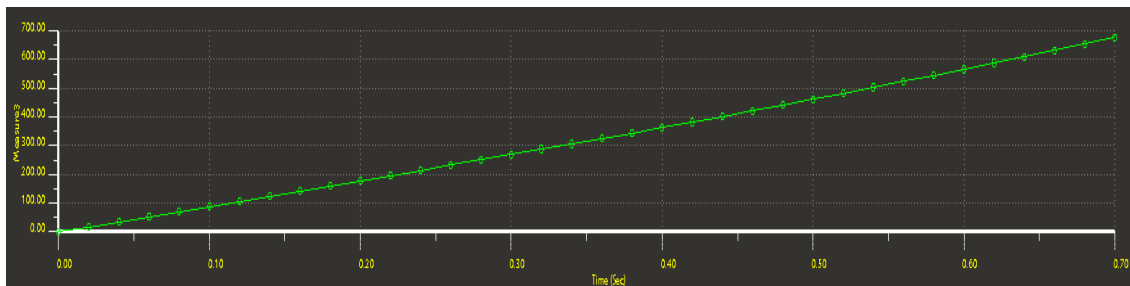


**Figure 16: Rotation from horizontal of the CAD Model**



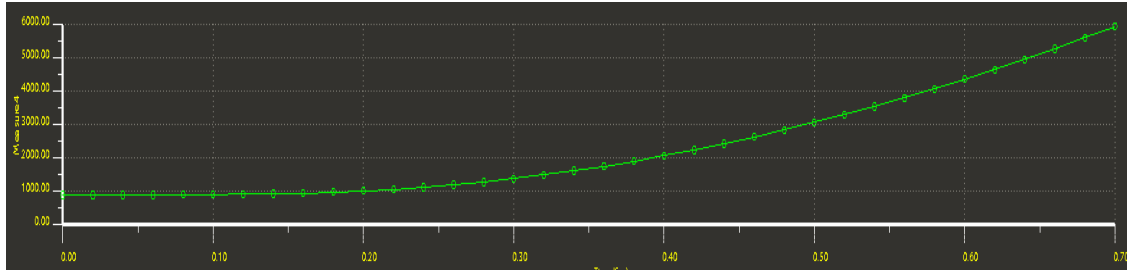
**Figure 17: Distance from ground of the pole tip of the CAD Model**

We were surprised to see that the velocity of the tip of the pole increased linearly within the model. It rose to approximately 17.78 m/s over the period of the cast. This velocity is only 39.77 miles/hour, which is reasonable considering velocities of balls thrown by professional athletes can range well above this.



**Figure 18: Velocity of pole tip in CAD model**

Acceleration of the tip rose in a quadratic fashion to a maximum acceleration of 152.4 m/s<sup>2</sup>. We felt this was reasonable as the acceleration is the sum of all accelerations before it, where it combines that of the upper arm, forearm, and wrist. The quadratic shape showed us that accelerations would rise gradually throughout the cast, giving an idea of what our later accelerometer readings would look like.



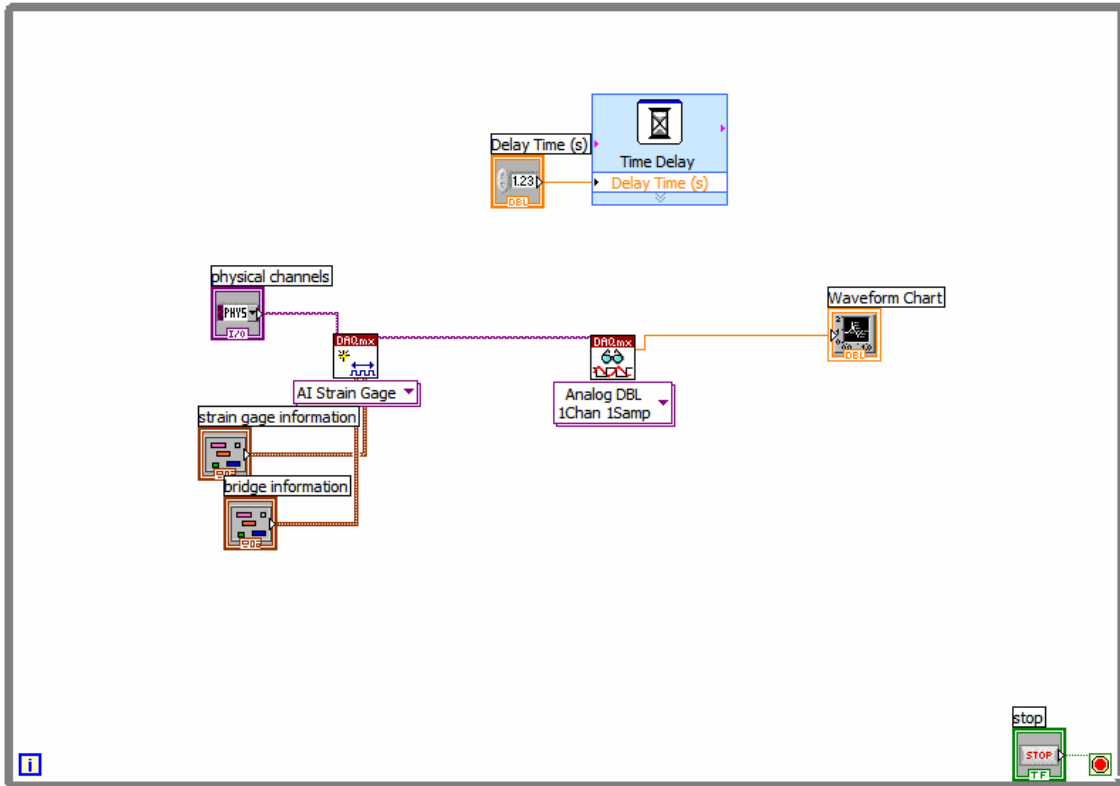
**Figure 19: Acceleration of pole tip in CAD model**

After the simulations were created the design of the program which would be hooked up to the fishing pole could be created. This was done in LabView, a program which is able to take measurements taken from accelerometers and strain gages and write them to a usable excel spreadsheet file. Within this program a VI is created which allows for the separation of the data received from the different measuring devices.

### ***LabView Experiment***

Measurements of pole movement are made with a lab console created in LabView 8. The console, called a VI, evolved over several iterations into its final form. These changes were spurred by modifications to our procedure, and our own expanding knowledge of the program.

Initially, the VI was developed as a simple one channel, single chart program to test the proper setups of both the accelerometers and the strain gauges. With this VI, the channel was inputted into the VI directly, as well as strain gauge excitation voltage, gage factor, initial voltage, and limits. These were placed as options of the channel input block, which was configured for the specific type of measurement. That block was then wired to the Data Acquisition (DAQ) Read block, and then into the waveform chart. This allowed us to refresh our knowledge of taking measurements with LabView.

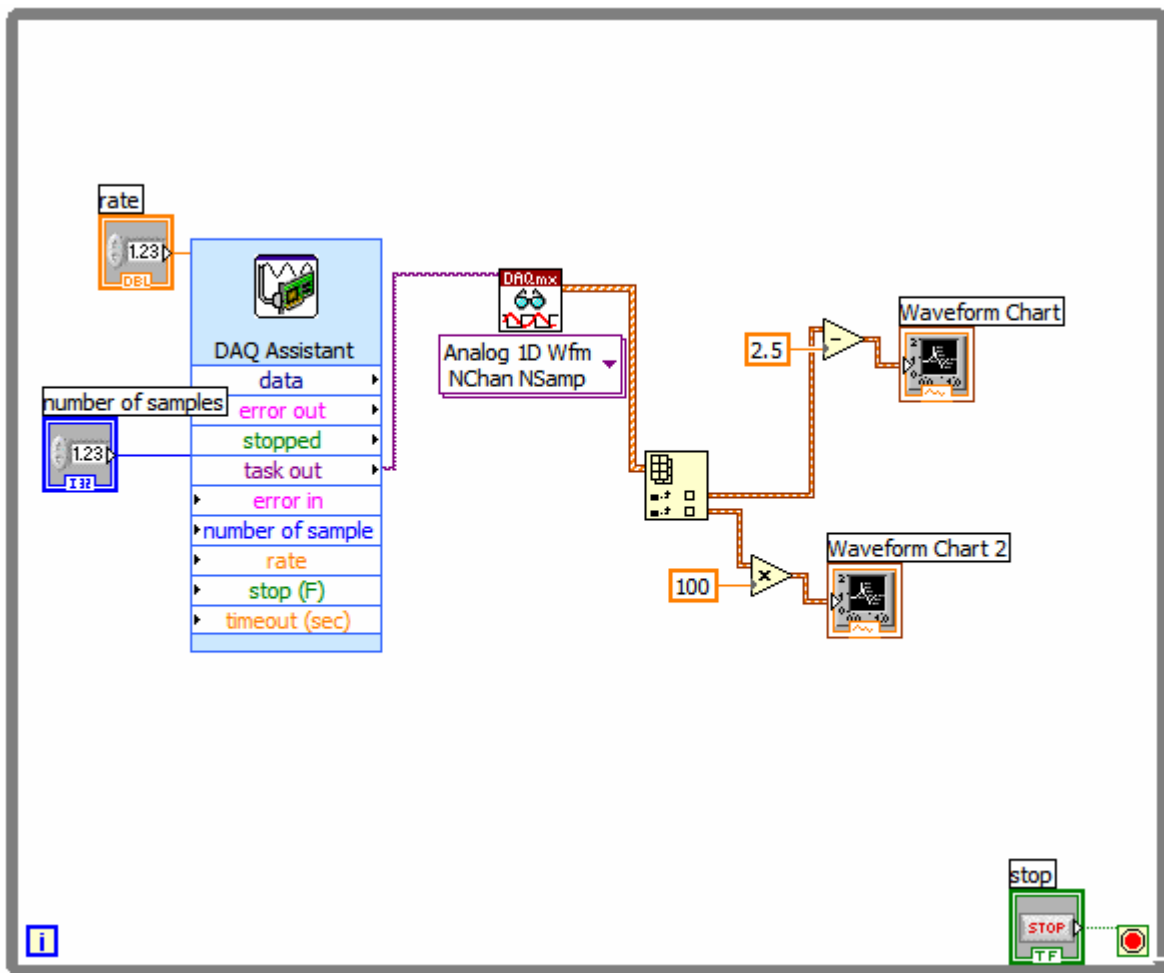


**Figure 20: Initial LabView block diagram**

The next step was to take multiple measurements simultaneously on the same DAQ device. Initially, this was done by simply having a separate but identical sequence of blocks within the while loop, with changes made to specify the measurement to either strain or acceleration. If the two measurements were of the same type, this setup worked well and provided little interference and high read rates. However, when taking measurements of different types, the read rate was less than one read per second. With a cast being performed in less than half a second, this was not acceptable.

After struggling with the basic block diagram to try to get multiple channels to read smoothly, a visit to the National Instruments website revealed that such capability required the use of the “DAQ Assistant”, a sub-VI which operates within the main VI. With this component, the strain gauge and accelerometers properties can be set externally from the block diagram. The component also contains the proper coding to allow for many

measurement types with only one data acquisition device, as well as manually setting the read rates. This setup is wired out to a DAQ Read block, which is in turn wired to an Index Array block to separate data into individual readings, which are each wired into the waveform charts. There are some limitations with this setup, however. The DAQ Assistant block can only be configured within LabView and within the block diagram, which poses a problem for us as our goal was to make a self-contained program. As such, another solution was needed.



**Figure 21: DAQ Assistant block diagram**

That solution came in the form of using the Measurement and Automation Explorer (MAX) to set up virtual task channels to handle the measurements. MAX works

identically to the DAQ Assistant, but can be setup independently of LabView, allowing us to create a single executable file for the VI. MAX also gives a wiring diagram based on the specified device. This was an invaluable feature for us as we struggled to find a setup which would allow for all of the measurements we wanted to take simultaneously. The virtual channel task is saved, and entered into a task/channel control block in LabView, which is then wired to the DAQ Read block, which in turn goes into the Index Array and waveform charts. Data is also sent to a Formula Node for processing into readable accelerations, velocities, and positions. Such features will be placed into sub-VIs to keep clutter from the primary VI and to organize data better.

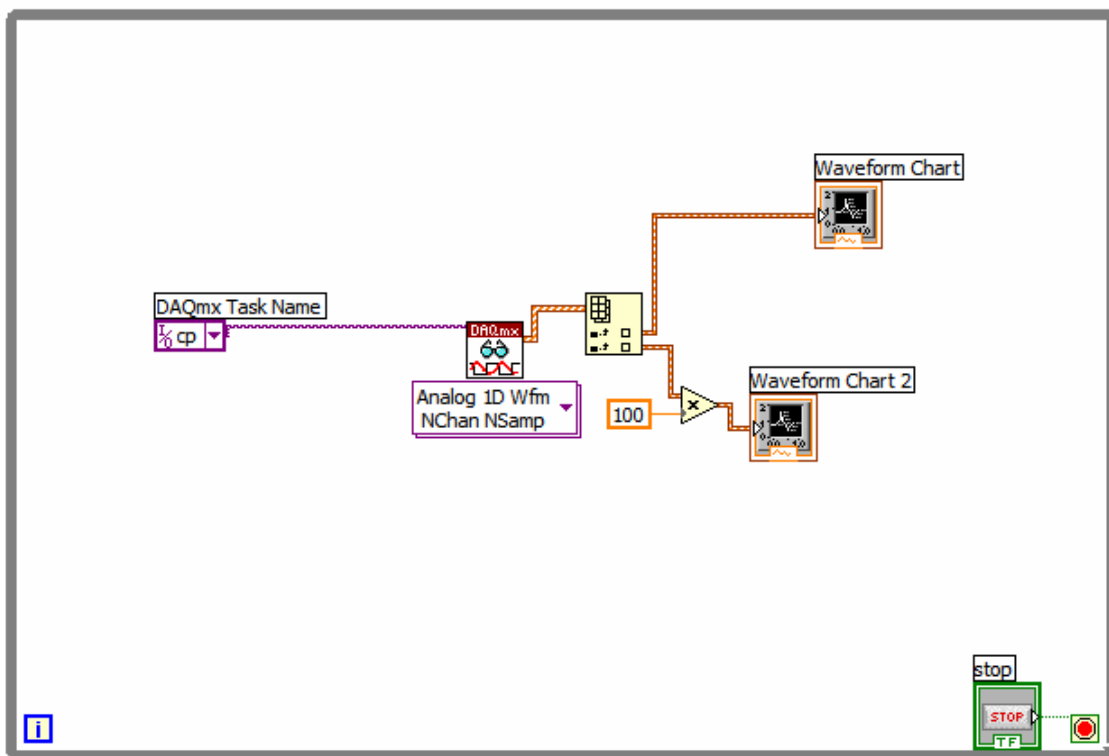


Figure 22: Simple block diagram for MAX created channels

## Single Accelerometer Experiment

During the creation of our final product we experienced an issue with our accelerometers picking up a large quantity of noise. To better determine the source of the changes in frequency we decided to assemble a single accelerometer to observe if the problems being seen were due to the wiring of the accelerometers or an outside source which the accelerometers are able to detect.

## **Experiment**

The single accelerometer was wired in the same configuration as on the pole and arm bands; the positive wire going to source voltage, the negative wire going to ground and a directional channel wire running to the positive terminal of the DAQ equipment. The DAQ equipment was configured to read NRSE to start and later configured to RSE and Differential. The accelerometer was given the same source voltage as the pole and put into a single VI showing the voltage readings.

## **Results**

The accelerometer experienced the same results for all of the DAQ settings, NRSE , RSE and differential. A significant amount of interference was detected by the accelerometer, displaying that the noise is not an effect of multiple accelerometers exclusively. Random spikes of significant value were observed indicating that the interference was not in fact due to wiring of the fishing pole rather due to an outside source which the accelerometers are able to detect.

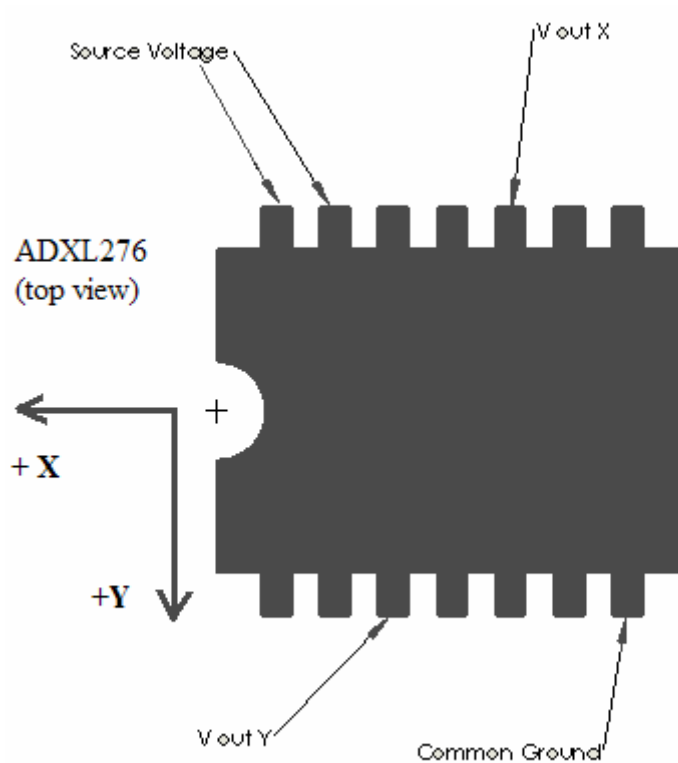
## **Results**

### **Fishing pole design**

#### ***Accelerometer setup***

The positive wire of each accelerometer gets wired to a common source voltage and the negative wires of each accelerometer gets wired to a common ground, both of which are found directly on the power supply. In our case the source voltage is the 5 volts. The remaining wire on the accelerometers are hooked up to the V-out terminal of the accelerometer which goes to the positive channel output. This is the terminal that reads the change in voltage during motion by comparison to the source terminal. The only other wire in the accelerometer configuration is the negative channel; this along with the negative wire of the accelerometer goes to ground. This configuration is the classic RSE setup in Measurements and Automation. An alternative to setting up the accelerometers is NRSE which allows for more possibilities in this MQP. When the Accelerometers are hooked up together they are able to share a single positive and negative wire connecting to the 5 V source and ground. They all keep a separate positive channel wire. The negative wire of the channels is setup as a common ground. This configuration was created by using the NRSE option in Measurements and Automations tool which provides separate positive channels for each accelerometer and a common ground for each accelerometers negative wire. A diagram of the ADXL276 accelerometer is shown below. The source voltage has two tabs into which the positive source voltage wire is soldered to and the common ground is on the opposite corner of the accelerometer in which the negative ground wire is soldered

to. The  $V_{out x}$  and  $y$  are where the channel wires are soldered which run to the DAQ device.

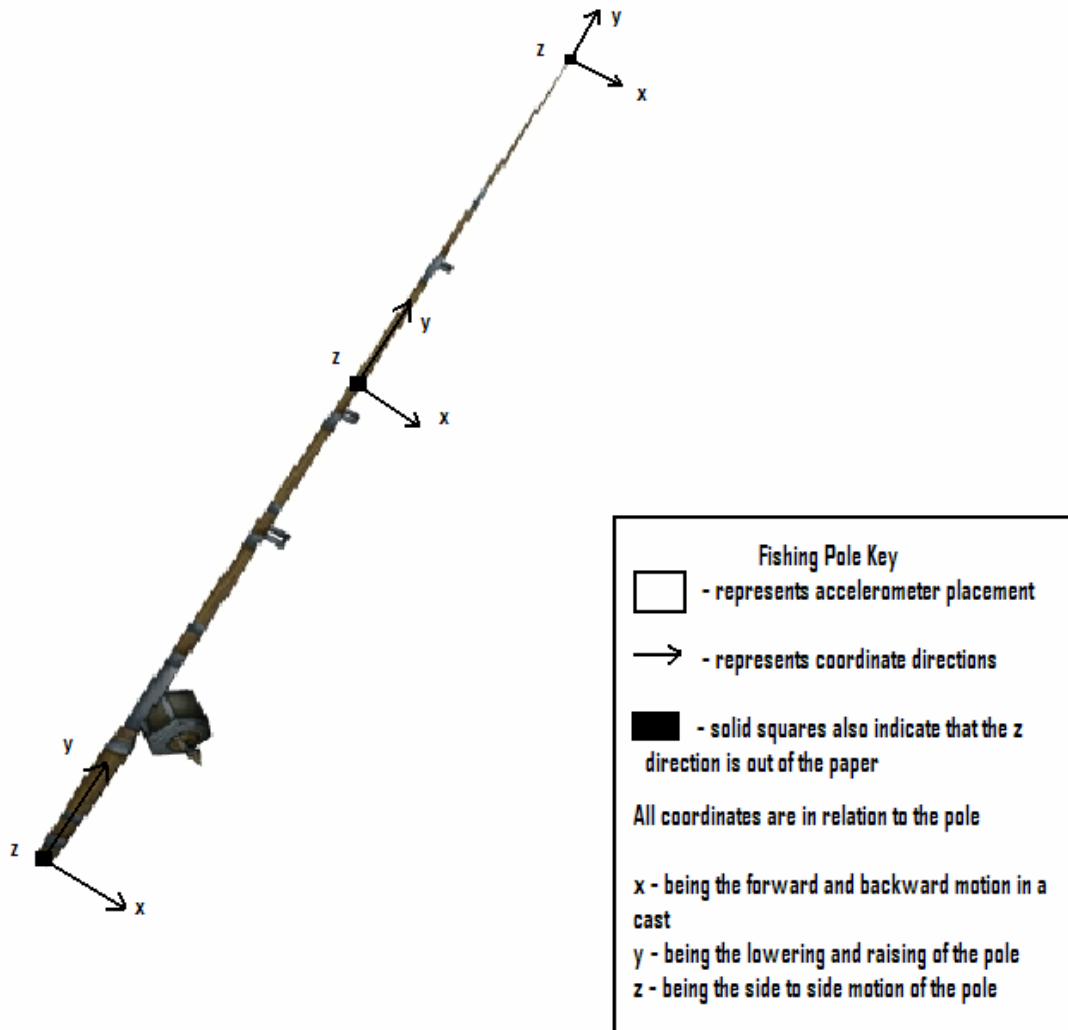


**Figure 23: Accelerometer diagram**

### **Accelerometer Placement on Pole and Arm**

For this project we used a total of 9, 2-axis accelerometers (ADXL276). We have two placed on the upper arm measuring all three x, y and z directions. We have two placed on the lower arm measuring all three x, y and z directions. We also have 2 accelerometers at the top and the bottom of the fishing pole measuring x, y and z directions and the last accelerometer was placed half way down the pole measuring only the x and y directions. When referring to the pole, the x direction is forward and backward motions seen in the cast. The y direction is the upward and downward motion of the pole; this direction runs

parallel with the pole. It is the direction that is the motion from side to side on the pole. The placements of the accelerometers on the pole and their directions are shown below. In the picture below the arrows indicate the positive direction.



**Figure 24: Accelerometer placement and orientation on pole**

For the upper and lower portion of the arm the x direction is the direction measuring the in plane forward and backwards motion. The y direction like in the y direction of the pole is measuring the direction that is parallel to each arm segment. Also like in the case of the fishing pole the z direction is measuring the out of plane motion of

each arm segment, the side to side motion. The placement of these accelerometers and directions can be seen below. The arrows in the picture below indicate the positive directions.

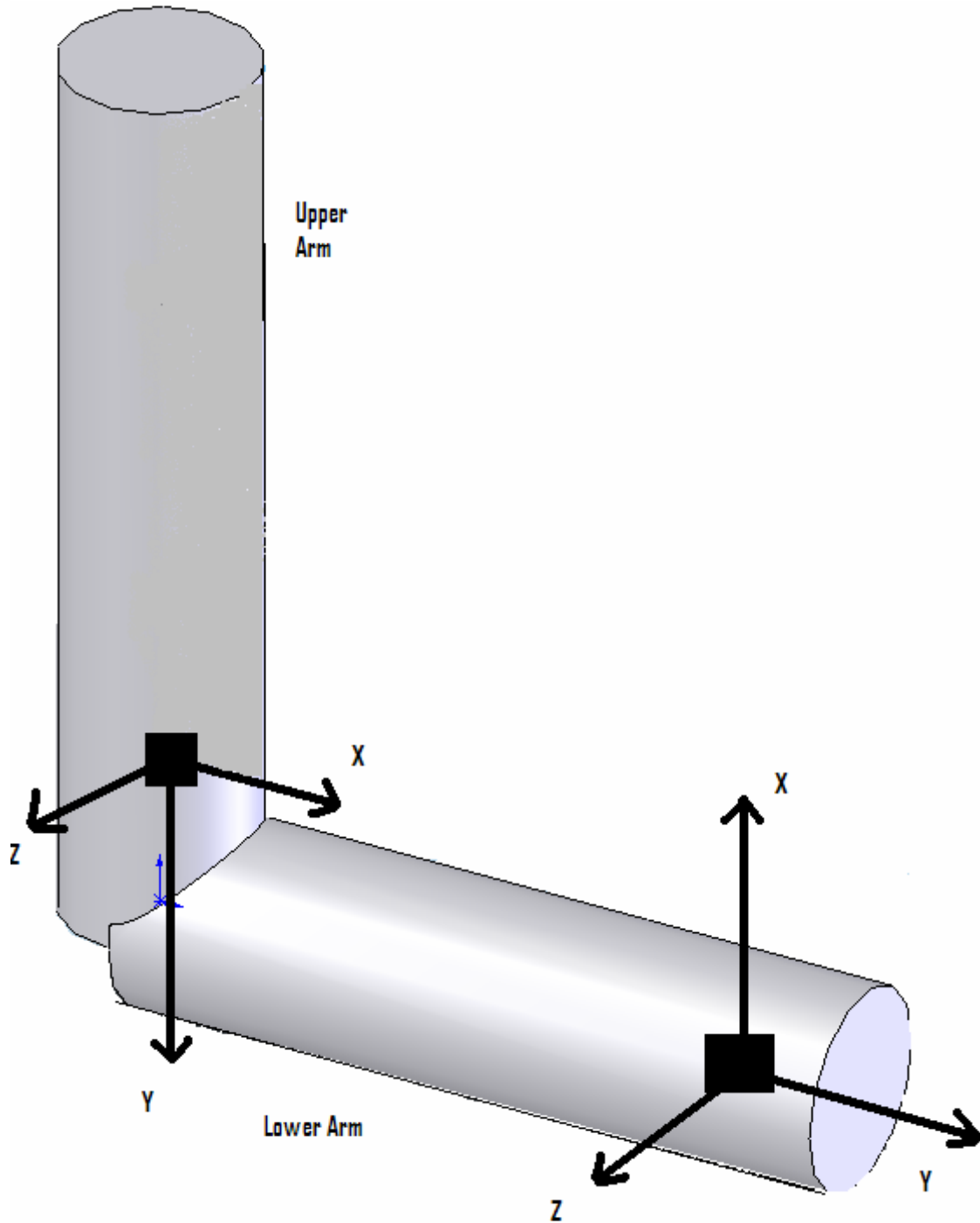
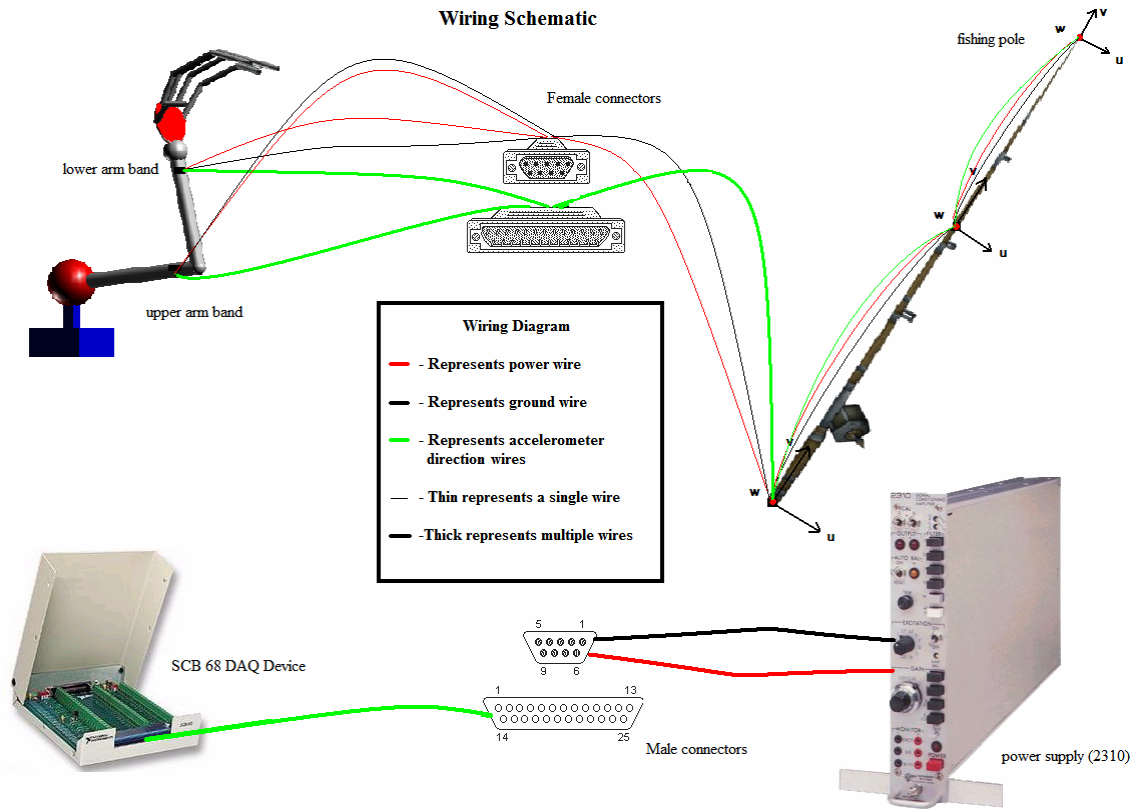


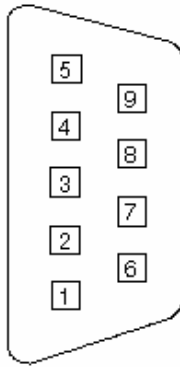
Figure 25: Accelerometer placement and orientation on the arm

## ***Wiring Configuration of the Pole and Arm***

The wiring for the fishing pole and the arm bands are as follows. A positive source voltage and ground wire runs from each of the accelerometers and a single channel wire runs from each of the directional prongs on the accelerometer. On the fishing pole there is only one positive and negative wire which allows all the accelerometers to share a positive source and ground. The accelerometers on the wrist share a positive and ground with each other and the accelerometer on the elbow share a positive and ground with each other as well. All of the positive and ground wires run to the power supply's source voltage and main ground. The channel wires from each accelerometer run to individual channels in a DAQ device that runs to the computer which is controlled through the Measurements and Automation software. The DAQ device that we used for our MQP is DS68. The DAQ device had a quick connect 25 port serial port that connected all of the channels from the pole and the arms to the DAQ device. There was another 9 port serial port connecting all the power and ground wires to the power supply. The wiring configurations are shown below.



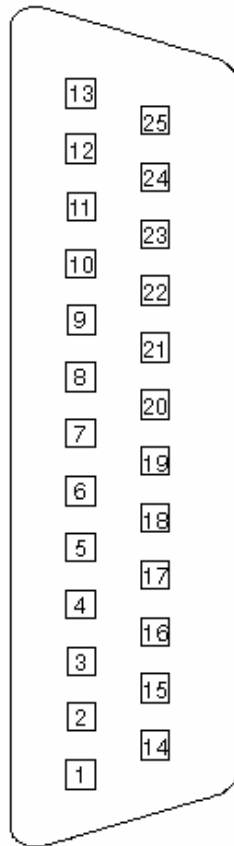
**Figure 26: Wiring diagram**



Male

### 9 Port Power/Ground Connector

Pin Number	Male side goes to	Female side comes from
1	power supply ground	fishing pole ground pins
2	power supply ground	upper arm band ground pin
3	power supply ground	lower arm band ground pin
4	power supply ground	SCB channel 62
5	n/a	n/a
6	power supply power	fishing pole power pins
7	power supply power	upper arm band power pins
8	power supply power	lower arm band power pins
9	n/a	n/a



Male

### 25 Port Channel Connector

Pin Number	Male side goes to	Female side comes from
1	n/a	n/a
2	n/a	n/a
3	n/a	n/a
4	n/a	n/a
5	n/a	n/a
6	n/a	n/a
7	n/a	n/a
8	SCB 68 pin 66 channel 9	tip of fishing pole w
9	SCB 68 pin 31 channel 10	tip of fishing pole v
10	SCB 68 pin 63 channel 11	tip of fishing pole u
11	SCB 68 pin 61 channel 12	middle of fishing pole w
12	SCB 68 pin 26 channel 13	middle of fishing pole u
13	SCB 68 pin 62 ground	power supply ground
14	n/a	n/a
15	n/a	n/a
16	SCB 68 pin 68 channel 0	bottom of fishing pole w
17	SCB 68 pin 33 channel 1	bottom of fishing pole v
18	SCB 68 pin 65 channel 2	bottom of fishing pole u
19	SCB 68 pin 30 channel 3	upper arm band w
20	SCB 68 pin 28 channel 4	upper arm band v
21	SCB 68 pin 60 channel 5	upper arm band u
22	SCB 68 pin 25 channel 6	lower arm band u
23	SCB 68 pin 57 channel 7	lower arm band w
24	SCB 68 pin 34 channel 8	lower arm band v
25	n/a	n/a

Figure 27: Pin diagram for quick-connects

## Mathematics

### Two Dimensional Space Equations

The Two dimensional analyses are done for the first two links of the arm. Link 1 being from the shoulder to the elbow,  $r_1$  and link 2 being from the elbow to the wrist,  $r_2$ . Our exact calculations are as follows. First we translate the voltages received in each direction from the accelerometers into  $m/s^2$ . We did this using the following equation provided to us by a spec sheet from analog devices in regards to the ADXL 276.

$$A = \frac{\left(\frac{V_s}{2} - A_v\right)}{(b * V_s) + c * V_s^2} * g$$

$A$  being the linear acceleration found from the accelerometer voltage readings, and  $A_v$  being the voltage reading taken from the accelerometer.  $V_s$  being the source voltage provided to the accelerometer.  $b$  and  $c$  both being constants provided by the makers of the ADXL 276 spread sheet and  $g$  being gravity.

Next these accelerations are then translated into accelerations at the wanted points, the elbow and the wrist. This was done by using the linear relationship between the differences in the radii from the accelerometer position to the wanted position. Those equations are as follows.

$$\begin{aligned} A_{1y} &= A_{1ya} * (r_1 / r_{1a}) \\ A_{2x} &= A_{2xa} * (r_2 / r_{2a}) \\ A_{2y} &= A_{2ya} * (r_2 / r_{2a}) \end{aligned}$$

Where  $A_{1y}$  is the new acceleration at the elbow based off the reading found at the accelerometer;  $r_1$  is the length from the shoulder to the elbow and  $r_{1a}$  is the length from the shoulder to the accelerometer.  $A_{2x}$  and  $A_{2y}$  are similar in relation to the wrist and elbow.

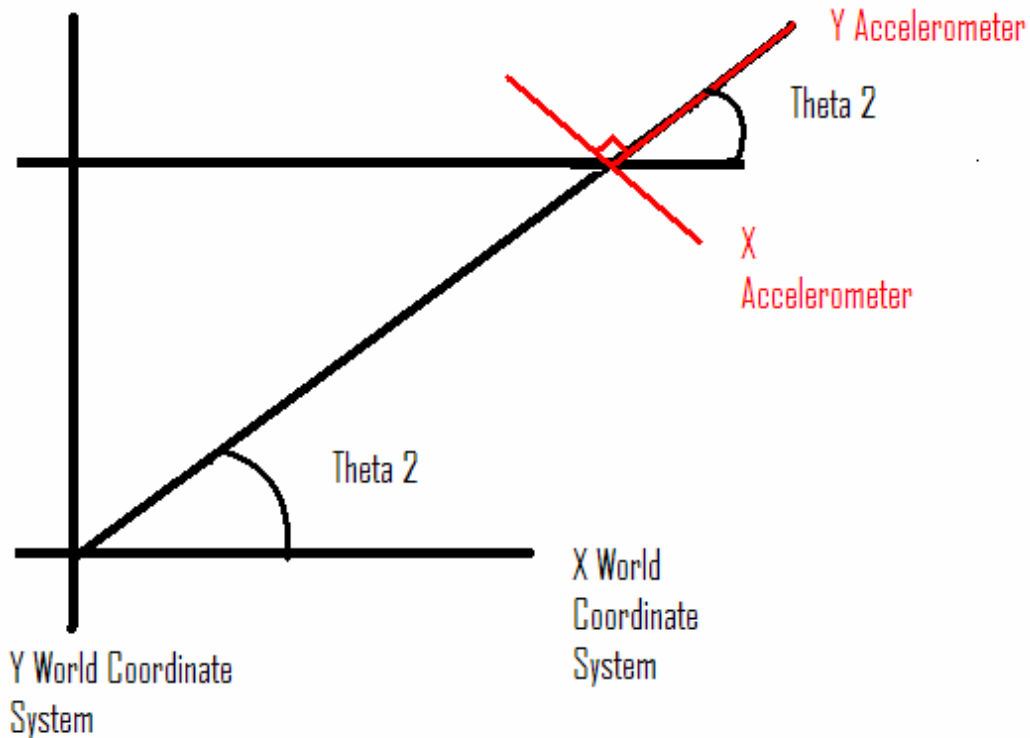
Next the acceleration of each of these is then broken down into their  $x$  and  $y$  components. As  $A_{1y}$  is based solely off of rotation it is now translated into angular acceleration. This is done using the following equation.

$$A1_{yc} = A1_y / r1$$

At the same time that acceleration for the first part of the arm is being translated into angular acceleration the second portion of the arm is being translated into its component vectors in the world coordinate system. This is done using the following equation.

$$A2_{xc} = (A2_{xa} * \cos(s20) + A2_{ya} * \sin(s20))$$
$$A2_{yc} = -(A2_{xa} * \sin(s20) + A2_{ya} * \cos(s20))$$

Here, s20 is the initial angle below the forearm. Here is one of the places in which we used a shift register in which an upgraded s20 is plugged in for each time step. The sin and cos relationship of coordinate can be seen in the picture shown below.



$$\begin{aligned} X_{\text{component of X Accelerometer}} &= X \text{ Accelerometer} * \cos(90 - \text{theta } 2) \\ Y_{\text{component of X Accelerometer}} &= X \text{ Accelerometer} * \sin(90 - \text{theta } 2) \\ X_{\text{component of Y Accelerometer}} &= Y \text{ Accelerometer} * \cos(\text{theta } 2) \\ Y_{\text{component of Y Accelerometer}} &= Y \text{ Accelerometer} * \sin(\text{theta } 2) \end{aligned}$$

**Figure 28: Two dimensional coordinate system transformation**

Next the angular velocity is found for the first part of the arm in order to use the angular acceleration and angular velocity to find the new angle after the time step. The equation used to find the angular velocity is as follows:

$$V1yc = (A1yc * dt) + V1yc0$$

V1yc is the angular velocity and V1yc0 being the previous time steps velocity and dt being the time in milliseconds. Once again a shift register is used to plug the V1yc from the first time step into V1yc0 in the next time step. From here the angle between the x axis and the first portion of the arm is found:

$$s1 = \frac{(A1_{yc} * dt^2)}{2} + (V1_{yc} * dt) + s10$$

s1 being the angle at the shoulder and s10 being the angle found in the previous time step, this as well uses a shift register to plug s1 found in time step 1 into s10 in time step 2 and so on. Next the linear components of the first part of the arm are found in order to track graphically the change in position. The first thing that is translated to linear is velocity. This is done so using the following equation.

$$V1 = (A1_{yc} * dt) * r1 + V10$$

Where V1 is the linear velocity and V10 is the linear velocity found from the previous time step. A1<sub>yc</sub> being the angular acceleration is translated to linear acceleration by multiplying by the radius of the upper arm. From here the velocity can be broken down into its component vectors by simply taking the sin and cos of the angle and multiplying by the non-componentized velocity.

$$V1_x = V1 * \cos(s1)$$

$$V1_y = V1 * \sin(s1)$$

Next the acceleration is componentized by first making the angular acceleration linear and then taking the sin and cos of the angle. This was done as follows.

$$A1_x = A1_{yc} * r1 * \sin(s1)$$

$$A1_y = A1_{yc} * r1 * \cos(s1)$$

This now gives us the acceleration due solely to the shoulder. Next the actual acceleration due to just the elbow at point 2 can be calculated by subtracting the acceleration due to the shoulder.

$$A2_x\_actual = A2_{xc} - A1_x$$

$$A2_y\_actual = A2_{yc} - A1_y$$

Now that the actual acceleration due to just the elbow is found the actual velocity due to just the elbow can be found using the equation below.

$$V2_{xc} = (A2_{x\_actual} * dt) + V2_{xc0}$$

$$V2_{yc} = (A2_{y\_actual} * dt) + V2_{yc0}$$

Where  $V2_{yc}$  and  $V2_{xc}$  are the actual velocities at the wrist and  $V2_{yc0}$  and  $V2_{xc0}$  are the actual velocities found in the previous time-step. In the next frame we find both the x and y position of the elbow and convert the acceleration of the second part of the arm into angular acceleration using the following equations.

$$x1 = r1 * \cos(s1)$$

$$y1 = r1 * \sin(s1)$$

$$a2_{ang} = \frac{(((A2_{x\_actual}^2) + (A2_{y\_actual}^2))^{1/2}}{r2}$$

Now all of the factors for the first part of the arm are known, and all that is left is to find the actual position of the second part of the arm and to track the angle of the second part of the arm. So in the next frame we find the angular velocity in order to later find our second angle.

$$v2_{ang} = (a2_{ang} * dt) + v20_{ang}$$

Now knowing the angular velocity and angular acceleration due to just the elbow we are able to find the angle below the forearm using the following equation.

$$s2 = \frac{((v2_{ang} * dt) + (a2_{ang} * dt^2))}{2} + s20$$

Now we find the x and y components of our new position based off of the origin using the following equations.

$$x_{2s} = \left( \frac{A_{2x} - a_{ct} * dt^2}{2} + V_{2xc} * dt + x_{2s0} \right)$$

$$y_{2s} = \left( \frac{A_{2y} - a_{ct} * dt^2}{2} + V_{2yc} * dt + y_{2s0} \right)$$

Now that we know the following position based off of the origin we need to convert it to the final position in WCS. Knowing that the equations for the second portion of the arm are using the elbow as our origin we can simply find the world coordinates by adding the elbow coordinates to the new coordinates. This is done as follows.

$$x_{end} = x_1 + x_{2s}$$

$$y_{end} = y_1 + y_{2s}$$

These are the final x and y coordinates of the wrist. We are now able to track both the elbow position and the wrist position. Using Lab-view and the VI system we are able to track and graph in plane motion of the arm based off of the previous equations as the equations are completed at every time step instantaneously.

### ***Three Dimensional Analysis***

We started the three dimensional analysis in the same manner as in the two dimensional analysis. We took the Voltage reading from the accelerometer and translated it into linear acceleration in terms of m/s<sup>2</sup>.

$$A = \frac{\left( \frac{V_s}{2} - Av \right)}{(b * V_s) + c * V_s^2} * g$$

We then converted these acceleration readings into the Cartesian coordinates in the World Coordinate system (WCS). We did this using the transformation matrix, which

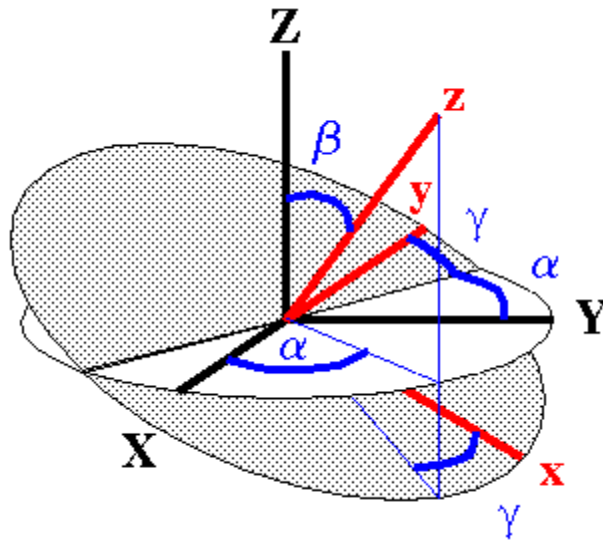
converts component vectors in one coordinate system into a related coordinate system. In our case the related coordinate system is the WCS. The matrix that converts World Coordinate components into the original coordinate system can be seen below.

$$\begin{array}{ccc} \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma & \sin \alpha \cos \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \beta \cos \gamma \\ -\cos \alpha \cos \beta \sin \gamma - \sin \alpha \cos \gamma & -\sin \alpha \cos \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \beta \sin \gamma \\ \cos \alpha \sin \beta & \sin \alpha \sin \beta & \cos \beta \end{array}$$

The matrix above multiplied by our coordinates gives the coordinates of the original matrix, not the world coordinate components. As we are looking for the world coordinates we need to multiply the known vectors by the transpose of that matrix. The transpose of the above matrix is as follows.

$$\begin{array}{ccc} \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma & -\cos \alpha \cos \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \\ \sin \alpha \cos \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \cos \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \\ -\sin \beta \cos \gamma & \sin \alpha \sin \beta & \cos \beta \end{array}$$

This matrix multiplied by our known accelerations will give us the accelerations in the World Coordinate system. Alpha Beta and Gamma represent the same angle in both matrices. Alpha represents the change in angle in the xy plane from our known coordinate values to our world coordinate system. Beta represents the change in angle off our xy plane. In other words the angle between our known z components to the world coordinates z component. Lastly gamma is the twist off of the xy plane. This is the angle that gives the projection onto the world coordinates plane. Below is a demonstration of the angles in relation to the known coordinates labeled in red with the world coordinates labeled in black.



**Figure 29: Three dimensional coordinate system transformation diagram (Eichele)**

As previously stated the world coordinate accelerations are found by multiplying the second matrix shown above by each accelerometers readings which have been translated into  $m/s^2$ . For the first portion of our arm, from the shoulder to the elbow, there is no internal rotation involved therefore gamma is going to equal 0 canceling out much of the above matrix. For the second portion of the arm the all of the angles are a factor therefore the entire matrix will be used for that portion of the arm. An example of the world coordinate accelerations is shown below:

$$\begin{aligned}
 Ax &= (\cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma) * AccelX + (-\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma) * AccelY + (\cos\alpha\sin\beta) * AccelZ \\
 Ay &= (\sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma) * AccelX + (-\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma) * AccelY + (\sin\alpha\sin\beta) * AccelZ \\
 Az &= (-\sin\beta\cos\gamma) * AccelX + (\sin\alpha\sin\beta) * AccelY + (\cos\beta) * AccelZ
 \end{aligned}$$

Where  $A_x$ ,  $A_y$  and  $A_z$  are the component accelerations in the world coordinate system.

$AccelX$ ,  $AccelY$  and  $AccelZ$  are the accelerations in initial coordinate system. Now that

the accelerations are known in the world coordinate system we now translate these

acceleration readings from the points where the accelerometers are on the arm to the points

which we would like to know about. For the first portion of the arm we translate the

acceleration from the point on the upper arm where the accelerometer is mounted to the elbow. We do this by simply using the ratio of the distance between the shoulder and the accelerometer ( $R1$ ) to the distance between the shoulder and elbow ( $R1a$ ). Same was done for the upper arm only this time  $R2$  represents the distance from the elbow to wrist and  $R2a$  the distance from the elbow to the accelerometer mounted just below the wrist. In the below equation  $A_{true}$  is the true acceleration at the point of interest and  $Ra$  is the length to the accelerometer in question i.e  $R1a$ ,  $R2a$ .

$$A_{true} = A * \frac{R}{Ra}$$

For the second portion of the arm the above acceleration is due to both the acceleration created by the shoulder and the acceleration created by the elbow therefore we need to translate the acceleration read at the second portion of the arm into acceleration due to only the elbow. We do this by subtracting the acceleration due to the shoulder which was found for portion 1 of the arm.

$$A2_{true} = (A2 * \frac{R2}{R2a}) - A1_{true}$$

Now that the accelerations are showing at the point on the arms in which we are interested in we then find the velocities that accompany those acceleration values. We do this using the equation shown below.

$$V = A_{true} * dt + V0$$

Where  $dt$  is the time taken to go from initial to end and  $V0$  is the initial velocity at the beginning of the time step. Now that the accelerations and velocities are known we then find the position in terms of  $x,y,z$ . This was done using the equation:

$$\begin{aligned}
x &= x_0 + V_{x0} * dt + ((A_x * (dt) ** 2) / 2) \\
y &= y_0 + V_{y0} * dt + ((A_y * (dt) ** 2) / 2) \\
z &= z_0 + V_{z0} * dt + ((A_z * (dt) ** 2) / 2)
\end{aligned}$$

Where  $x_0$ ,  $y_0$  and  $z_0$  are the initial positions at the beginning of the time step. This was done for both portion of the arm. Based off of the calculations until this point both arm portions are based off of the origin. This is fine for the first part of the arm as the shoulder is a fixed point but the second part of the arm need to move in relation to the first part of the arm. We fix this issue by adding the changing elbow position to the new found coordinates for the wrist using the equation below.

$$\begin{aligned}
x_{2true} &= x_2 + x_1; \\
y_{2true} &= y_2 + y_1; \\
z_{2true} &= z_2 + z_1;
\end{aligned}$$

Now the true  $x$   $y$  and  $z$  positions for the second part of the arm in relation to the changing elbow position are known. These points are then graphed in a three dimensional graph to track the motion of the arm. The only thing that is needed to track these motions is to be able to account for the change in the original angles  $\alpha$   $\beta$  and  $\gamma$ . This information will be found after the positions and placed back into the system to account for these changing angles. In order to track the changing angles we first need to work backwards from where we are now. We need to take the initial matrix that was found and relate the positions back to our original coordinate system. To do this we take the matrix below and multiply that by our world coordinate position.

$$\begin{aligned}
x_{local} &= (\cos\alpha \cos\beta \cos\gamma - \sin\alpha \sin\gamma) * x_1 & + (\sin\alpha \cos\beta \cos\gamma + \cos\alpha \sin\gamma) * y_1 & + (-\sin\beta \cos\gamma) * z_1 \\
y_{local} &= (-\cos\alpha \cos\beta \sin\gamma - \sin\alpha \cos\gamma) * x_1 & + (-\sin\alpha \cos\beta \sin\gamma + \cos\alpha \cos\gamma) * y_1 & + (\sin\beta \sin\gamma) * z_1 \\
z_{local} &= (\cos\alpha \sin\beta) * x_1 & + (\sin\alpha \sin\beta) * y_1 & + \cos\beta * z_1
\end{aligned}$$

Where  $x_{local}$ ,  $y_{local}$ , and  $z_{local}$  are the coordinates in the original coordinate system and  $x_1$ ,  $y_1$  and  $z_1$  are the coordinates in the world coordinate system. Now that we know the new position in the world coordinate system and the new position in the local coordinate system we are able to find how the angles change through the following relationship.

$$\begin{aligned}\alpha &= \cos^{-1}(y_1/y_{1local}); \\ \beta &= \cos^{-1}(z_1/z_{1local}); \\ \gamma &= \cos^{-1}(x_1/x_{1local}) - \alpha;\end{aligned}$$

We then plug these new angles into the next time step which now update the next time steps positions and components. We did however run into some issues with these angles when trying to use these equations in our VI.

## **Final LabView VIs**

At the end of the project, we had created two LabView programs to record readings and convert them from voltages to accelerations, and eventually to positions. The first program converts the readings from voltages to accelerations, in the process smoothing them through a running average, and zeroing them out through a true-false loop. These calculated values are then saved and loaded into the second program, where they are converted to positions and ultimately graphed.

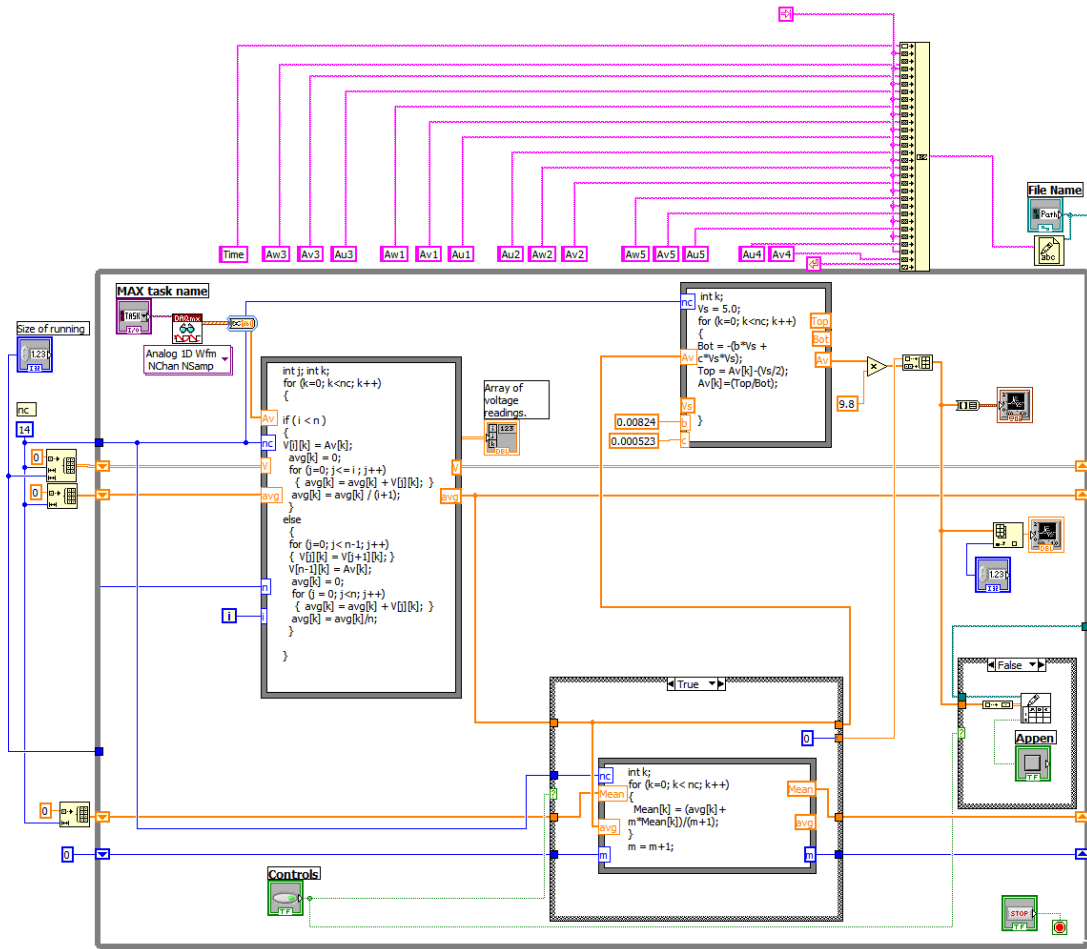


Figure 30: Block Diagram of First VI

Our first VI takes the voltage readings as an array, and runs them first through a formula node. This formula node is set to run a moving average of each of the channels, using array notation. We are able to adjust the amount of readings which the average uses by means of an “Array Size” numerical control. We found that an array size of 200 readings provided a good balance between smoothness and sensitivity.

```

int j; int k;
for (k=0; k<nc; k++)
{
  if (i < n)
  {
    V[i][k] = Av[k];
    avg[k] = 0;
    for (j=0; j<=i; j++)
    { avg[k] = avg[k] + V[j][k]; }
    avg[k] = avg[k] / (i+1);
  }
  else
  {
    for (j=0; j< n-1; j++)
    { V[j][k] = V[j+1][k]; }
    V[n-1][k] = Av[k];
    avg[k] = 0;
    for (j = 0; j<n; j++)
    { avg[k] = avg[k] + V[j][k]; }
    avg[k] = avg[k]/n;
  }
}

```

**Figure 31: First formula node of first VI**

After leaving the moving average, the readings move into a True/False loop. Due to the number of channels and the single ground setup, we were getting actual steady state readings in a range from 2.4 to 2.6 volts. The transfer function requires an exact reading of 2.5 volts for the acceleration in g to be zero at steady state. We are able to transfer our readings to 2.5 volts with the loop. When true, the readings enter a formula node where they are averaged over an indeterminate amount of time, controlled by the user. This average is taken while the physical system is not moving.

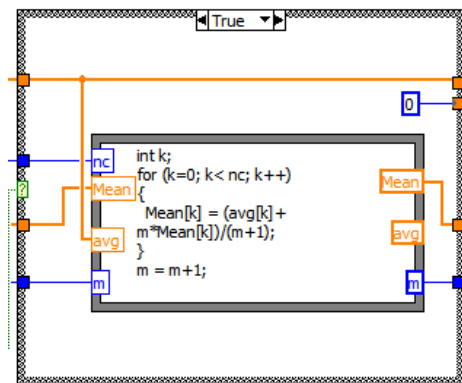
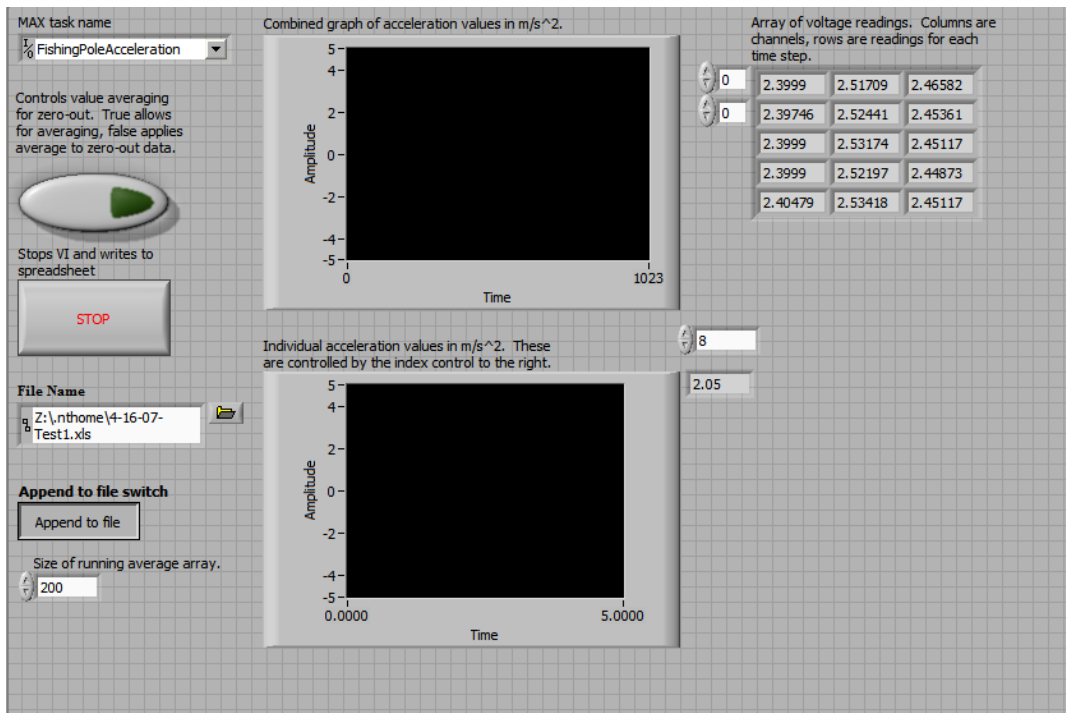


Figure 32: True/false loop and second formula node of first VI

When the loop is false, the calculated average is subtracted from 2.5 volts, to find the difference between where the readings are and where they should be. This difference is then added back onto the ongoing readings to translate them to 2.5v. The same button which controls this True/False loop also controls a subsequent loop which houses the Write to Spreadsheet block. When the button is set to false, the program records the files to the spreadsheet, as they have been corrected.



**Figure 33: Front Panel of First VI**

The corrected voltages then go into another formula node, this time containing the transfer function for acceleration. Once again, the node is in array notation and simply iterates through all of the channels.

```

nc
int k;
Vs = 5.0;
for (k=0; k<nc; k++)
{
  Bot = -(b*Vs +
  c*Vs*Vs);
  Top = Av[k]-(Vs/2);
  Av[k]=(Top/Bot);
}

```

Figure 34: Third formula node of second VI

The output of this node is in g, and must be converted to m/s<sup>2</sup> to be used in the second VI. Therefore, the output is multiplied by 9.8. Along with being written to the spreadsheet file, the output readings are plotted on two waveform charts. The first plots all of them simultaneously for quick visual verification of the readings. The second uses an array indexer to cycle through each channel, and is primarily used for debugging any issues. Finally, the output accelerations are written to the spreadsheet file for the next VI.

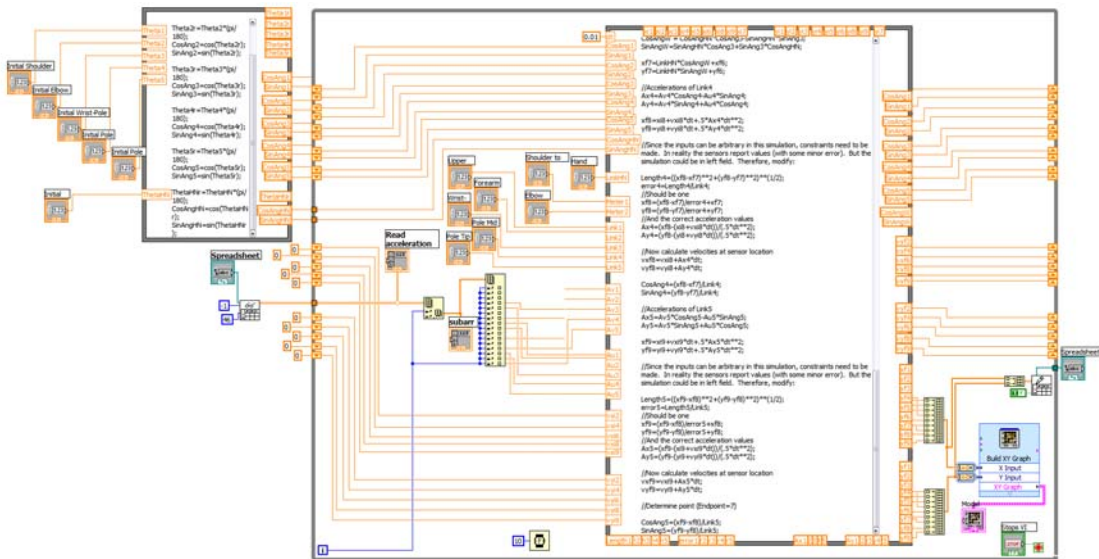
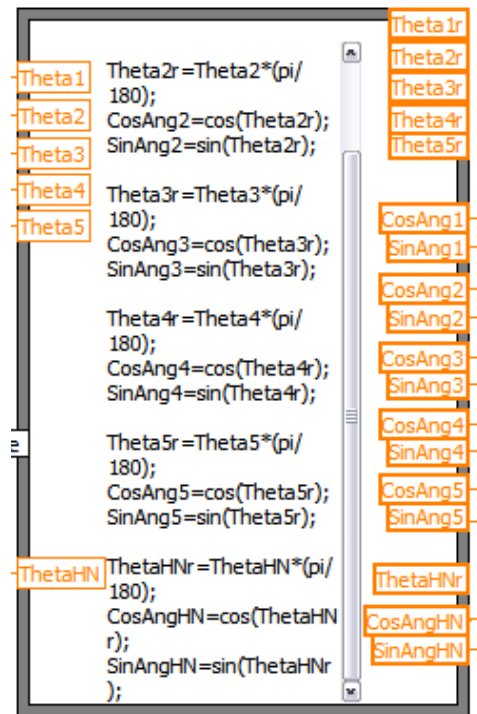


Figure 35: Block Diagram of Second VI

Our second VI begins with measurements of the various lengths and initial angles for the current cast. Lengths are entered in numerical controls which are wired directly to inputs in the main formula node of the system. The angles, however, are entered in

controls which are wired into a formula node outside of the while loop. This formula node converts the angles to their appropriate cosine and sine functions. We do this as the trigonometric functions are recalculated at the end of each time step, then shift registered back into the while loop for the following time step. Placing the calculation of the initial functions outside of the while loop allows them to be used for the first time step, while the recalculated values are used for every subsequent step.



**Figure 36: First formula node of second VI**

Our accelerations are read in the “Read Spreadsheet” block, and then come into an Index Array block, where they are separated for analysis. At this point, the accelerations are arrays, and cannot be plugged directly into the formula node without conversion.

Therefore, we use both to dynamic data and from dynamic data conversions to get the accelerations in the appropriate format. They are then wired to the appropriate inputs in the main formula node. This main formula node then performs all of the appropriate

calculations to produce a graph of the arm positions. Velocities and the trigonometric functions of each angle are recalculated at the end of each time step and shift register back into the loop for use in the following step. The change in time,  $dt$ , is actually read from the recorded times of the previous VI, and is a calculation of the difference between two time steps.

```

dt          xi1 xi2 xi3 xi4 xi5 xi6 xi8 xi9 xi7      yi1 yi2 yi3 yi4 yi5 yi6 yi8 yi9      yi7
-CosAngW   CosAngW = CosAngHN*cosAng3-SinAngHN*sinAng3;
SinAngW   SinAngW = SinAngHN*cosAng3+SinAng3*cosAngHN;
-CosAng1
-SinAng1
-CosAng2   xf7=LinkHN*cosAngW+xf6;
-SinAng2   yf7=LinkHN*sinAngW+yf6;
-CosAng3
-SinAng3   //Accelerations of Link4
-CosAng4   Ax4=Av4*cosAng4-Au4*sinAng4;
-SinAng4   Ay4=Av4*sinAng4+Au4*cosAng4;
-CosAng5   xf8=xi8+vxi8*dt+.5*Ax4*dt**2;
-SinAng5   yf8=yi8+vyi8*dt+.5*Ay4*dt**2;
-CosAngHN
-SinAngHN  //Since the inputs can be arbitrary in this simulation, constraints need to be
           made. In reality the sensors report values (with some minor error). But the
           simulation could be in left field. Therefore, modify:
-LinkHN    Length4=((xf8-xf7)**2+(yf8-yf7)**2)**(1/2);
           error4=Length4/Link4;
           //Should be one
-Meter1    xf8=(xf8-xf7)/error4+xf7;
-Meter2    yf8=(yf8-yf7)/error4+yf7;
           //And the correct acceleration values
-Link1     Ax4=(xf8-(xi8+vxi8*dt))/(.5*dt**2);
-Link2     Ay4=(yf8-(yi8+vyi8*dt))/(.5*dt**2);
-Link3
-Link4     //Now calculate velocities at sensor location
-Link5     vxf8=vxi8+Ax4*dt;
           vyf8=vyi8+Ay4*dt;
-Av1       CosAng4=(xf8-xf7)/Link4;
-Av2       SinAng4=(yf8-yf7)/Link4;
-Av3
-Av4
-Av5       //Accelerations of Link5
           Ax5=Av5*cosAng5-Au5*sinAng5;
           Ay5=Av5*sinAng5+Au5*cosAng5;
-Au1       xf9=xi9+vxi9*dt+.5*Ax5*dt**2;
-Au2       yf9=yi9+vyi9*dt+.5*Ay5*dt**2;
-Au3
-Au4
-Au5       //Since the inputs can be arbitrary in this simulation, constraints need to be
           made. In reality the sensors report values (with some minor error). But the
           simulation could be in left field. Therefore, modify:
-vxi2      Length5=((xf9-xf8)**2+(yf9-yf8)**2)**(1/2);
-vxi4      error5=Length5/Link5;
           //Should be one
-vxi6      xf9=(xf9-xf8)/error5+xf8;
-vxi8      yf9=(yf9-yf8)/error5+yf8;
-vxi9      //And the correct acceleration values
           Ax5=(xf9-(xi9+vxi9*dt))/(.5*dt**2);
           Ay5=(yf9-(yi9+vyi9*dt))/(.5*dt**2);
-vyi2      //Now calculate velocities at sensor location
-vyi4      vxf9=vxi9+Ax5*dt;
-vyi6      vyf9=vyi9+Ay5*dt;
-vyi8
-vyi9      //Determine point (Endpoint=7)
           CosAng5=(xf9-xf8)/Link5;
           SinAng5=(yf9-yf8)/Link5;
-LinkHN    Length1=h2+h3+h4+h5;
           error1=r2+r3+r4+r5;
-Ax1       Ax1=r2+r3+r4+r5;
-Ay1       Ay1=r2+r3+r4+r5;
           vxf2
           vxf4
           vxf6
           vxf8
           vxf9
           vyf2
           vyf4
           vyf6
           vyf8
           vyf9
           xf1
           xf2
           xf3
           xf4
           xf5
           xf6
           xf7
           xf8
           xf9
           yf1
           yf2
           yf3
           yf4
           yf5
           yf6
           yf7
           yf8
           yf9

```

Figure 37: Second formula node of second VI

The final end positions are then gathered in two arrays, one for the x direction and one for the y direction, and plugged into an XY Graph, where they are then displayed on the front panel. The positions are also written to a spreadsheet for use in a subsequent VI, where two

sets of positions are read into a graph simultaneously, for visual comparisons between two casts.

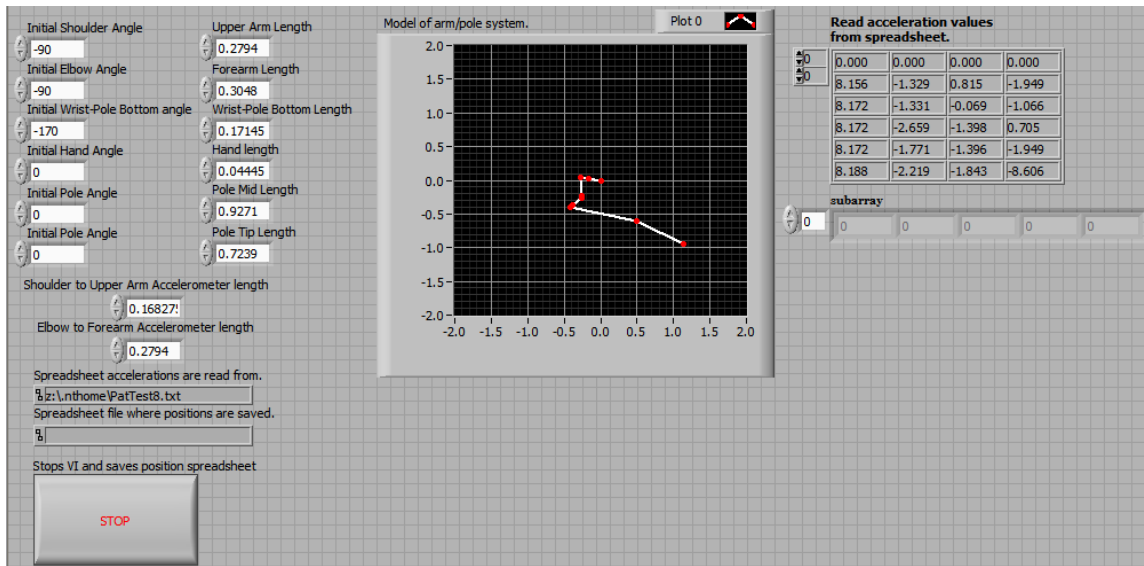


Figure 38: Front Panel of Second VI

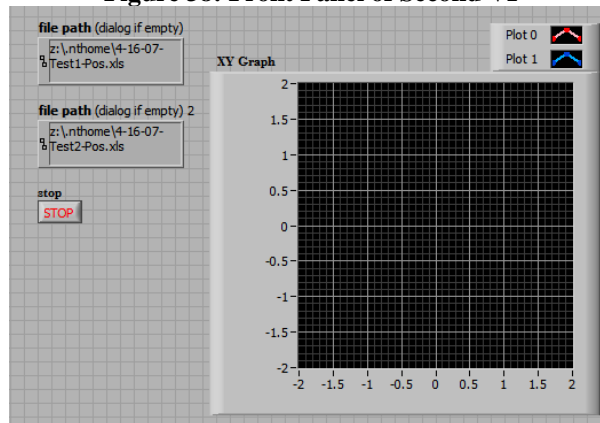


Figure 39: Front Panel of Third VI

## Formula Node Calculations

Within the VIs, there are formula nodes which manipulate our readings mathematically to produce the desired results. Each of these nodes has various inputs and outputs to define the variables within them. Those inputs and outputs are then wired to the appropriate places throughout the rest of the VI, depending on the task. Our first VI has three formula nodes, while our second has two.

## **First VI**

### **Moving Average**

The moving average node reduces noise in the readings by taking the average of several readings to output one reading. In this node, Av, the input voltage, comes in as an array of the channels. V is a two dimensional array which takes the actual running averages, and is shift registered back into the node to continue the averaging. Avg is an array of the averaged values (also shift registered to be used in the next time step), j and k are both dimensions, i is the number of time steps, and nc and n are the number of values to be averaged.

The first two lines define j and k to be iterated, and the entire node to be run in iterations from k equaling zero throughout the number of values to be averaged.

```
int j; int k;  
for (k=0; k<nc; k++)
```

The “if” section of the node runs if the number of the current time step is less than the number of values to be averaged. It subsequently defines the average to be taken up to the current time step.

```
{  
if ( i < n )  
{  
V[i][k] = Av[k];  
avg[k] = 0;  
for (j=0; j<= i ; j++)  
{ avg[k] = avg[k] + V[j][k]; }  
avg[k] = avg[k] / (i+1);  
}
```

The “else” section node runs as long as the number of the time step is higher than the number of values to be averaged. This allows the average to be taken as normal, using the total number of values specified.

```

else
{
for (j=0; j < n-1; j++)
{ V[j][k] = V[j+1][k]; }
V[n-1][k] = Av[k];
avg[k] = 0;
for (j = 0; j < n; j++)
{ avg[k] = avg[k] + V[j][k]; }
avg[k] = avg[k]/n;
}
}

```

## Zero Out

The next formula node also takes a moving average, but this time is used solely to find the steady-state average. The output of the previous formula node, avg, comes in as an array and is added to the array Mean, which is initially zero but is recalculated each time step and shift-registered back in for the next step. This value is then divided by m, the number of iterations the node has gone through, added to one. This is then the new Mean which is outputted to both a shift register and the next frame of the stacked sequence. Once again, the first two lines define the arrays and iterate them.

```

int k;
for (k=0; k < m; k++)
{
Mean[k] = (avg[k]+ m*Mean[k])/(m+1);
}
m = m+1;

```

## Conversion to Acceleration

The final node in this VI takes the array of voltages and converts them to accelerations. We define Vs as the source voltage, and Av as an array of voltages. Both b and c are constants which are defined by the specifications of the accelerometer. The

transfer function is divided into Top and Bot (bottom) components for simplification. Av is recalculated using the function, and outputs into the arrays for the spreadsheet.

```
int k;
Vs = 5.0;
for (k=0; k<nc; k++)
{
Bot = -(b*Vs + c*Vs*Vs);
Top = Av[k]-(Vs/2);
Av[k]=(Top/Bot);
}
```

## **Second VI**

### **Trigonometric Functions**

The first formula node in the second VI uses our inputted initial angles of Theta to be converted to cosine and sine functions. First, Theta is converted to radians, resulting in Theta-r. Then the trigonometric functions are found and defined for that angle (CosAng and SinAng, respectively). These calculations are only shown for the first angle; all of the calculations for subsequent angles are identical.

```
Theta1r=Theta1*(pi /180);
CosAng1=cos(Theta1r);
SinAng1=sin(Theta1r);
```

### **Acceleration to Position**

Our largest formula node takes the accelerations, read from a spreadsheet from the first VI, and turns them into positions which can then be graphed. First, the initial positions of x and y for each arm must be defined for later in the calculation. We also define the final position for the shoulder because it is ground and is stationary. As with the above calculations, these are repeated for every link on the arm, with special instances noted where necessary.

```

xi 1= 0. 0;
yi 1= 0. 0;

xf1= 0. 0;
yf1= 0. 0;

xi 2=CosAng1*Meter1+xi 1;
yi 2=Si nAng1*Meter1+yi 1;
xi 3=CosAng1*Li nk1+xi 1;
yi 3=Si nAng1*Li nk1+yi 1;

```

We also define the angles for our hand correction. Angle W (the angle from horizontal to the imaginary line between the wrist and the bottom of the pole) is the sum of angles 3 (the hand to the imaginary line) and HN (the hand to horizontal), and the cosine and sine of that angle is calculated accordingly.

```

CosAngW = CosAngHN*CosAng3-Si nAngHN*Si nAng3;
Si nAngW=Si nAngHN*CosAng3+Si nAng3*CosAngHN;

```

Next we can determine the x and y accelerations for each angle based on the u and v accelerations from the spreadsheet file, using the cosine and sine of the angle.

```

Ax1=Av1*CosAng1-Au1*Si nAng1;
Ay1=Av1*Si nAng1+Au1*CosAng1;

```

Then we can find the position of the point. Vxi and vyi, the velocities at the start of the time step, are initially zero and get recalculated later in the node, to be shift registered in for the next step.

```

xf2=xi 2+vxi 2*dt+. 5*Ax1*dt**2;
yf2=yi 2+vyi 2*dt+. 5*Ay1*dt**2;

```

To ensure that our positions are properly constrained to the length of the actual arm segment, they go through the following correction. The length dictated by the x and y positions above is divided by the physical length of the segment to find an error value, which should ideally be 1. This error value is then used to correct the positions and constrain them. For the first two links, representing the upper arm and forearm, the error

correction uses the length to the accelerometers and not to the actual joints. For subsequent segments, the length to the joints (defined as “Link”) is used.

$$\begin{aligned} \text{Length1} &= ((x_{f2}-x_{f1})^{**2}+(y_{f2}-y_{f1})^{**2})^{**}(1/2); \\ \text{error1} &= \text{Length1}/\text{Meter1}; \\ x_{f2} &= (x_{f2}-x_{f1})/\text{error1}+x_{f1}; \\ y_{f2} &= (y_{f2}-y_{f1})/\text{error1}+y_{f1}; \end{aligned}$$

The corrected position values are then used to recalculate the acceleration values, and subsequently the velocities at the end of the time step. These velocities come into the next step as initial velocities.

$$\begin{aligned} A_{x1} &= (x_{f2}-(x_{i2}+v_{xi2}*dt))/(.5*dt^{**2}); \\ A_{y1} &= (y_{f2}-(y_{i2}+v_{yi2}*dt))/(.5*dt^{**2}); \\ v_{xf2} &= v_{xi2}+A_{x1}*dt; \\ v_{yf2} &= v_{yi2}+A_{y1}*dt; \end{aligned}$$

Finally, the cosine and sine of the angle of the segment are recalculated for use in the next time step.

$$\begin{aligned} \text{CosAng1} &= x_{f2}/\text{Meter1}; \\ \text{Si nAng1} &= y_{f2}/\text{Meter1}; \end{aligned}$$

For the first two segments only, the position of the actual end of the segment is calculated using the corrected angles and length of the actual segment.

$$\begin{aligned} x_{f3} &= \text{CosAng1}*\text{Li nk1} + x_{f1}; \\ y_{f3} &= \text{Si nAng1}*\text{Li nk1} + y_{f1}; \end{aligned}$$

A special case is the correction for the wrist. Using the recalculated cosine and sine of the link 3 angle, the wrist angle is recalculated. The position of the subsequent point is then calculated using the length of the hand (LinkHN) and the recalculated wrist angle.

$$\begin{aligned} \text{CosAngW} &= \text{CosAngHN}*\text{CosAng3}-\text{Si nAngHN}*\text{Si nAng3}; \\ \text{Si nAngW} &= \text{Si nAngHN}*\text{CosAng3}+\text{Si nAng3}*\text{CosAngHN}; \\ x_{f7} &= \text{Li nkHN}*\text{CosAngW}+x_{f6}; \end{aligned}$$

$yf7 = \text{LinkHN} * \text{SinAngW} + yf6;$

## Conclusions

.Although we were unable to replicate the motion of a cast through LabView, we were successful in creating a device which could read the required amount of data for proper analysis. Our inability to properly replicate the motion was due to drift in our voltage readings which we were unable to determine the source of. It is a known issue that the room in which we were working in HL031 does experience quite a bit of noise, especially from the air conditioning system, which we suspect was the primary cause of our problems. Through tests with a single accelerometer, we know that noise is an issue regardless of the number of channels read, and most noise was able to be cancelled through the use of a moving average. The drifting, however, was less consistent than regular noise, would occur only at seemingly random intervals, and was at magnitudes great enough to appear as accelerations. This meant that our second VI was not tested with accurate readings, and is therefore unverified. Despite this, we are confident that the second VI would be able to accurately map the positions of the arm in a cast, as the plot was properly constrained and moved in the expected manner when given user-defined data. We are also satisfied with our first VI, which is able to run a moving average, zero-out steady-state data, and convert read voltages into accelerations. We believe that with further work, any obstacles would be overcome and the system would perform as intended.

## Bibliography

Durham, Wayne, *Chapter 3: Coordinate System Transformations*, Department of Aerospace and Ocean Engineering, 2004, Virginia Tech, Jan. 2007,  
<http://www.aoe.vt.edu/~durham/Ch03.pdf>

Eichele, Klaus, *Euler Angles*, 18 Jan. 2005, University of Tübingen, Jan. 2007,  
<http://casgm3.anorg.chemie.uni-tuebingen.de/klaus/nmr/conventions/euler/euler.html>

Horn, Berthold K.P. and Inoue, Hirochika, *Kinematics of the MIT-AI-VICARM Manipulator*, Artificial Intelligence Laboratory, May 1974, Massachusetts Institute of Technology, Jan. 2007,  
[http://people.csail.mit.edu/bkph/articles/Kinematics\\_Vicarm\\_WP\\_69.pdf](http://people.csail.mit.edu/bkph/articles/Kinematics_Vicarm_WP_69.pdf)