Damper Dyno

ME 495M
May 15, 2016

Jon Anderson
Ashlen Funke
Bryan Lin

Executive Summary

TBD

Force Displacement Curve

Summarize variance from manufacturers curve

Project Overview

Problem Statement

Currently, the only damper force curves we have for any of the dampers we have used over the years come straight from the manufacturer or noisy on car data. Due to manufacturing imperfections, wear, damper bleed quality, and other factors there is the potential for a large variance in damper forces for different dampers. Manufacturer documentation states that the force variance can be as high as 25% for the same settings on two different dampers. Because of this, the ability to compare force readings taken in a controlled environment would prove invaluable for making intelligent damper tuning decisions. To provide a controlled environment for such testing, our group would like to develop a simple, portable damper dyno to test and create force-velocity curves for various dampers at different settings.

Goals

The goal of this project is to design, build, and test a dyno with at least one damper and provide a real world Force-Velocity curve at various damper settings. Essentially, we'd like to replicate the manufacturer's own testing to provide more insight on how changing settings impacts the damper's performance.

Constraints

The damper dyno must be portable, stiff and easily powered. Because the ability to collect this data would be valuable at competition, our dyno must be self-contained and easily packed in the team trailer. The dyno must also be stiff. Any significant deflection would yield inaccuracies in the force-velocity relationship. The air cylinder for this dyno will be powered by either shop air or compressed air from a scuba tank. Assuming that shop air is roughly 80 psi, the piston must be sized such that the maximum working load can be applied to the damper and that air pressure. Designing to a lower air pressure will prevent us from unnecessarily having to use compressed air tanks when performing testing in our own pit.

Opportunities

The damper dyno will be large enough to accommodate the Penske 7800 dampers in addition to the Ohlins TTX25 dampers so that future groups would have the option to test either. The structure will also be designed to support the much higher loads that might be imposed by testing springs. This will allow the future option to retrofit the machine to test springs, without creating any structural concerns.

Functional Requirements

Load case

The fundamental load cases were derived from two sources: on track data, and industry standard testing practices. The T26 cCar was fitted with both damper potentiometers and corner load cells at FSAE Lincoln. This allowed us to derive a data channel displaying damper force. This was done by treating the damper and spring as if they were in series and subtracting the spring force from the total force on the damper/spring combination. The spring force was derived from knowing the spring deflection from the potentiometer measurements as well as the known spring rate, while the the total force was collected at the load cell. By looking at both autocross and endurance data from all four corners, a maximum competition load case of 117.66 lb at 9.99 in./sec.

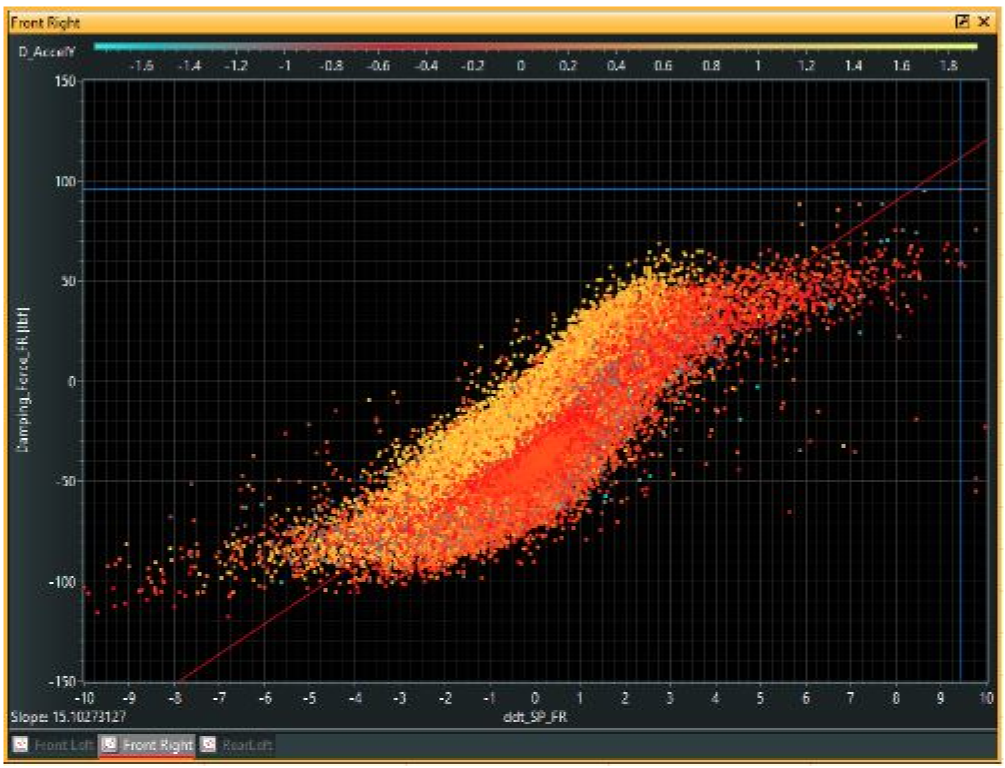


Figure XX: Damping force vs. velocity data from FSAE Lincoln endurance

By looking at documentation from Roehrig, a major damper dyno manufacturer, we were able to determine industry standard testing practices. Commonly, the maximum damper velocity tested is 10 in./sec. in both the compression and rebound directions. From this, we determined that would be the maximum tested velocity for our dyno. Comparing this data to damper curves for provided by Ohlins, the TTX25 MkII dampers see a maximum damping force of 367.2 lb.

Test Profile

As previously mentioned, the test profile was derived by researching documentation from Roehrig. The most common industry standard profile is a Continuous Velocity Plot (CVP) at 10 in./sec. A CVP sweeps a sinusoidal velocity profile from the maximum negative (compression) velocity to the maximum positive (rebound) velocity and back again. Generally, this sweep reaches a maximum positional amplitude of +/- 1". During this sweep, velocity and force is recorded throughout, creating a continuous damper curve containing force outputs for every possible velocity input from the minimum compression velocity to the maximum rebound. A continuous velocity sweep greatly simplifies the control scheme as it is not necessary to produce the exact finite velocity at varying damper settings.

Mechanical System Design

Figure xxx: Damper Dyno Mechanical Components

The current design consists of a large aluminum base and a pneumatic actuator to guide and oscillate the damper. From the manufacturer's data and T26 endurance data we can expect see a max force of a little over 100lb and a max velocity of 10 in/sec, which seems well within reason of a pneumatic system operated off of shop air. Hose lines connect the two ports of the actuator to a solenoid valve which allows air from the source in and exhausts air based on command codes. An Arduino microcontroller will be used to control the valve position as a function of time. A load cell in series with the damper will measure the applied force and a linear potentiometer in parallel with the damper measures displacement. The data we acquire from these sensors can be plotted as a function of force and velocity to be compared with the manufacturer's curves for the same setting.

Air Cylinder

The air cylinder was selected using three major requirements: maximum force, maximum velocity, and maximum stroke. We are designing our dyno to be utilized running off shop air, which generally falls around 120 psi, meaning we must be able to produce a maximum velocity of 10 in./sec. and a maximum force of 367.2 lb at that pressure range. At 120 psi, a 2 in. diameter cylinder produces 377 lb of force. At that same 120 psi, a 2 in. diameter cylinder with a ¼” National Pipe Thread (NPT) input can see a maximum flow rate of 786 in./sec. Both of these numbers fall well above our design requirements, and can be supplemented by running the cylinder off of a high pressure SCUBA tank. All 2 in. diameter cylinders produced by Bimba are rated to a maximum of 250 psi which corresponds to a maximum possible force of 785 lb.

The maximum damper stroke of a Penske 7800 shock is 2.75 in. In order to allow for the fitment of the Penske 7800 shock as well as the shorter Ohlins TTX25, 3 in. of cylinder stroke is required. The final selection requirements are a 2 in. diameter cylinder w/ 3 in. of stroke and a ¼” NPT inlet. From these requirements, a Bimba SR-313-.5-D cylinder was selected. This is a standard double action “Original Line” air cylinder which is constructed entirely of stainless steel, allowing us to use the dyno in outdoor testing facilities without fear of corrosion.

Pneumatic Solenoid

The solenoid valve was decided based off the the capabilities of the air cylinder. We wanted to air cylinder to be the dyno’s limiting factor and not the solenoid valve. For the air lines to not be the bottleneck we determined that ⅜” tubing was appropriate. The valve we settled on was a 4-Way, 3 Position, Center Exhaust Valve. A center exhaust design was chosen because it allows both sides of the piston in the air cylinder to be exhausted to allow the damper to return to its resting position. This will ensure that the dyno will have a consistent start point whenever it begins running.

The initial plan was to use PWM to control the valve which would allow us to use a constant pressure source between runs and allow us to quickly flutter the valve to control pressure. However, upon assembling the system, we found that the 50ms response time and 5hz actuation frequency was way to slow to make PWM possible. All the solenoids in this size range seem to have a limit to their switching speed that is not easily or cheaply overcome. The best solution to this would have been to run multiple solenoids in parallel to give us the faster switching speed and provide the flow we needed. However, to do so would compromise the simplicity of system and increase the cost dramatically.

Data Acquisition

The Futek load cell will be mounted in series with the damper and will output force data with respect to time. The CLP-75 potentiometer will be mounted in parallel with the damper and will output position data with respect to time. Data will be extrapolated using the Arduino to log the force vs. velocity curve.

Structural Components

The main structural components for the damper dyno are an aluminum base, damper mounting bracket, damper interface and load cell. All structural components were designed to withstand cyclical loads of 600lb applied by the piston. Analyzed as a whole, the structure has an approximate axial stiffness of 2.310^5 lb/in during the compression stroke and 1.9x10^5 during the rebound stroke. At the estimated applied load of 100lb the deflection of the system will be less than 0.001” meaning that deflection in the structure will not have a significant effect on our Force-Velocity curve results.

Figure xxxx: Dyno Base

The base was made from 7050 Aluminum bar stock. The air cylinder mounts directly to the base at the tapped hole indicated on the diagram. At this threaded joint, the air cylinder will transfer the reaction force, opposite the direction of piston travel, to the base. The mounting bracket for the damper mounts inside of the box at the indicated location. At this interface the reaction force from the damper, in the direction of piston travel, will be transferred into the base. In theory, the base will be reacting either tensile or compressive forces at each end, that are equal in magnitude and opposite in direction.

Figure xxx: Load Cell Adapter

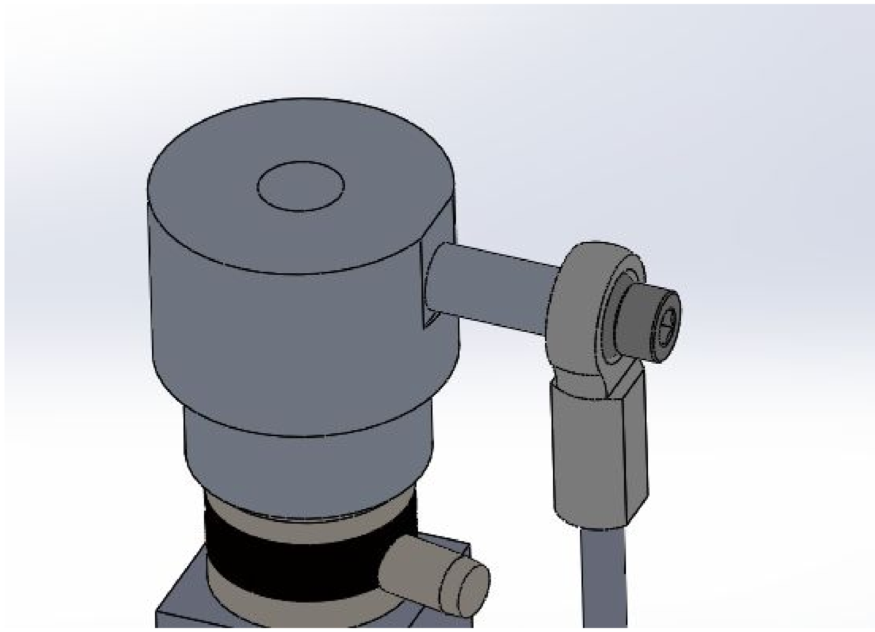


Figure xxxx: Load Cell Adapter / Linear Potentiometer

The load cell adapter was made out of 7075 aluminum round stock. Its function is to transfer load from the threaded end of the piston to the threaded end of the Futek load cell. The internally threaded ends of this part experience only axial tension and compression. The load transfer capacity of these threaded connections far exceeds the functional requirements of this dyno. The linear potentiometer is fastened to this component at a through an M5 helicoil hole on the side of the part.

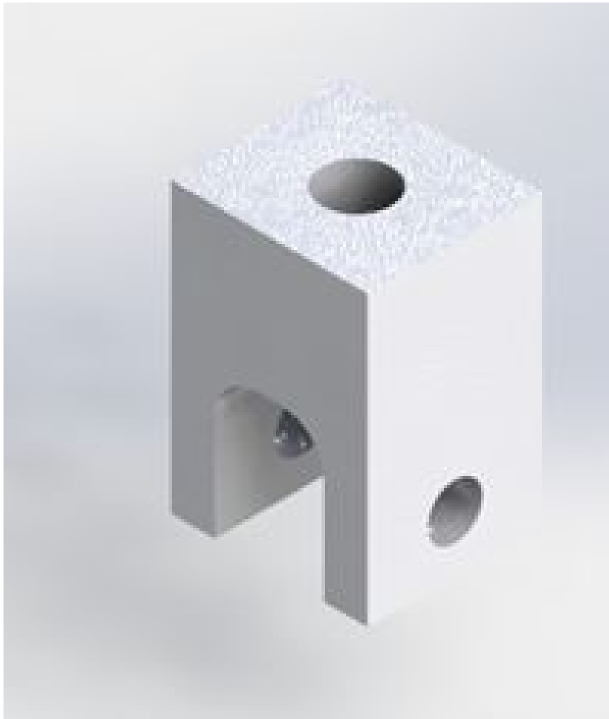


Figure xxxx: Damper Interface

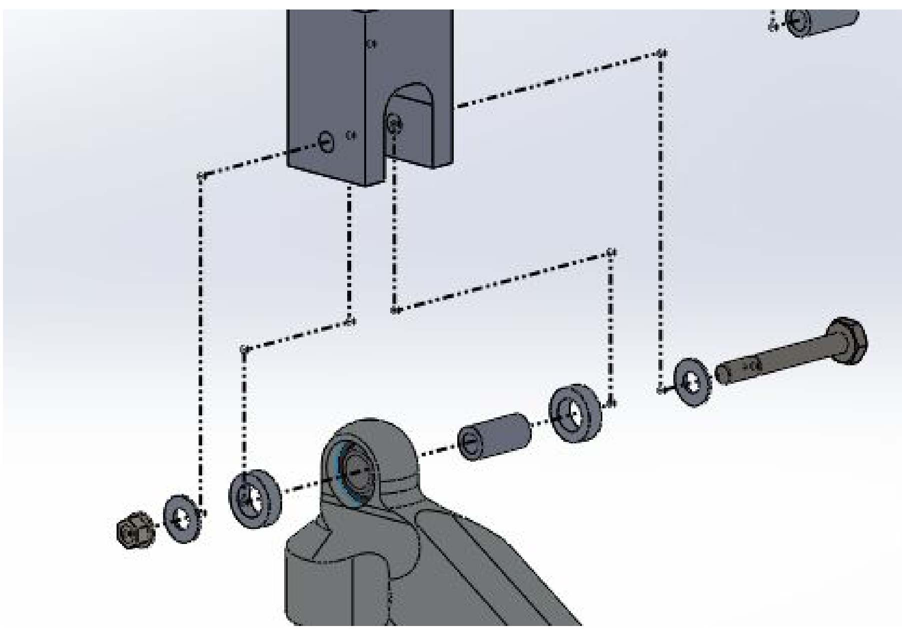


Figure xxxx: Damper Interface / Damper

The Damper Interface is made out of 7050 aluminum bar stock. The purpose of this component is to transfer load from the load cell to the damper. Similar to the load cell interface, the connection between the damper interface and the load cell is threaded. The clevis at the bottom of the damper interface bolts to the damper and transfers load to the damper in bearing. The mounting hardware are an AN3 bolt, three spacers, and a K-nut.

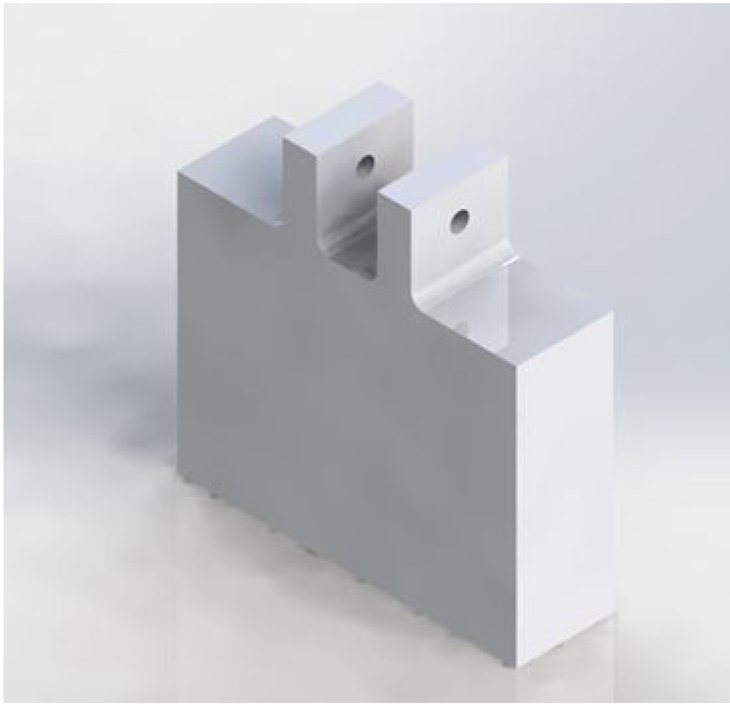


Figure xxxx: Mounting Bracket

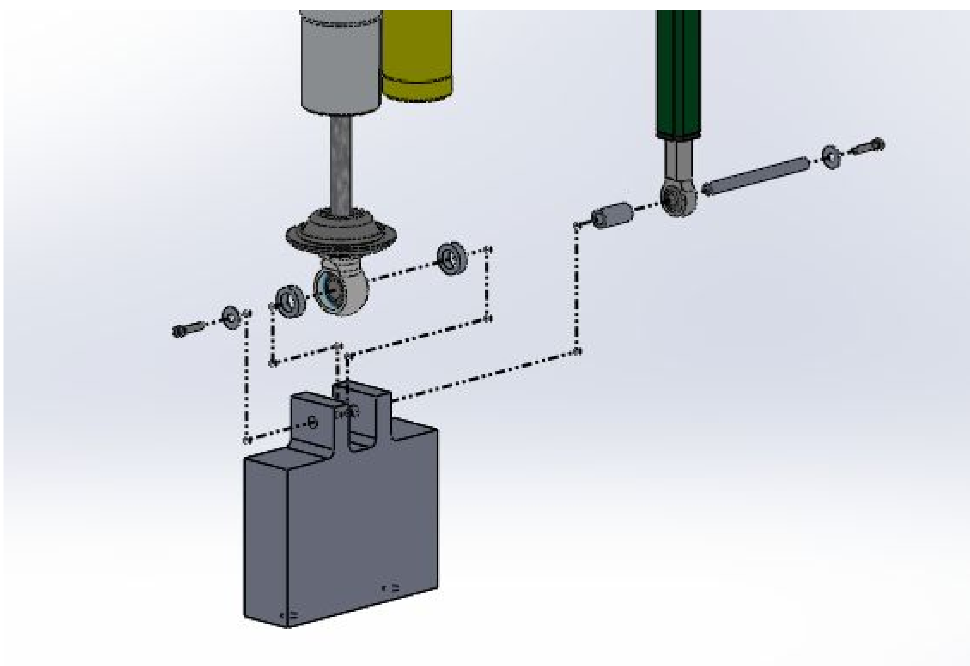


Figure xxxx: Mounting Bracket / Damper / Linear Potentiometer

The mounting bracket is made out of 7050 aluminum bar stock. The purpose of this component is to transfer load from the damper at the clevis end of the bracket into the base through threaded connections. A 5mm aluminum rod passes through the potentiometer, mounting bracket and damper. It is retained by a washer and 4-40 screw at each end. There are also four spacers that locate the damper and potentiometer. We used the aluminum rod instead of an M5 bolt because of the unusually long shank required for this application, and the precision fit required in this joint.

Control System Design

Bryan

System Structure

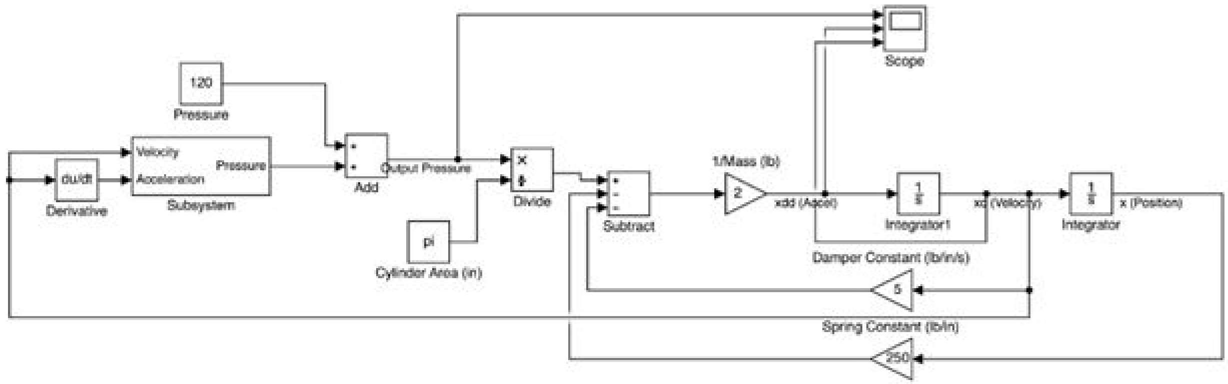


Figure xxxx: Simulink System Model

The system structure shown here was an early model of how it was imagined that the system would work. The system on the left symbolizes the inputs back into the Arduino. One difference here is that the force generated by a load cell and from a derivative of velocity and the mass of the system.

Theoretical System Response

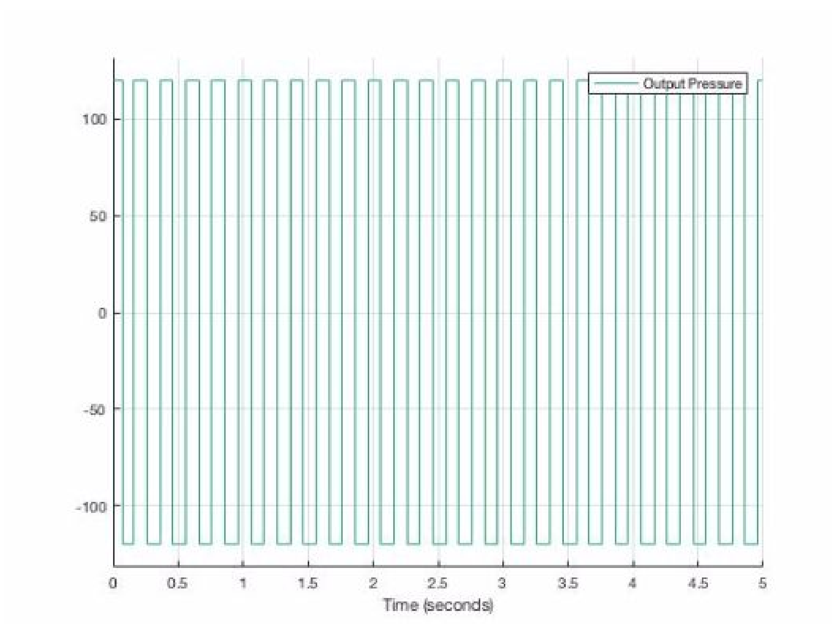


Figure 1: Output Pressure

The pressure curve shown here is idealized but clearly shows what we wanted the behavior to be generally. In the real application there is a lag time between the compression and rebound solenoids from the actuation of the valves as well as the time it takes to vent the pressure.

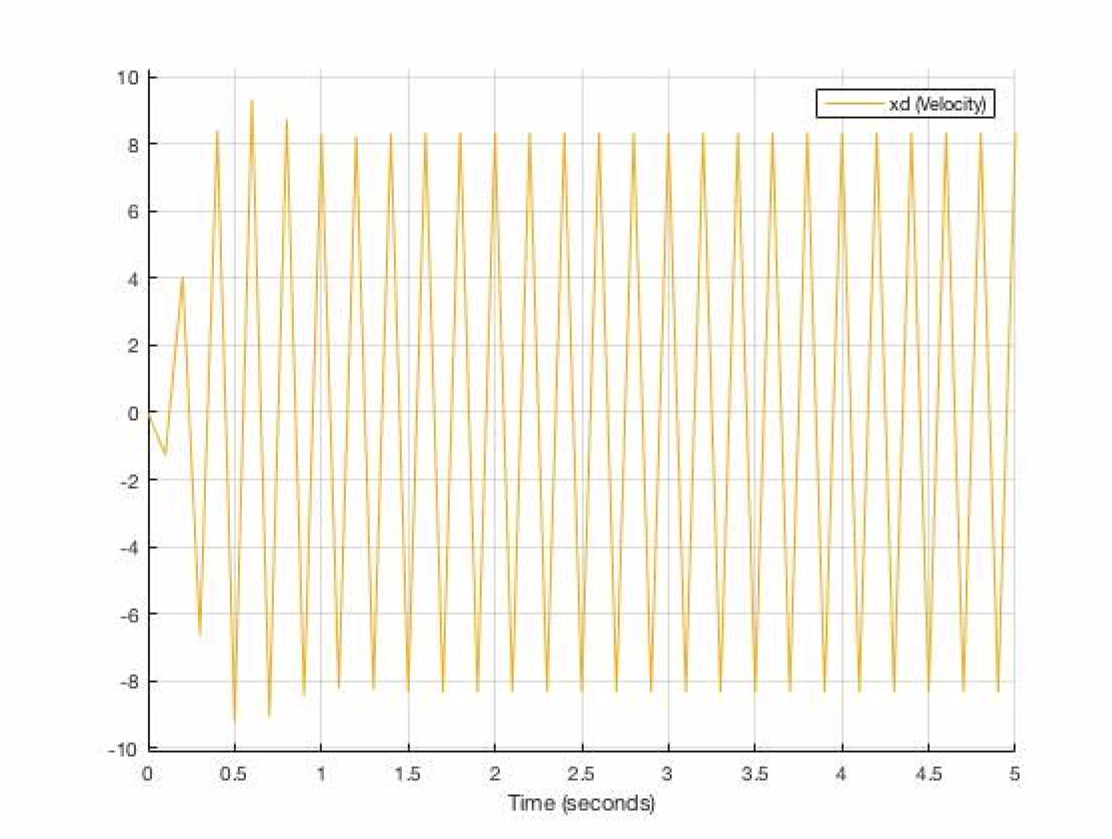


Figure 2: Velocity

This is what the velocity plot was expected to look like based on the pressure input to the system. The velocity comes to very sharp points because the system assume no switching lag which is not true.

Controls Hardware

The solenoids are controlled by a 5V digital output from the Arduino unit. The 5V output is used to switch a MOSFET which would allow for PWM control by the Arduino. The Arduino will monitor the velocity of the damper using the potentiometer. When the velocity of the damper has a magnitude greater than 10 in/s the solenoids will actuate and apply opposing pressure to the opposite side of the piston in the air cylinder. This will slow down the damper and move it in the opposite direction until it reaches 10 in/s again.

Testing Results

Initial testing encountered issues with the logging frequency of the arduino. This has partially been overcome by storing the desired information in an array until the entire run is complete. This limits the number of samples that can be taken because of the arduino's limited memory, but is less processing intensive than continually printing to the serial output. Currently, the log rate can go as low as 333.33 Hz. This is not quite ideal because of the logged velocities, only two or three values are captured before the velocity exceeds 10 in/s. Another issue we ran into is that the solenoid are not very quick. They have a 50ms response time. This is a result of their size and the flow rate they are designed for. Because of the slow response time, PWM is not really an option. To get around the problem, a pressure regulator was used instead. Another issue caused by the slow response is that even though the solenoids are switched when a velocity above 10 in/s is logged the velocity continues to shoot up to 30 or 40 in/s before starting to reverse.This is also partially due to a logging frequency that is too low because there are few values before tripping the system to switch cylinder pressure side. In addition, calibrating the load cell has proven a challenge. The load cell is designed for up to 3000lb but we realistically only need up to 500lb. Currently we are not sure if the limited resolution of 1023 divisions on an arduino analog input can be mapped to 500lb and below. If it is not possible, the resolution would be very poor at around 3lb. This could be solved through the strain gage amplifier setting or arduino code. The upper potentiometer mount also has the tendency to rotate which introduces some inaccuracy in the readings. This could be alleviated through a support that would keep it from rotating. The position accuracy of the arduino is .003" which is not ideal, but usable. This is another limitation caused by the arduino's 1023 division limit. The linear potentiometer acts as a voltage divider so the divisions cannot be remapped to a narrower range unfortunately. Most of the testing runs were done at 10 or 20 psi. Both proved to be sufficient at the damper's softest compression and rebound settings. A lower pressure was chosen for testing to not damage any dampers in case they were bottomed out or topped out.

The strain gage amplifier was reprogrammed to the highest gain value and the offset tuned so that it the load cell outputted 2.5V at rest. Even with these settings, the arduino was only able to resolve a resolution of about 1lb increments.

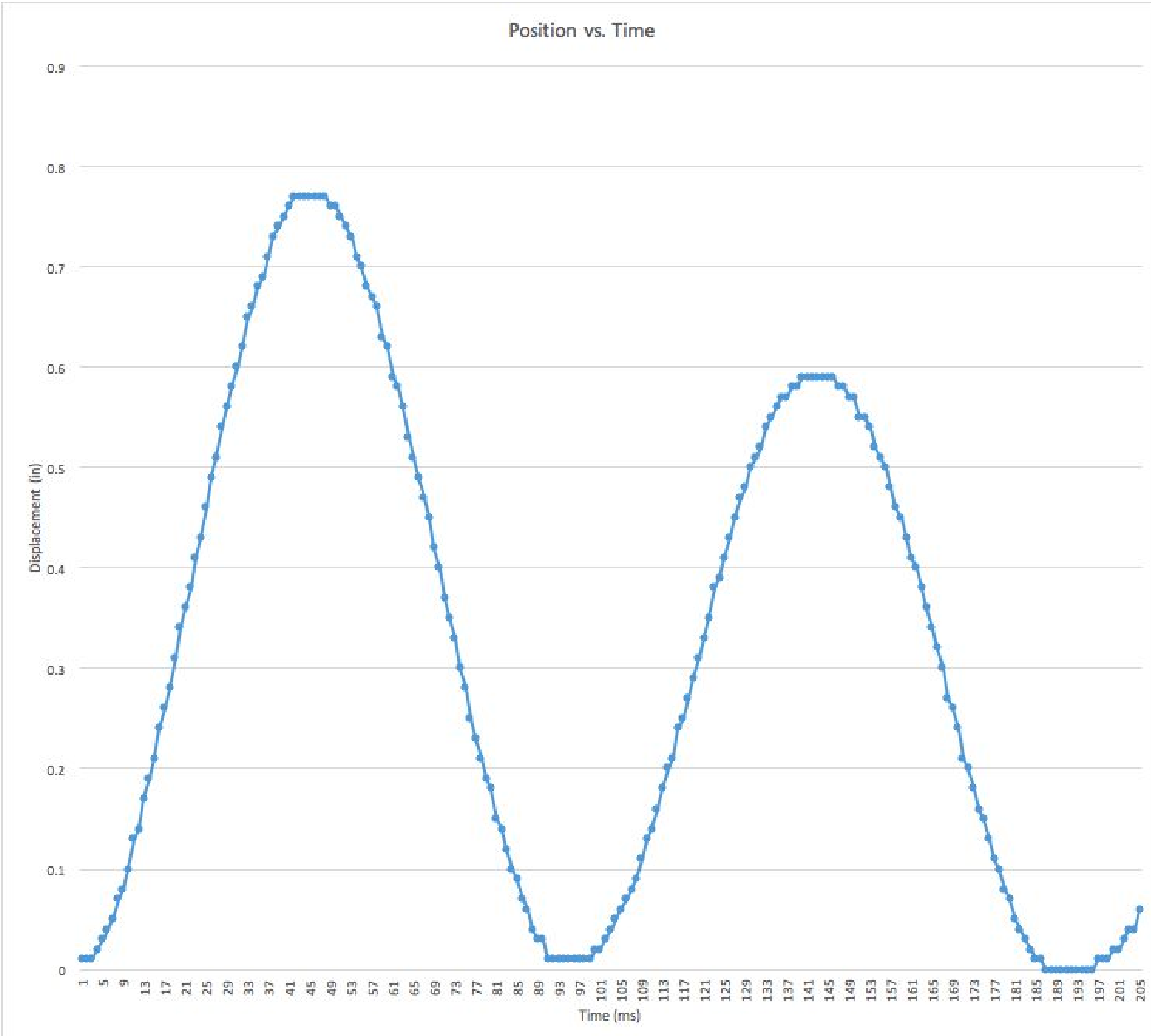


Figure 2: Position vs. Time Graph

This was a plot generated from a standard sweep in order to investigate noise from the linear potentiometer data. It is also exciting to notice that the curve generated matches that of what we wanted based on our logic. The decreasing amplitude can be attributed to the pressure of the air supply settling. At initial actuation the pressure drops. After the pneumatic solenoid reaches steady-state flow the pressure stabilizes. An air capacitor would help solve this, but the drop in pressure is not harmful to the data since it stabilizes quickly. Although, the overall data is clean and follows a sinusoidal path as desired a closer look at the curve reveals small amounts of non-smooth data. Even though the deviation is small, usually one division, the effect is magnified when velocity is being calculated. See Figure 3 of the velocity plot.

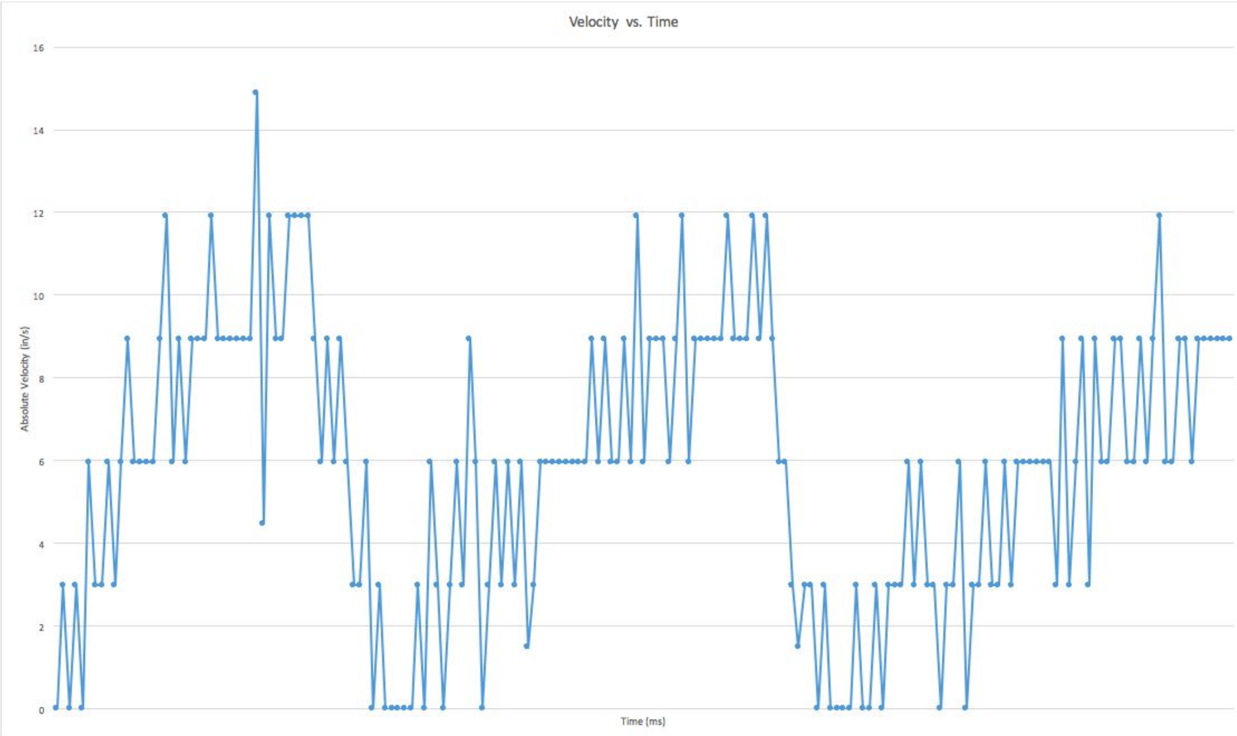


Figure 3: Velocity Plot

The noise while the system was not moving which equates to about .003”. When applied to a velocity calculation that noise results in a change in velocity of about 3 in/s which is a result of the conversion from millisecond units to the seconds for the velocity to be standard to the test. Which is significant when compared to the testing range of 0 to 10 in/s. The instability of the reading also made it difficult for the control logic to work because the solenoids are driven off of the velocity. The velocity could prematurely jump beyond 10 in/s which would trigger the solenoids to switch from compression to rebound or vice versa. This could cause the sweep to be cut short. An RC circuit was wired into the linear potentiometer analog input in an attempt to absorb the noise however, this did not work. This left us to speculate that the noise is a result of Arduino internals and is why this method did not work. In order, to solve this problem a better Analog-to-Digital Converter (ADC) than the one on the Arduino would have to be used. The Arduino’s ADC is a 8-bit one which is the source of the 1023 division resolution it has on its analog inputs. A 10-bit ADC would provide 4095 division over the 5V output of the sensors. This would likely solve most of our issues and allow us to generate a more accurate force vs. velocity curve.

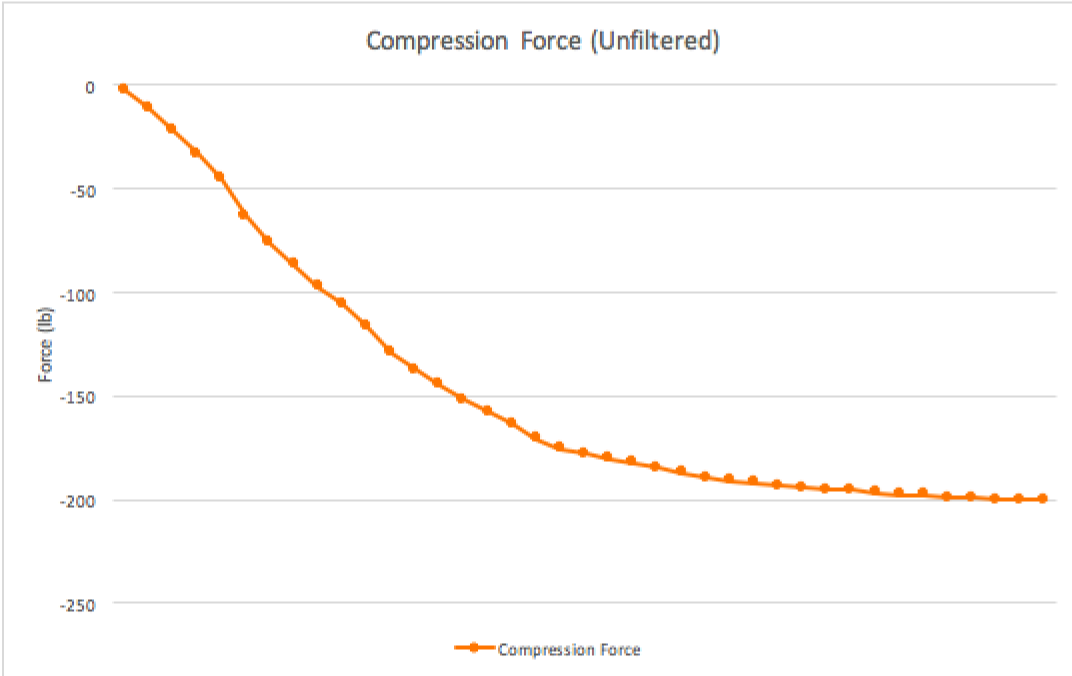


Figure 3: Compression Forces

Figure 3 shows the force data plotted in chronological order of how it was captured in one compression stroke. There is no horizontal axis labels because the velocity measurements taken were do not make much sense due to the linear potentiometer noise issue. The time elapsed between each dot is equal to 1ms in time. The plot shows that the force data captured is clean and smooth. The same smoothness can be shown with the subsequent rebound force to the compression stroke shown in Figure 3.

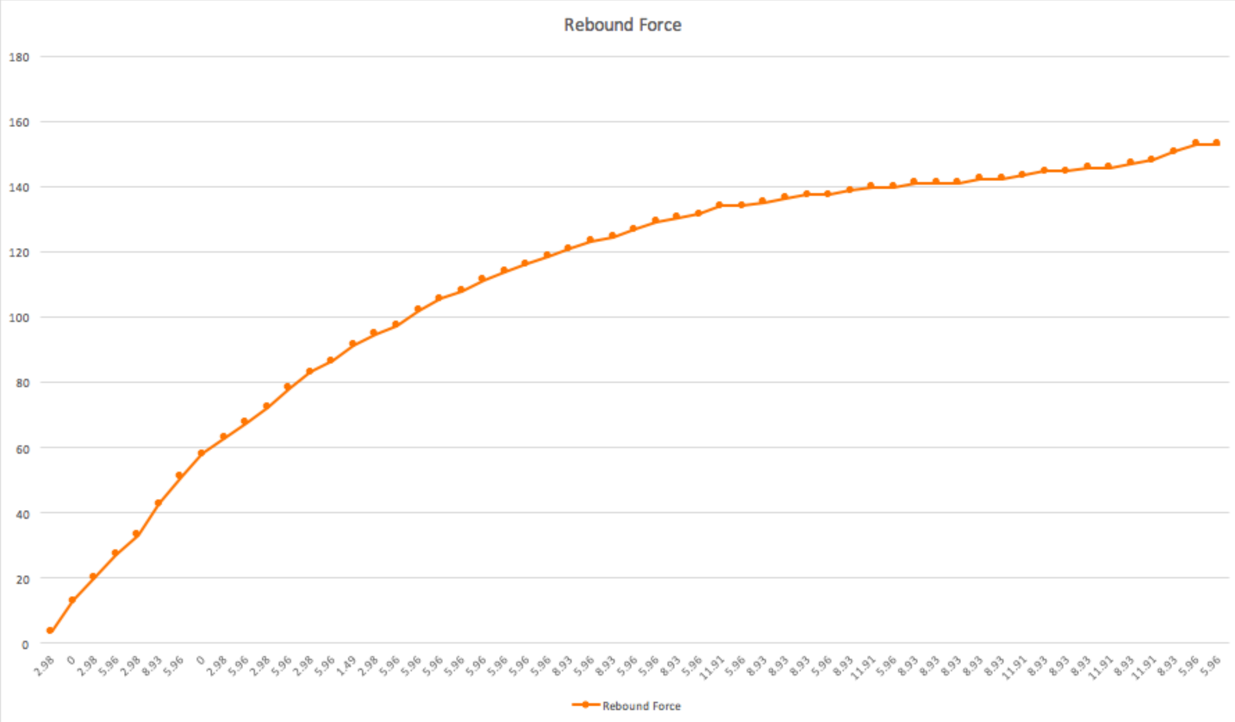


Figure 4: Rebound Force

0	-1.185
0	-3.556
0	-8.298
2.98	-11.854
2.98	-16.596
2.98	-20.152
5.96	-24.893
8.93	-27.264
11.91	-28.45
14.89	-29.635
17.87	-30.82
20.85	-33.191
23.82	-36.747
23.82	-37.933
38.71	-40.304
23.82	-42.674
29.78	-43.86
26.8	-43.86
29.78	-45.045
29.78	-46.231
29.78	-46.231
38.71	-47.416
32.76	-47.416
29.78	-47.416
29.78	-47.416
29.78	-48.601
29.78	-48.601
29.78	-48.601

Figure 4: Snippet of Compression Stroke Data (Soft)

Figure 4 shows how the velocity data makes the force data not very useful. The velocity data jumps up and down even as force continues to rise. What is expected is that the damping is a function of velocity and as velocity increases the damping should increase along with it. That cannot be shown with the data because of its limited resolution and the instability of the measurements.

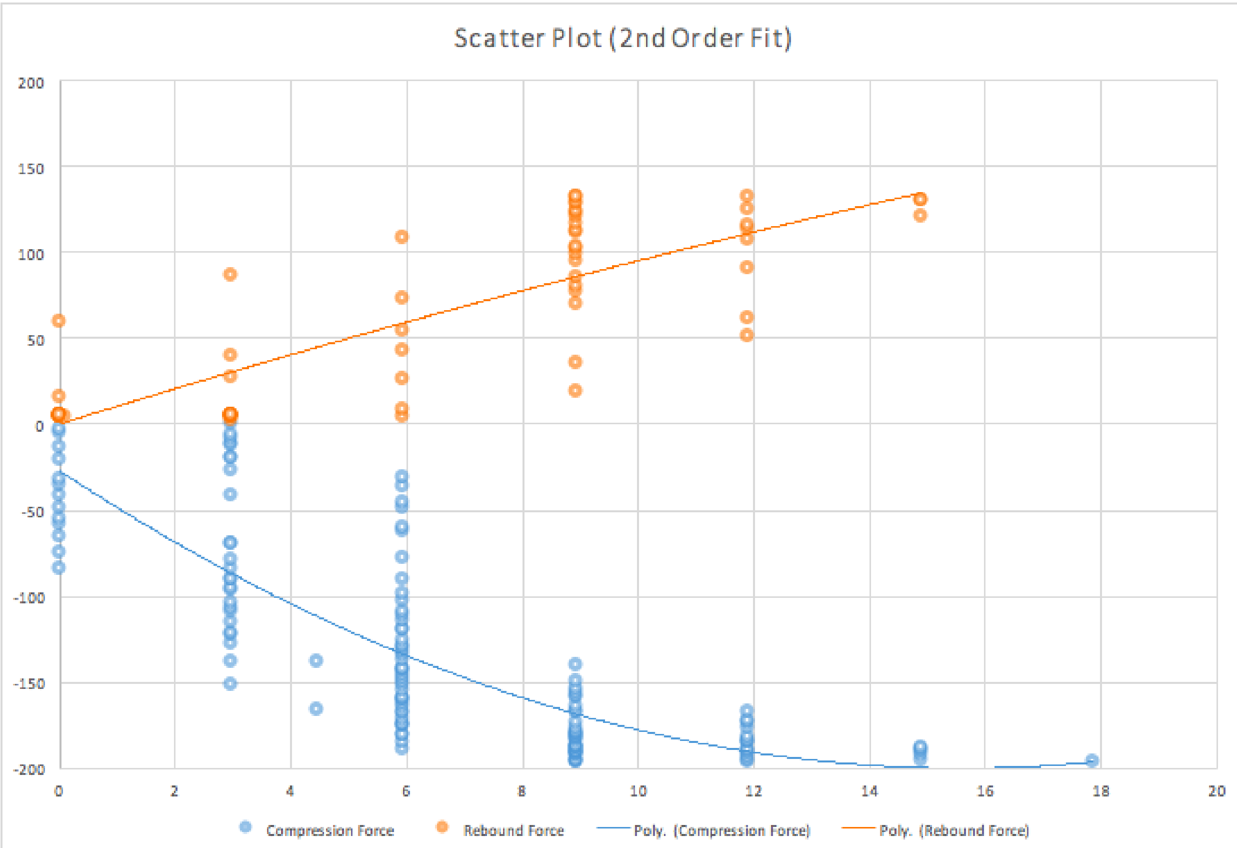


Figure 3: Data Distribution (Stiff Setting)

This plot was created to see if the majority of force measurements for a given velocity were correct. While looking at the force vs. velocity plots, the compression stroke is defined as the negative forces and the rebound being the positive side. Although, many of the force measurements are accurate the ratio is not great enough partially due to the sampling frequency being lower than ideal. This made it not possible to do a best fit line to try to generate an accurate dyno plot.

In an attempt to work around the noisy data we implemented an air capacitor on the compression side. In effect the air capacitor would delay the time it takes the cylinder to reach full pressure. This increases the time that damper spends sweeping up velocity which allows more data points to be gathered in that range. Since more data points are being captured, the outliers do not have as large an effect on the data and a more accurate curve can be generated. Figure 4 shows a force vs. velocity plot with the air capacitor attached to the compression line.

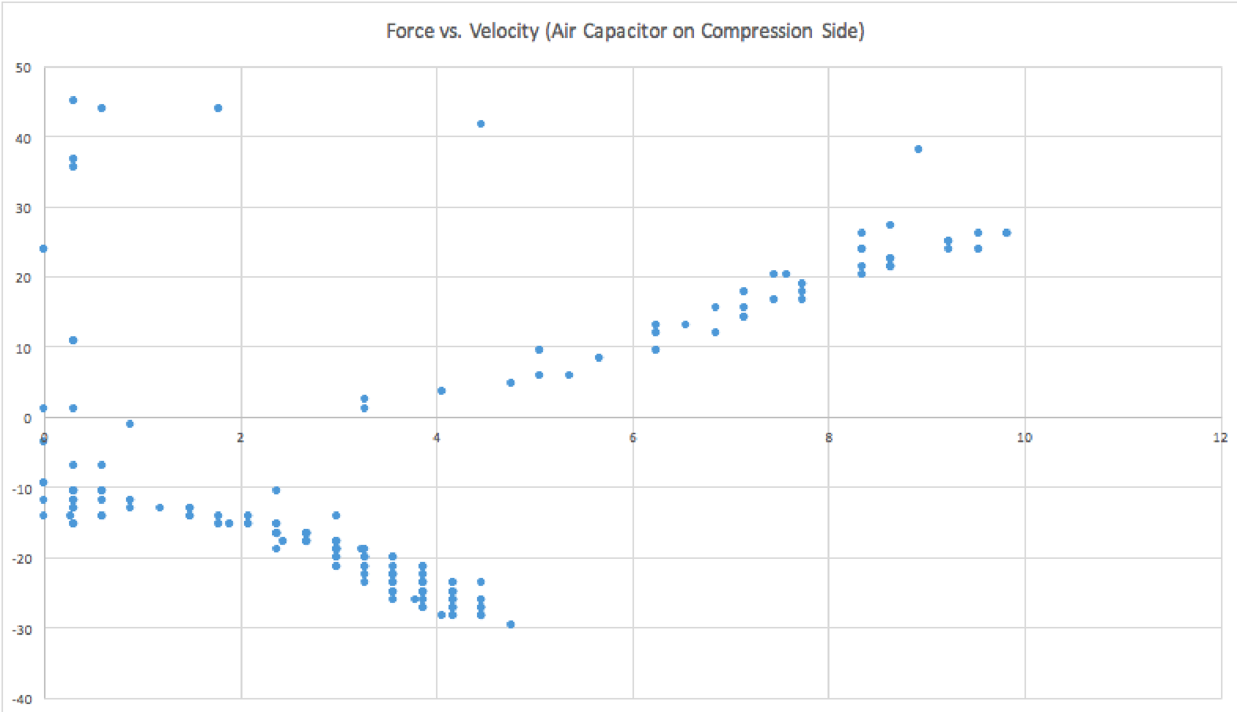


Figure 4: Force vs. Velocity Curve (Air Capacitor on Compression Side)

It can be seen that the compression stroke has a more defined plot than the rebound side.

Future Recommendations

References

TBD

APPENDIX 1: Damper Characteristics

A1.1 External dimensions

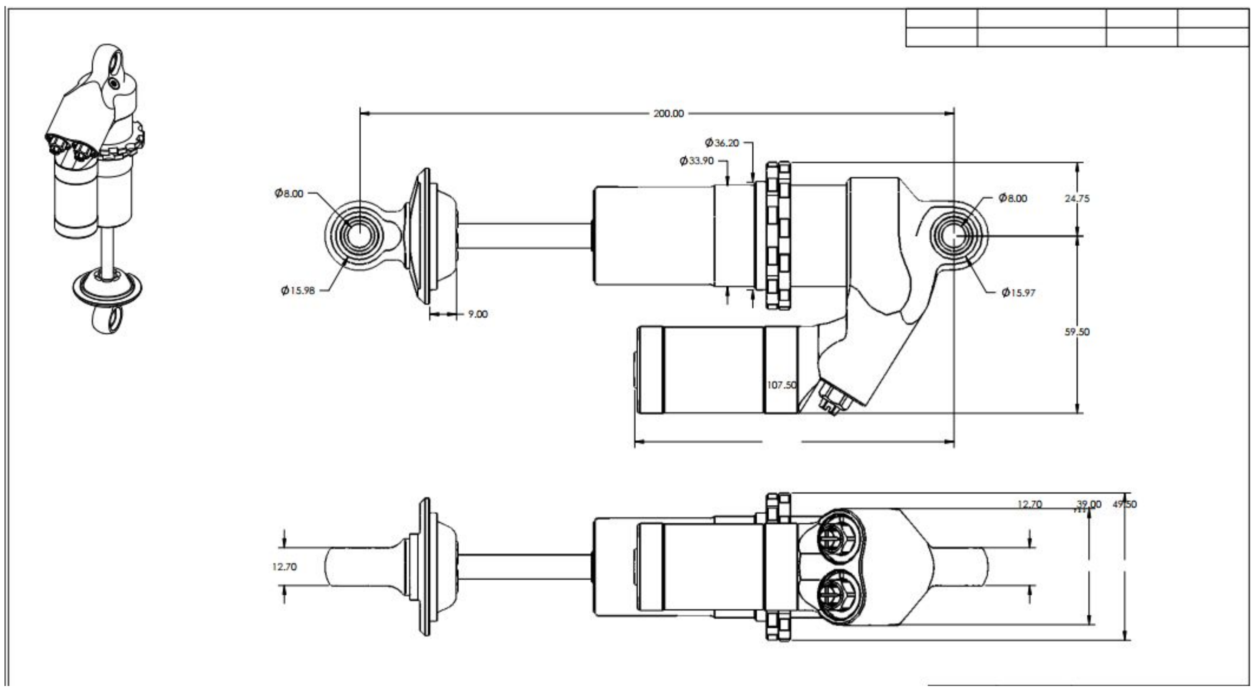


Figure xxxx: Damper External Dimensions

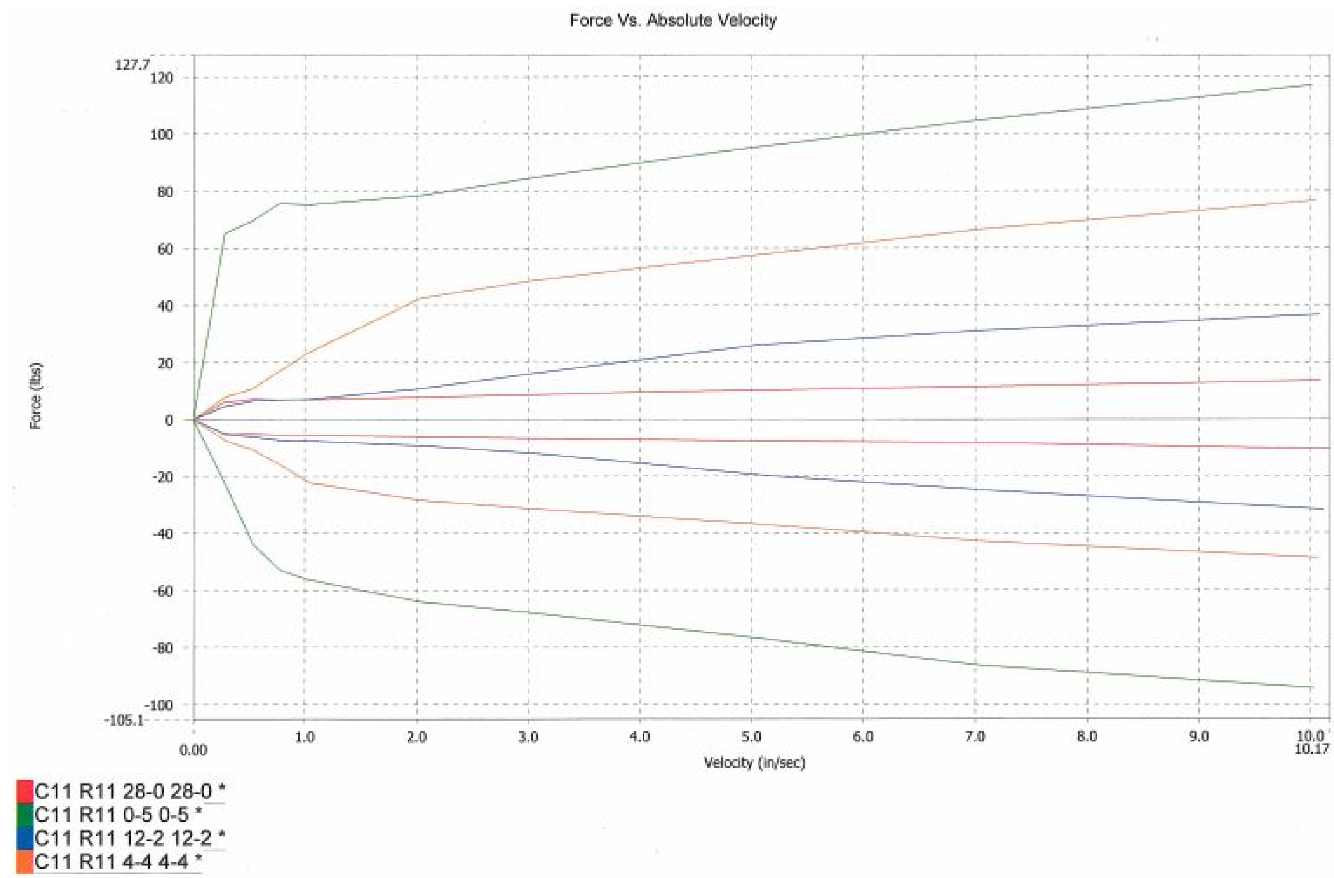
A1.2 Specifications

Table xx: Damper Specifications

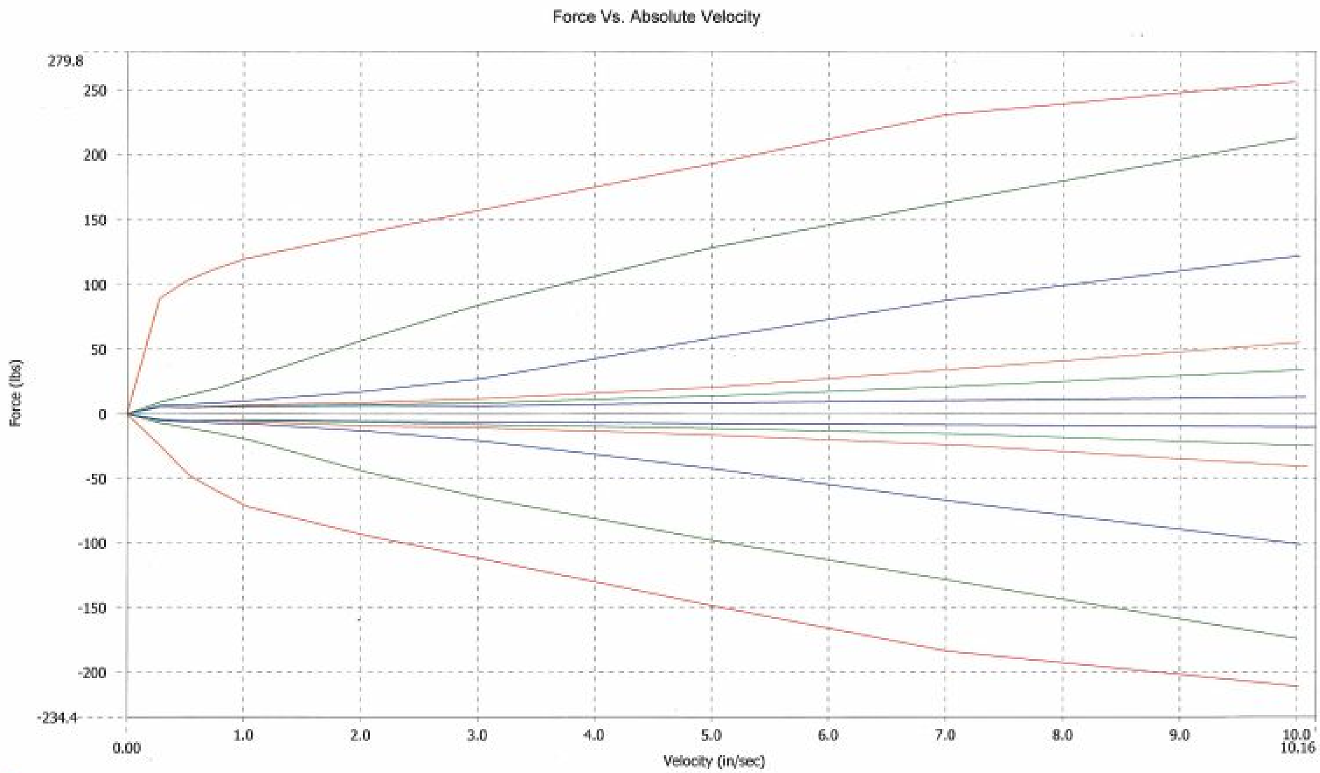
Overall length = 200 or 267mm (center to center of spherical bearings, fully extended)
Stroke = 57 or 90mm
Weight = 57mm stroke = 394g without spring, 90mm stroke = 446 g without spring
Spherical Bearing dimensions:
ID = 8 mm

Ball Width = 8 mm
OD = 15 mm

A1.3 Manufacturer’s damper curves

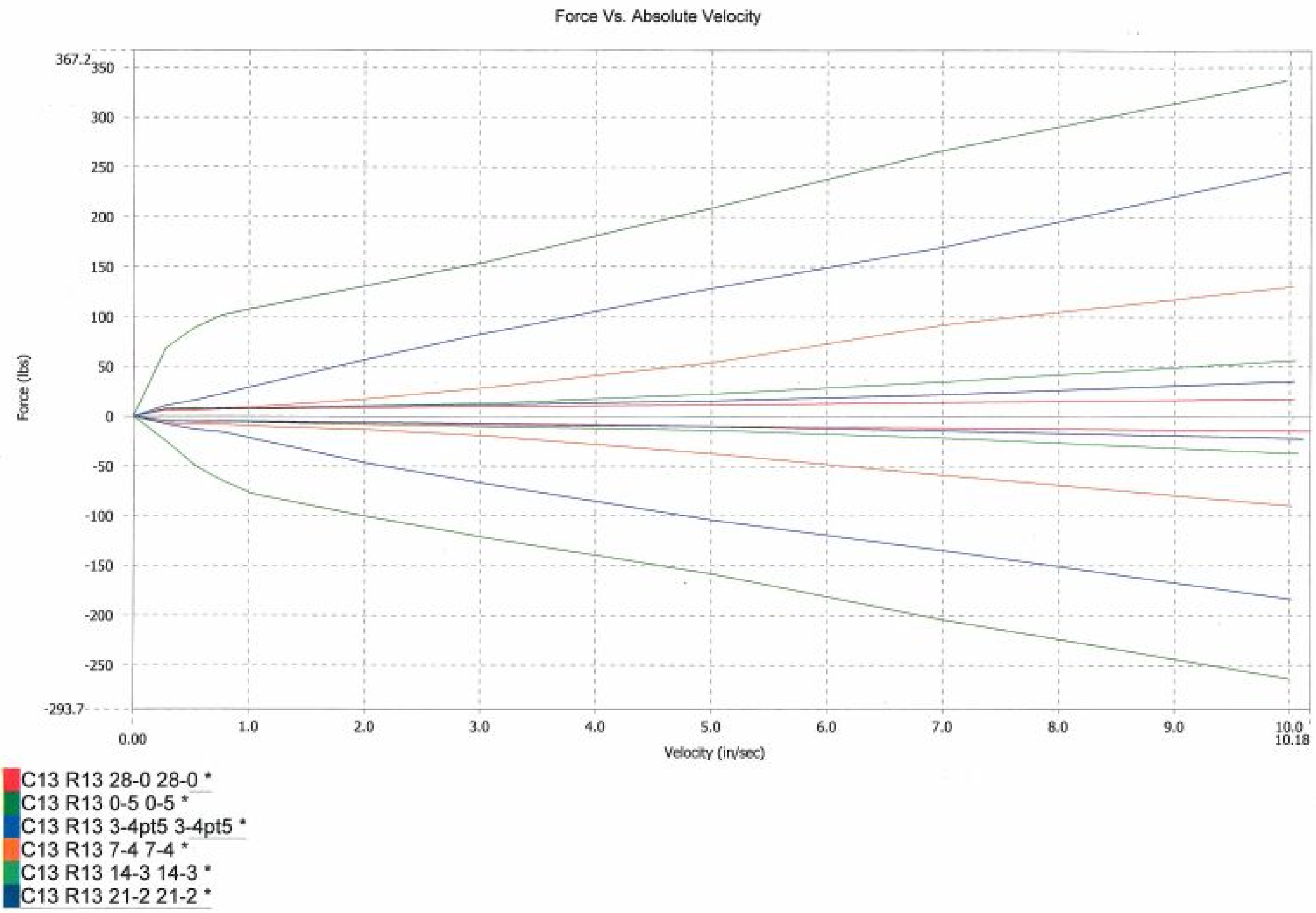


TTX 25MkII FSAE - (LSC-HSC LSR-HSR) Low speed clicks counted from fully closed (clockwise) - High speed turns counted from fully open (counter clockwise)

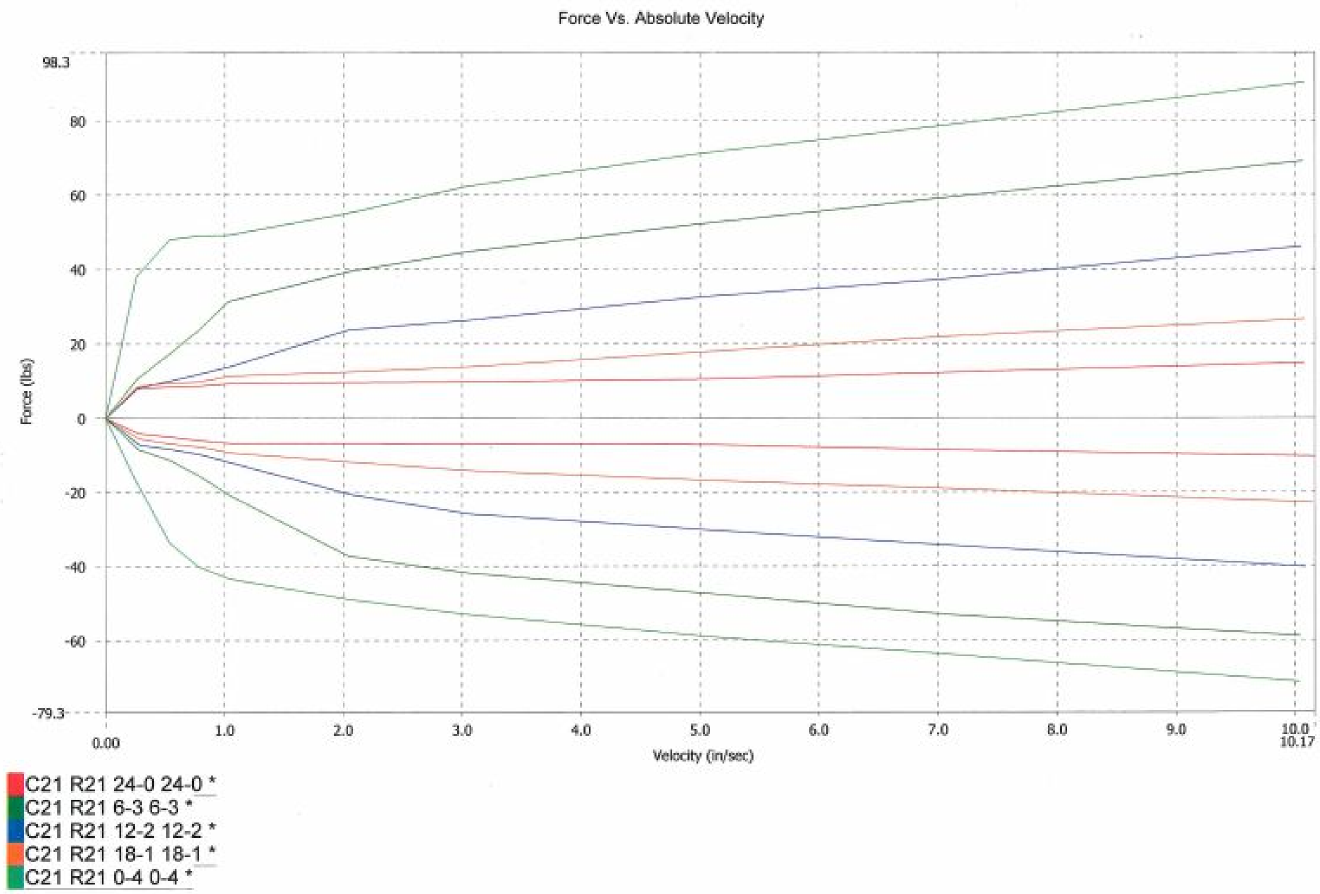


C12 R12 0-5 0-5 *
C12 R12 4-4pt5 4-4pt5 *
C12 R12 7-4 7-4 *
C12 R12 14-3 14-3 *
C12 R12 21-2 21-2 *
C12 R12 28-0 28-0 *

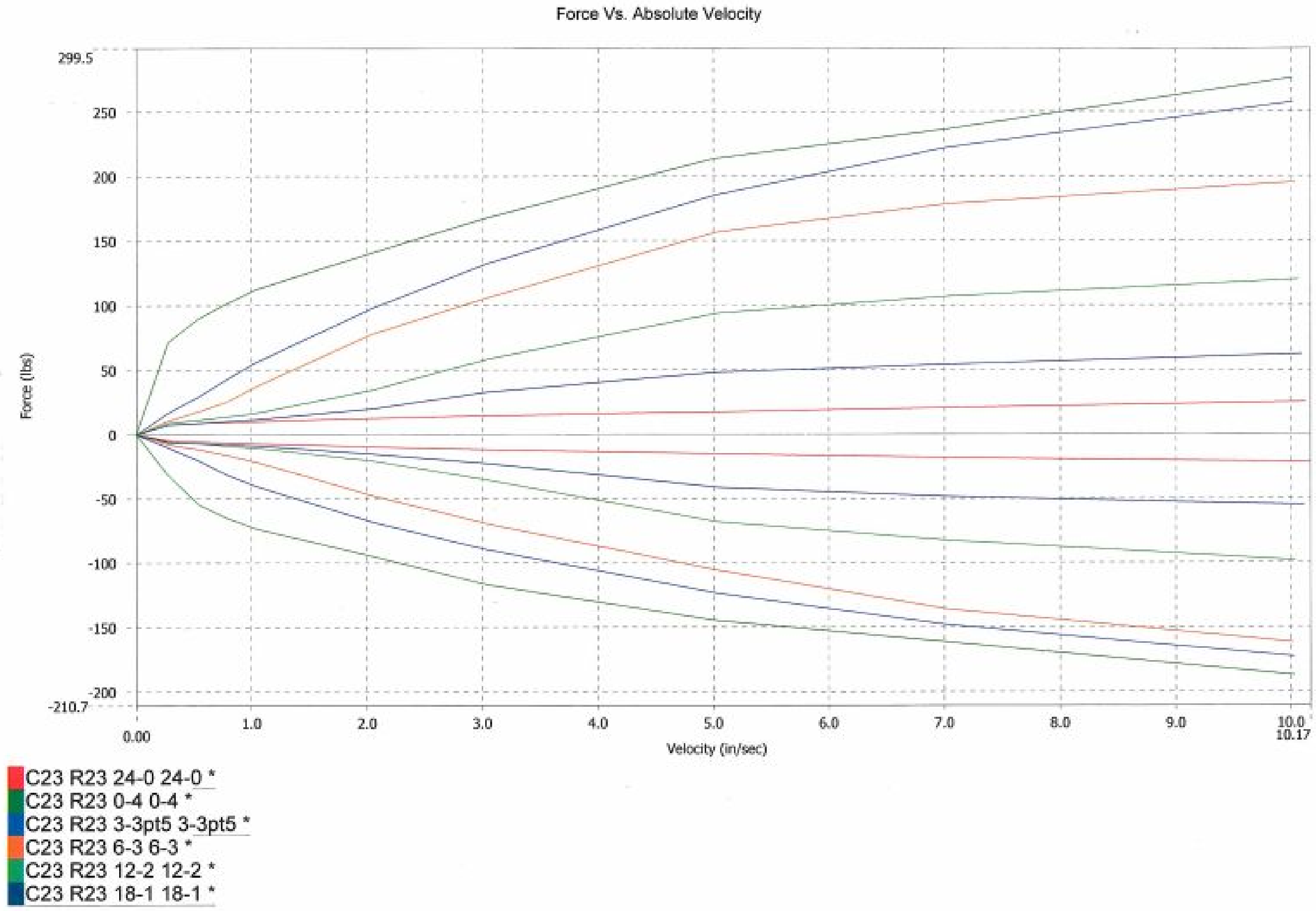
TTX 25MkII FSAE - (LSC-HSC LSR-HSR) Low speed clicks counted from fully closed (clockwise) - High speed turns counted from fully open (counter clockwise)



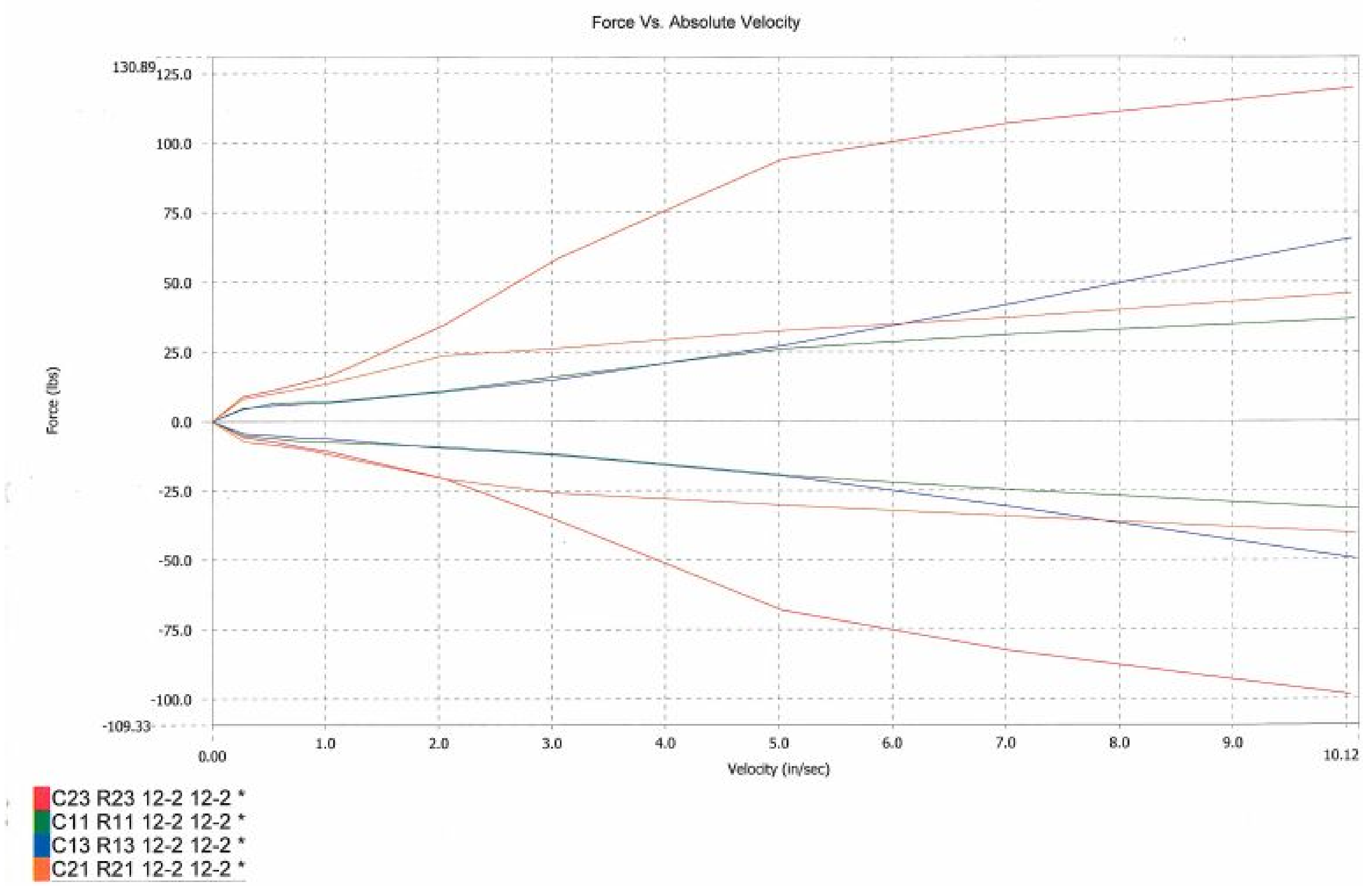
TTX 25MkII FSAE - (LSC-HSC LSR-HSR) Low speed clicks counted from fully closed (clockwise) - High speed turns counted from fully open (counter clockwise)



TTX 25MkII FSAE - (LSC-HSC LSR-HSR) Low speed clicks counted from fully closed (clockwise) - High speed turns counted from fully open (counter clockwise)



TTX 25MkII FSAE - (LSC-HSC LSR-HSR) Low speed clicks counted from fully closed (clockwise) - High speed turns counted from fully open (counter clockwise)



TTX 25MkII FSAE - Valving options (Stock Valving = C12 R12)

APPENDIX 2: Mathematical Modeling

In order to properly select components, a simple mathematical model was developed using estimations of the prototype's physical characteristics.

A2.1 Flow rate / Applied Force Calculations

$$F = Pressure * Area$$
$$\dot{Q} = Area * Velocity$$

Table xx: Target flow rate and pressure calculations

	Area (in^2)	Target force	Target Velocity (in/s)	Target Flow rate (in^3/s)	Target Pressure (psi)	Target Flow rate (liter/min)
Compression	3.14	102.13	9.94	31.23	32.51	30.70
Rebound	3.13	108.81	9.71	30.39	34.77	29.88

A2.2 Ultimate Loading Hand Calculations

$$Sigma_{threads,\ tension} = Force * Area_{tensile\ threads}$$
$$Factor\ of\ Safety = S_{yield} / Sigma_{threads,\ tension}$$

Table xxx: factor of safety, threaded connections loaded in tension

Threads	Tensile Stress Area (in^2)	Applied Force (lb)	Weakest Joint Material	Yield Strength (ksi)	Joint Stress (KSI)	Factor of Safety
M8	0.06076	117.66	7050 Al	71	1.94	36.66
M10	0.09486	117.66	6061-T6 Al	40	1.24	32.25
1/2 - 20	0.1599	117.66	6061-T6 Al	40	0.74	54.36

$Sigma_{bearing} = Force / (2 * thickness * diameter) = S_y/n_d$
$n_d = S_y / (F/(2 * thickness * diameter))$

$Tau = 0.577 * (S_y/ n_d) = F / (pi * d^2/2)$
$n_d = (0.577S_y * pi * d^2)/2 * F$

Joint	Diameter (in)	Clevis, Bearing Ys (ksi)	Bolt / Rod FSy (ksi)	thickness (in)	Applied force (lb)	Bearing FOS	Bolt Shear FOS
Upper Clevis	0.190	71.000	56	0.250	117.660	57.326	15.573
Lower Clevis	0.197	71.000	30	0.350	117.660	83.150	8.955

A2.3 Finite Element Analysis Results

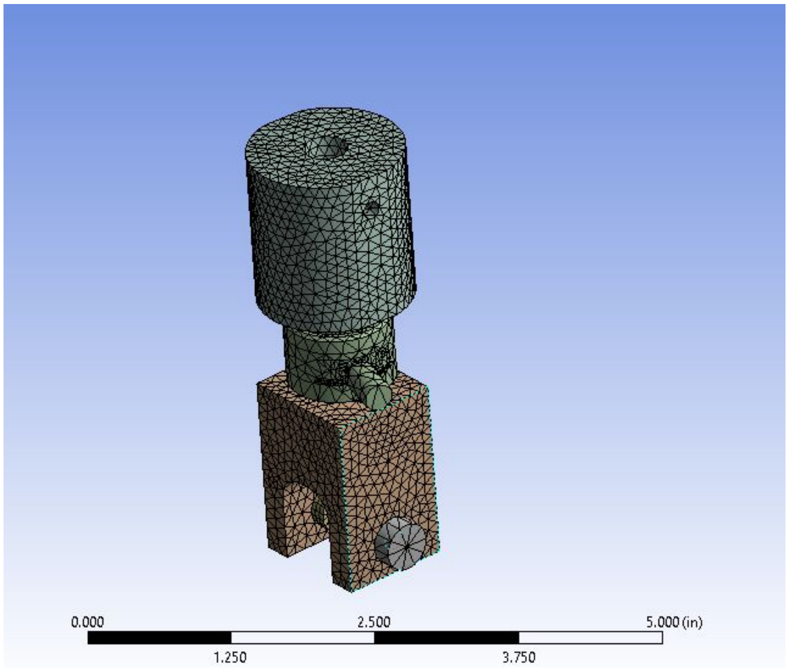


Figure xxx: Load Cell Adapter / Load Cell / Damper Interface Mesh

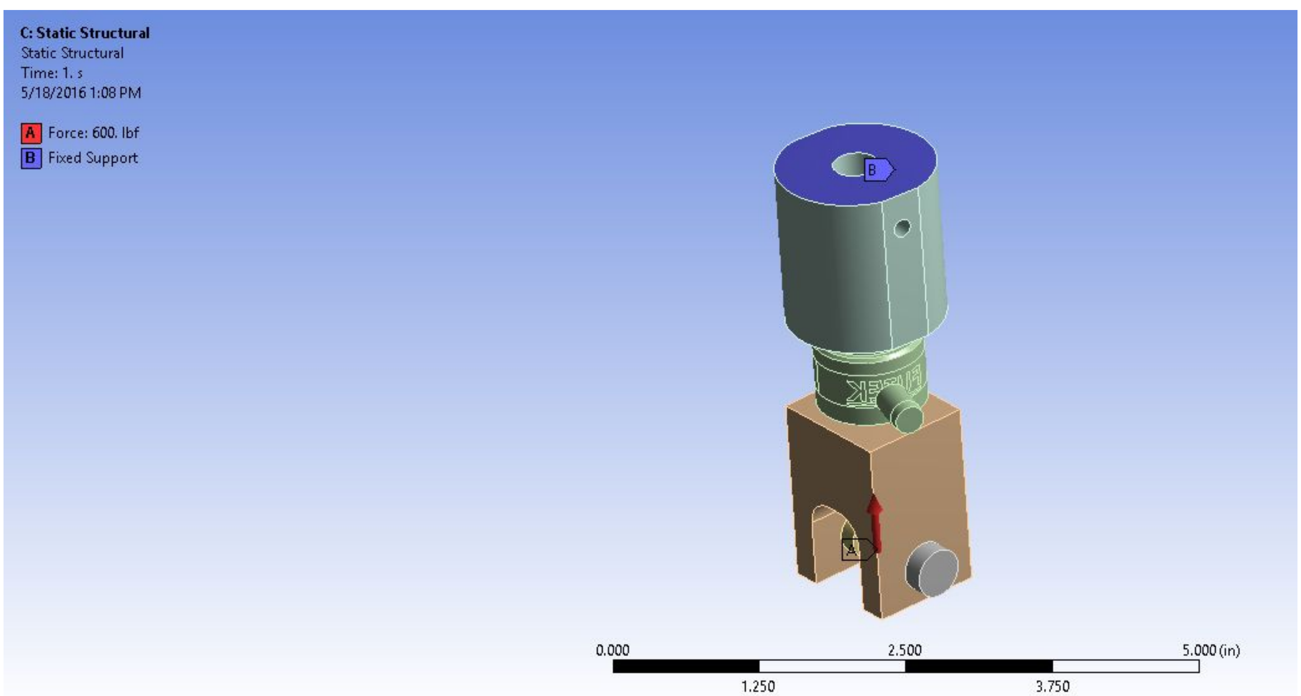


Figure xxx: Load Cell Adapter / Load Cell / Damper Interface, compression stroke

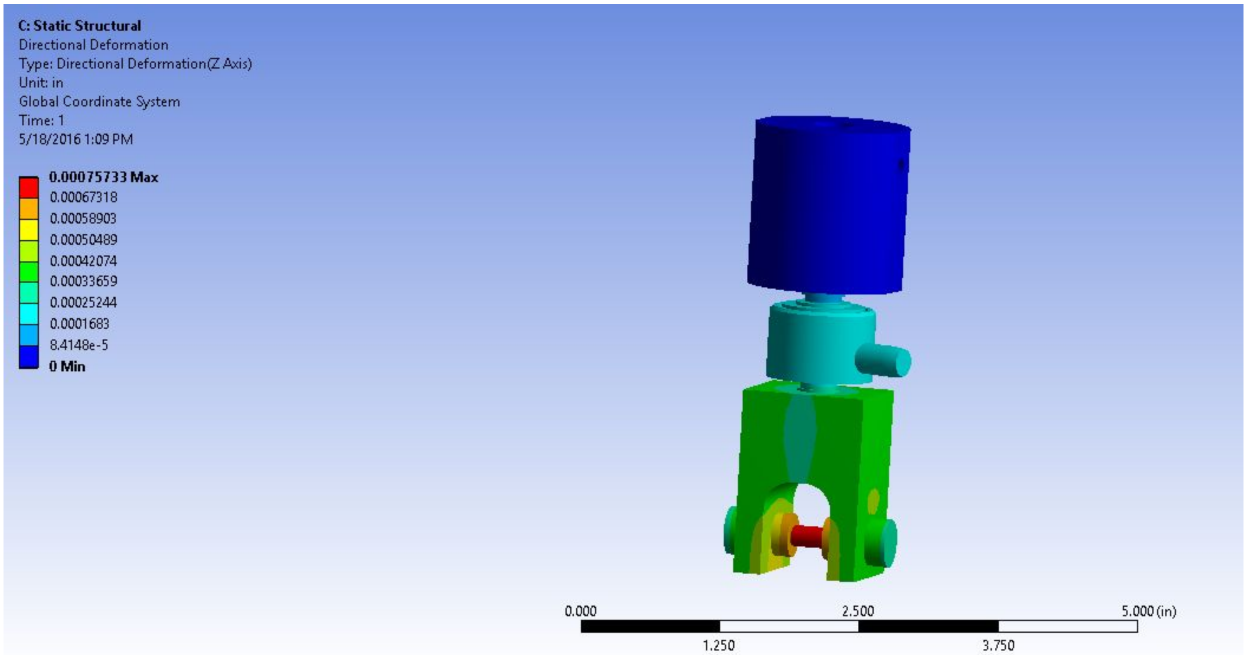


Figure xxx: Load Cell Adapter / Load Cell / Damper Interface, compression stroke, directional deformation

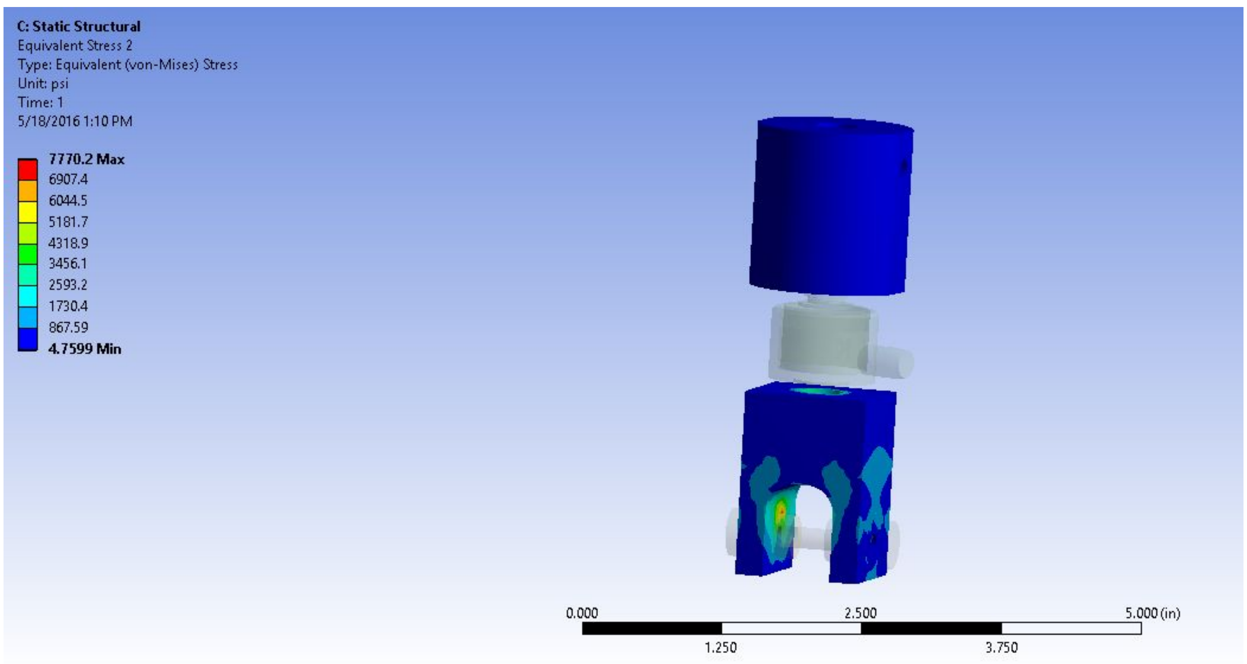


Figure xxx: Load Cell Adapter / Load Cell / Damper Interface, compression stroke, Von-mises equivalent stress

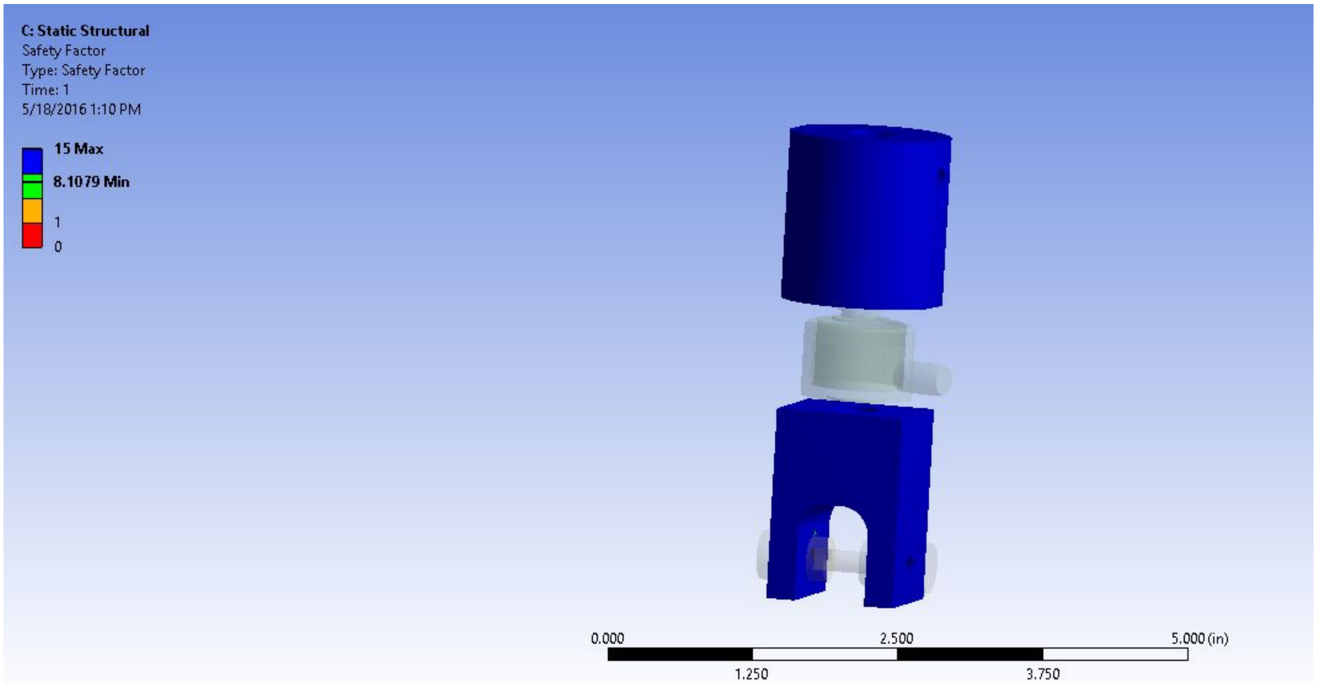


Figure xxx: Load Cell Adapter / Load Cell / Damper Interface, compression stroke, factor of safety

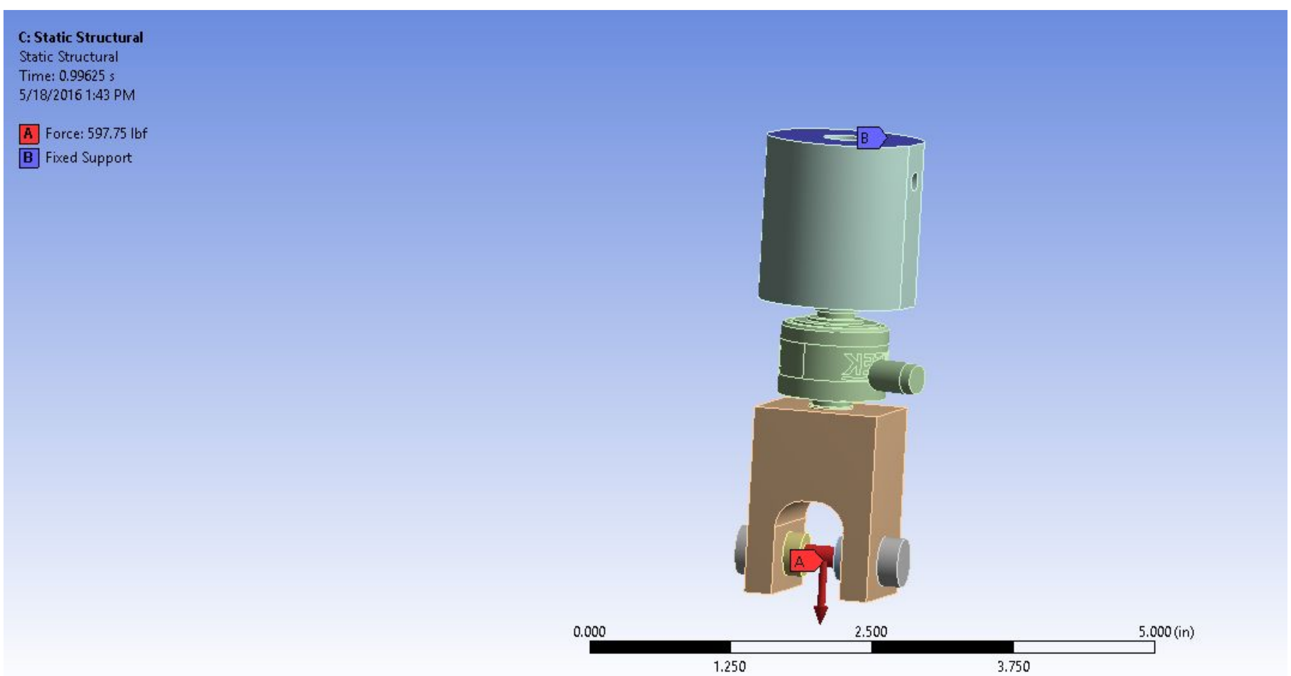


Figure xxx: Load Cell Adapter / Load Cell / Damper Interface, expansion stroke, applied loads

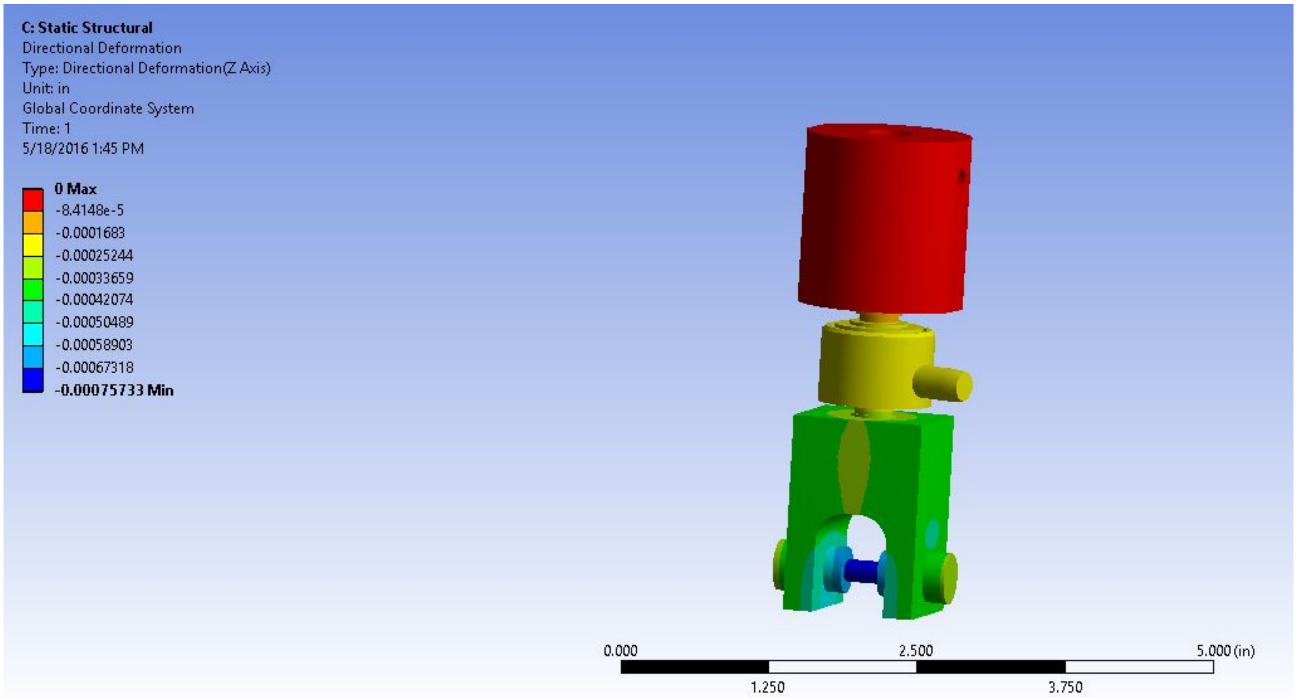


Figure xxx: Load Cell Adapter / Load Cell / Damper, expansion stroke, directional deformation

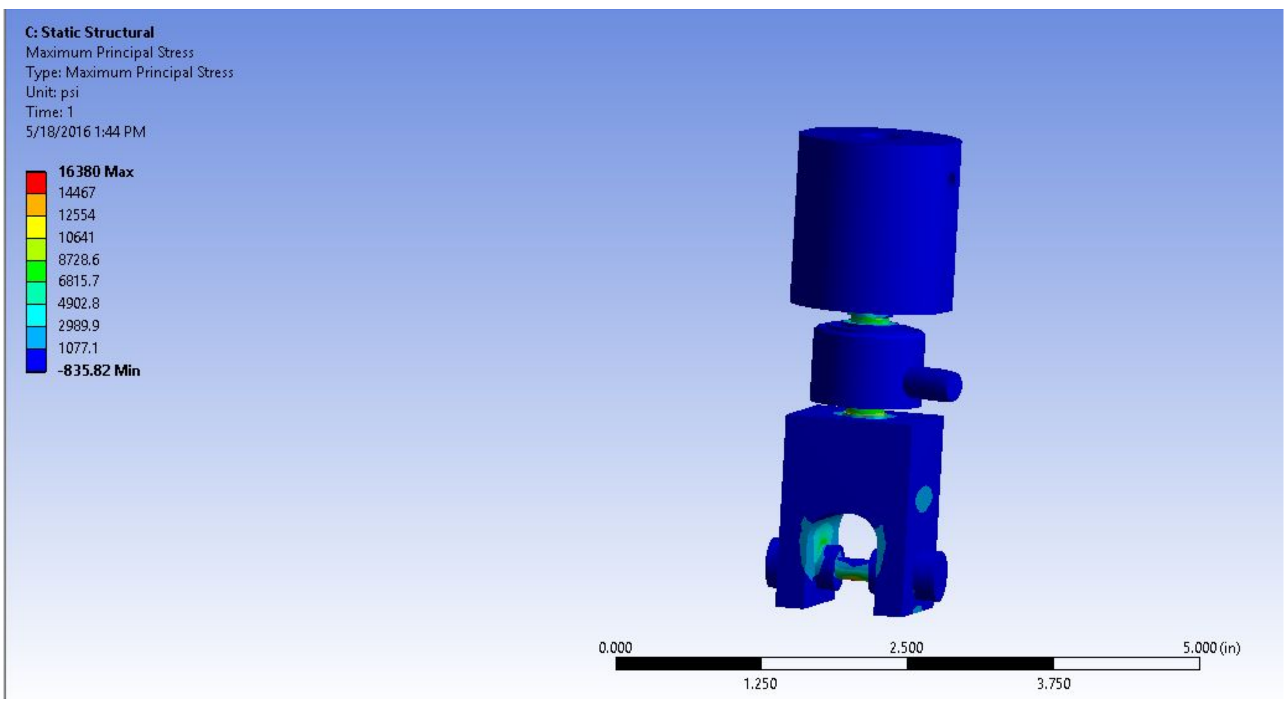


Figure xxx: Load Cell Adapter / Load Cell / Damper, expansion stroke,, Von-Mises equivalent stress

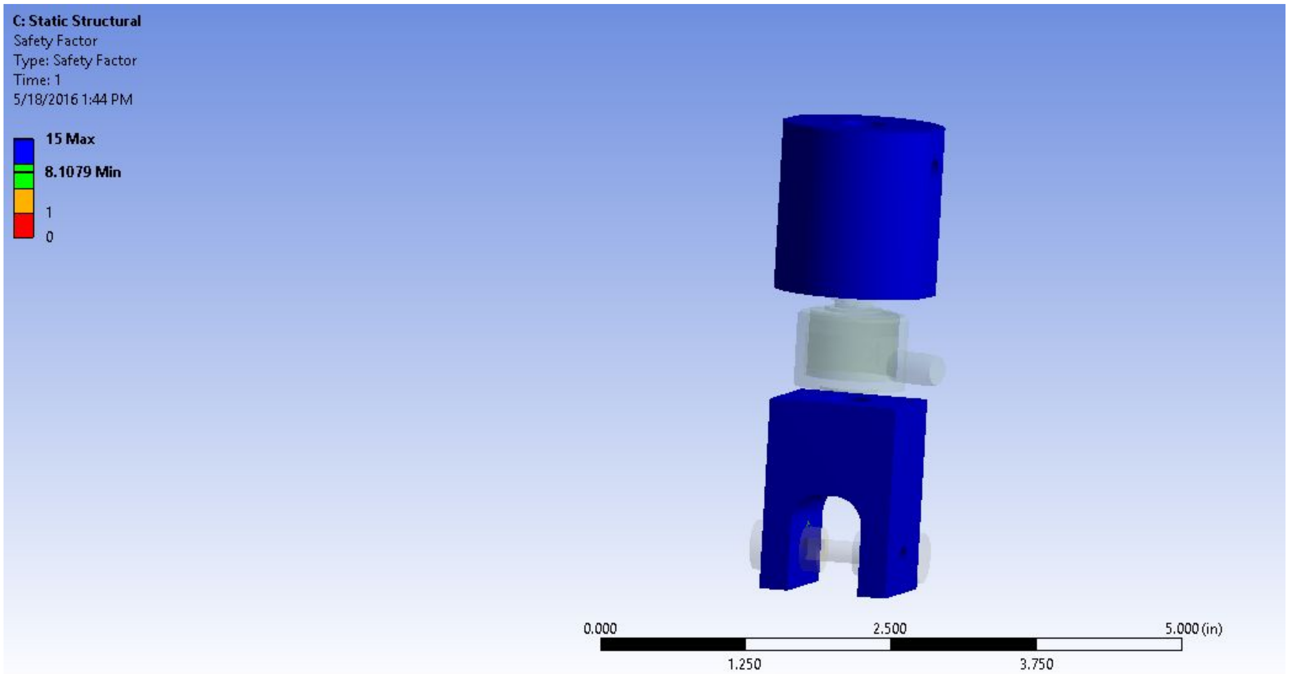


Figure xxx: Load Cell Adapter / Load Cell / Damper, expansion stroke, factor of safety

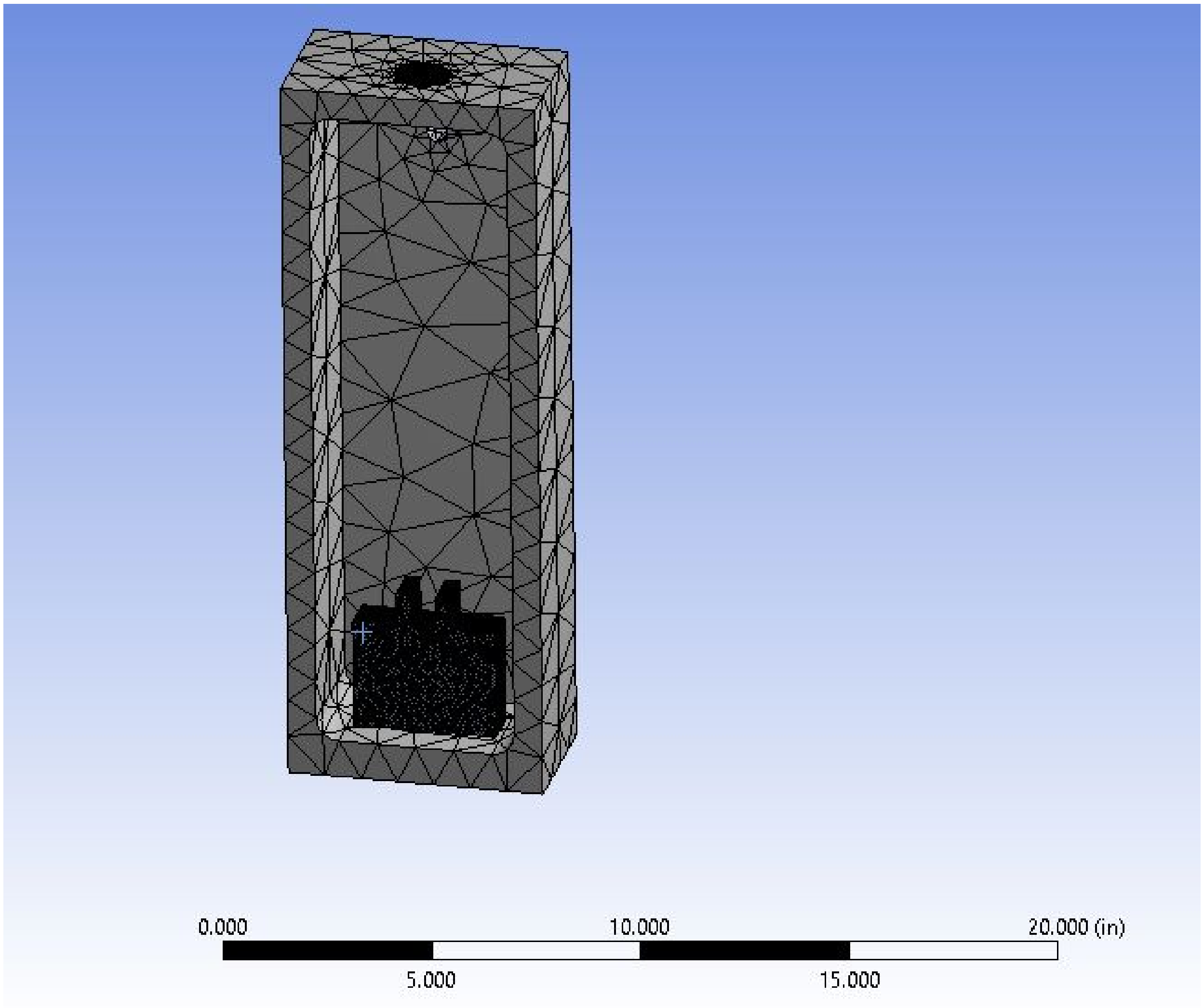


Figure xxx: Damper Base / Mounting Bracket, Mesh

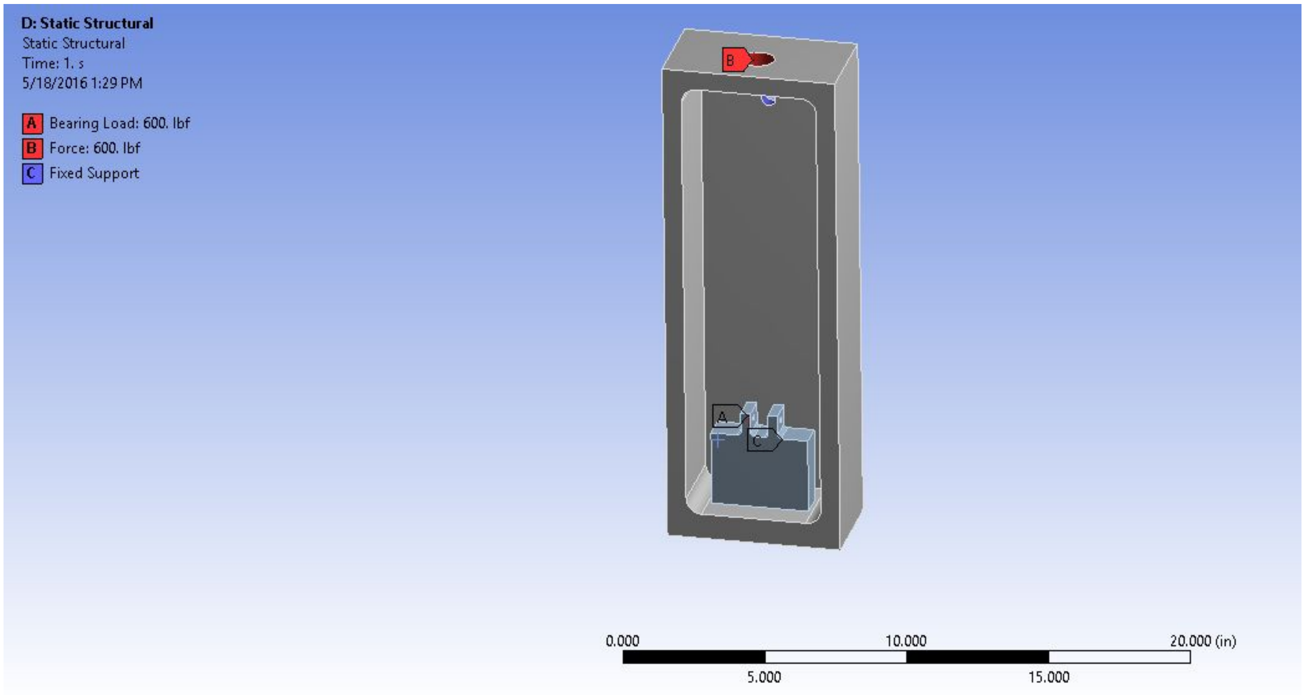


Figure xxx: Damper Base / Mounting Bracket, compression stroke, applied load

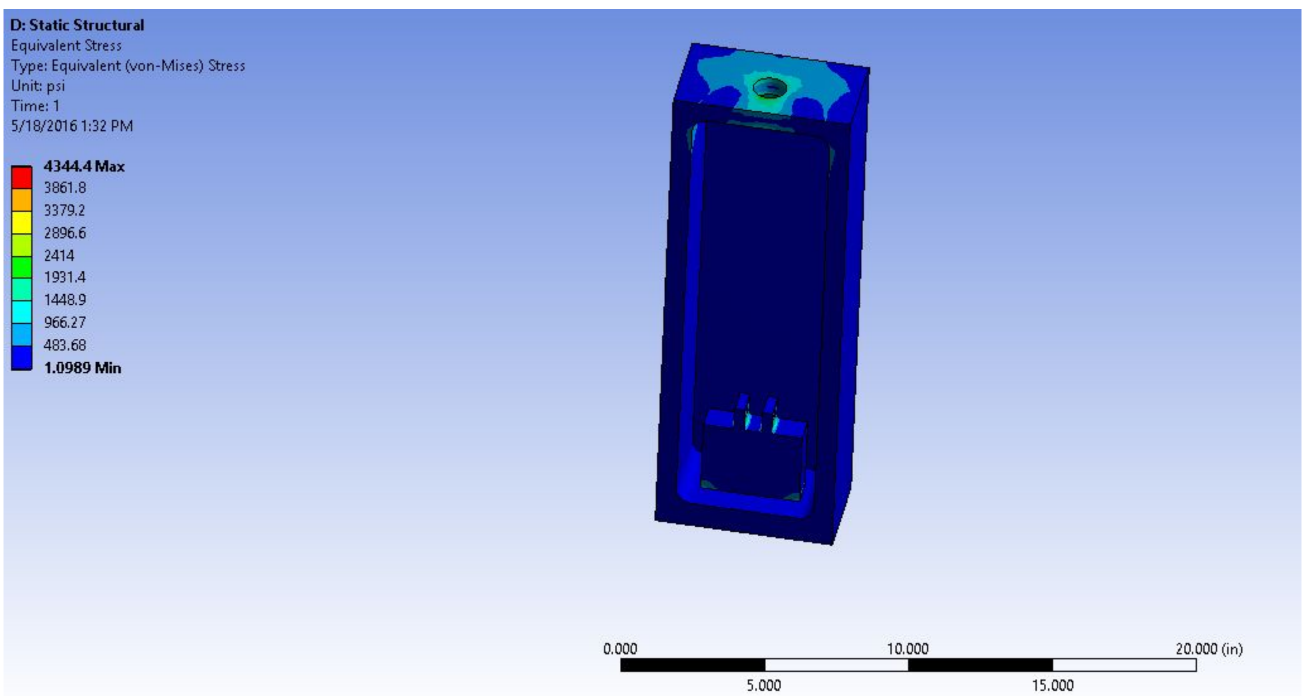


Figure xxx: Damper Base / Mounting Bracket, compression stroke, Von-mises equivalent stress

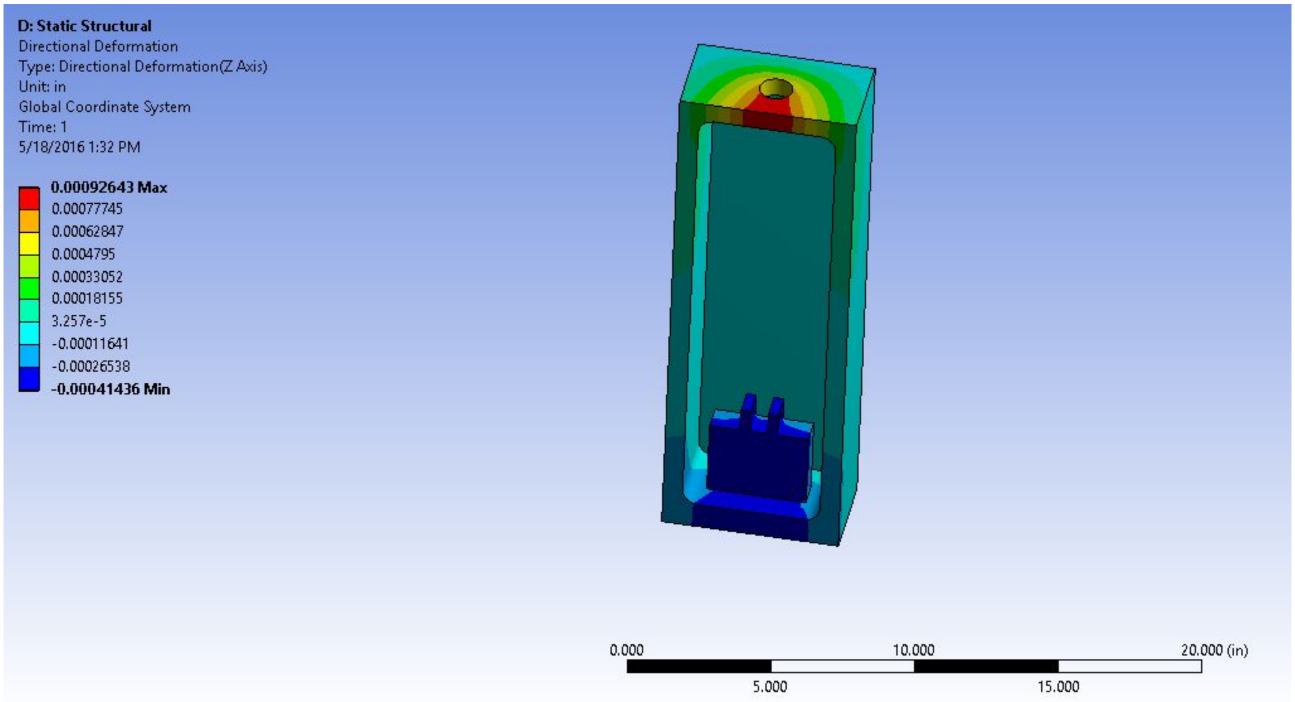


Figure xxx: Damper Base / Mounting Bracket, compression stroke, Von-mises equivalent stress

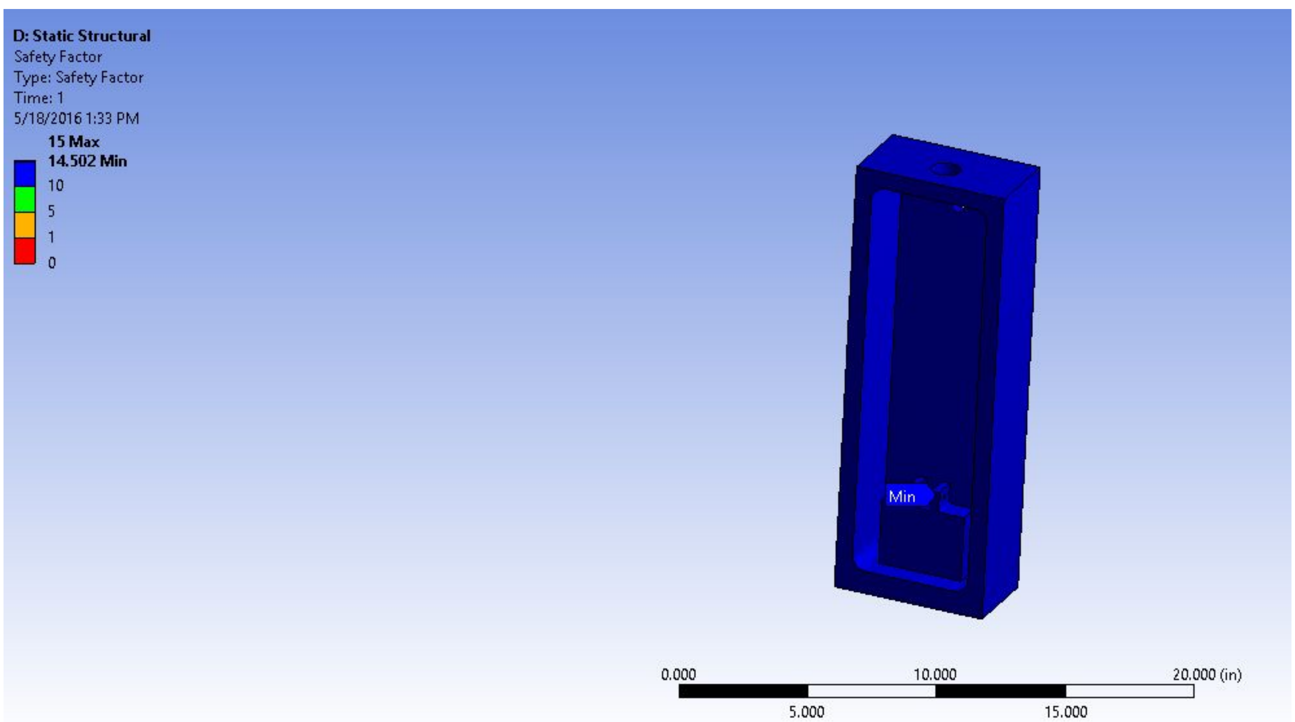


Figure xxx: Damper Base / Mounting Bracket, compression stroke, factor of safety

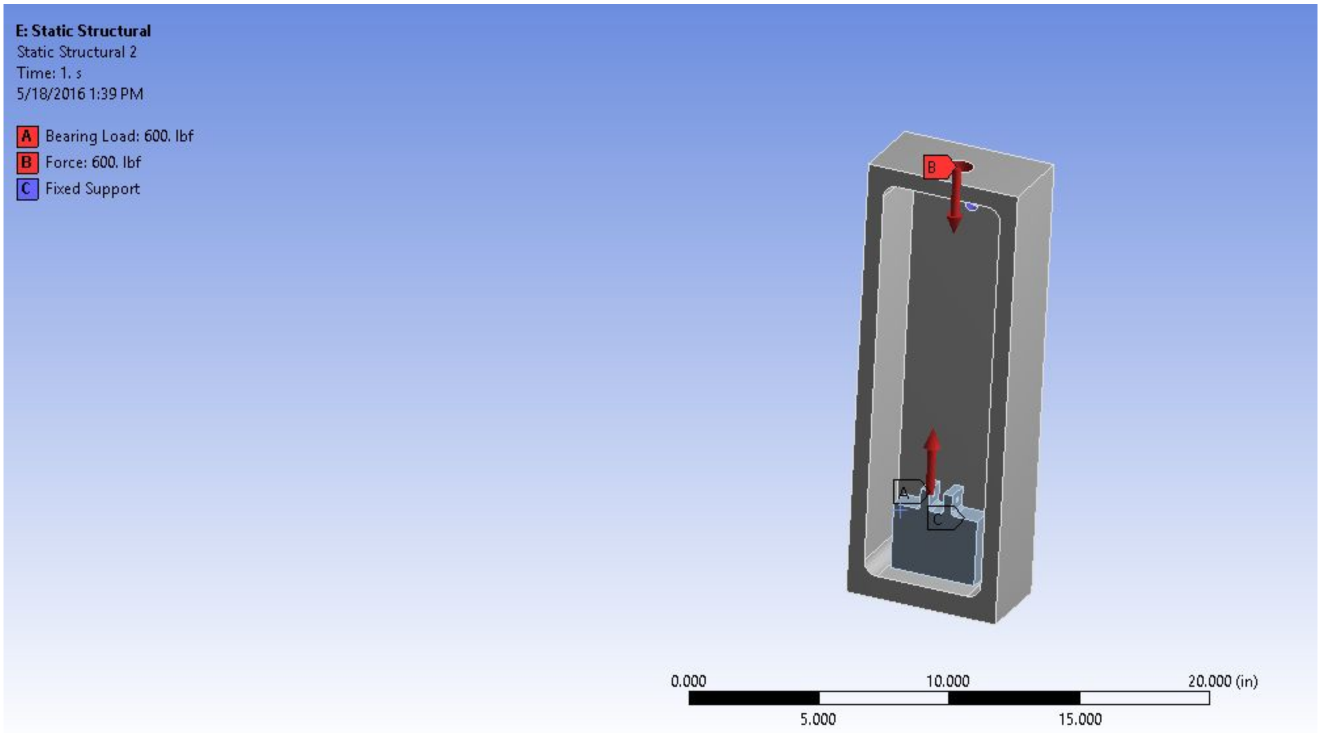


Figure xxx: Damper Base / Mounting Bracket, expansion stroke, applied load

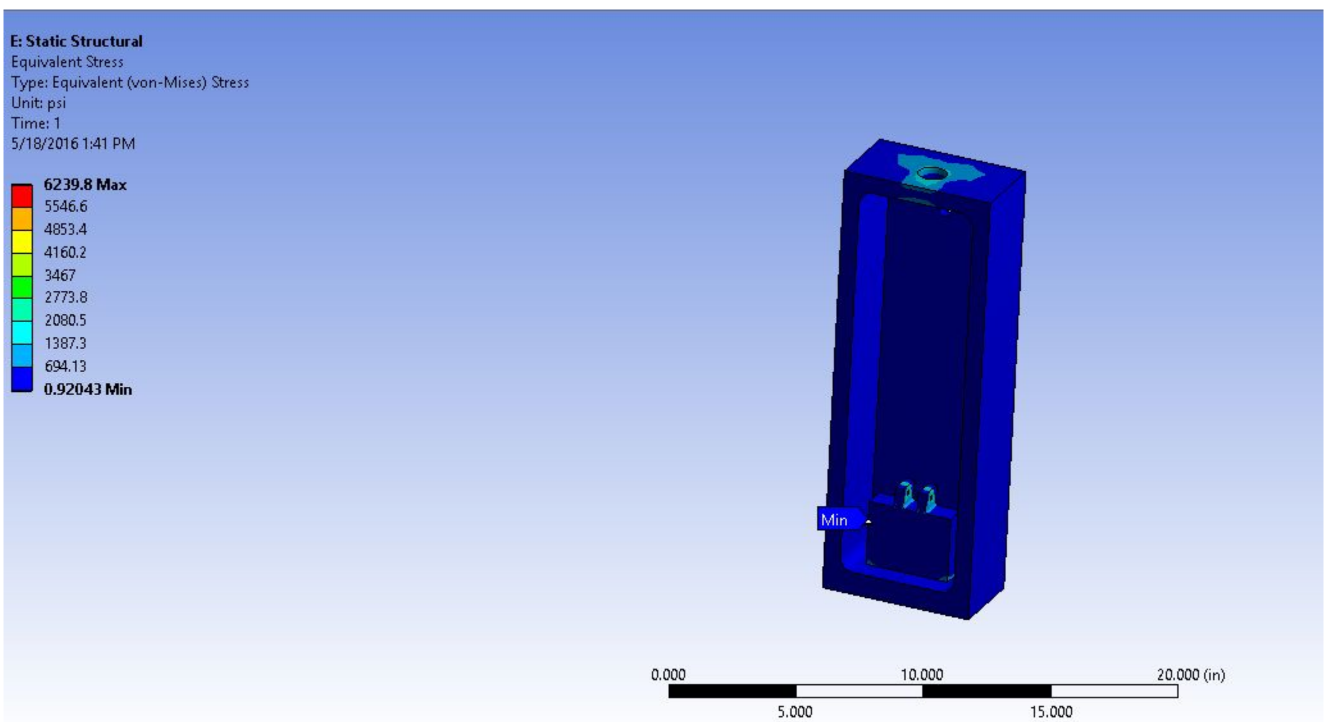


Figure xxx: Damper Base / Mounting Bracket, expansion stroke, Von-mises equivalent stress

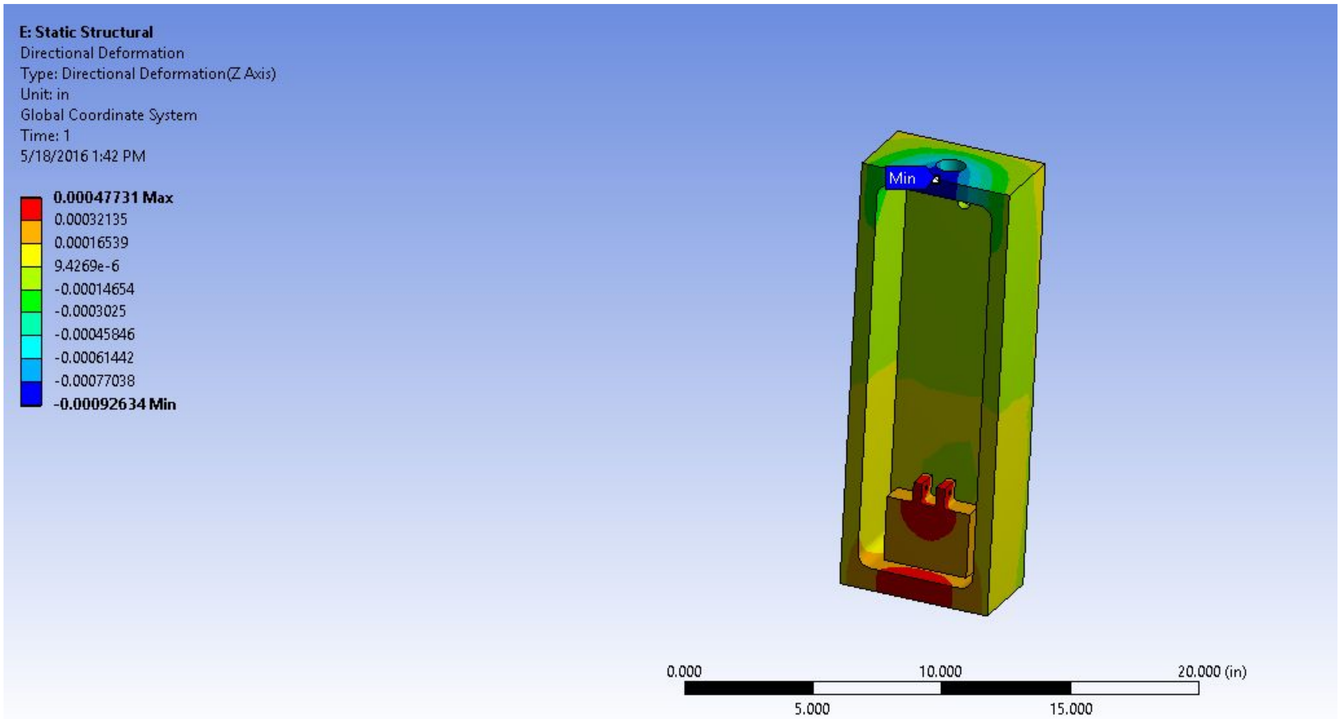


Figure xxx: Damper Base / Mounting Bracket, expansion stroke, directional deformation

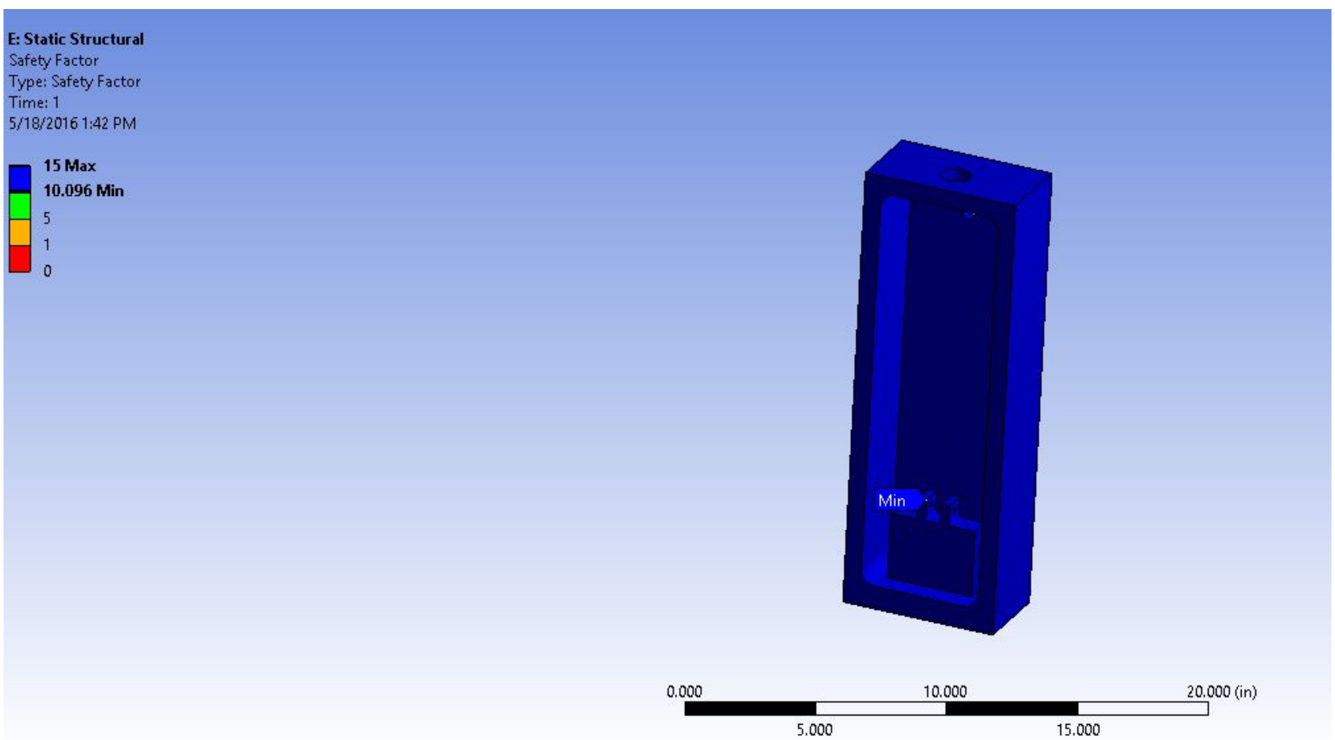


Figure xxx: Damper Base / Mounting Bracket, expansion stroke, factor of safety

APPENDIX 3: Test Results

A3.1 Data

A3.2 Data Analysis

TBD

APPENDIX 4: Microcontroller

A4.1 Arduino Code

Note: Parts of the code used for debugging are still present but they are commented out.

```
int linpotPin = A0;
int linpotValue = 0;
int loadPin = A1;
int loadValue = 0;
int sensorMin = 0;
int sensorMax = 1023;
int start = 8;
int t1;
int t2;
int t3;
int t4;
int nonzero = 0;
float p1;
float p2;
int count = 0;
int state = -1;
int comp = 3;
int rebound = 5;
int Button;
int press = 0;
//Status = 0 is safe
int status;
int s = 0;
```

```
int maxPos;
int duty = 100;
#define datapts 205
float dataArray [datapts] [2];
int counter = 0;
int initial;
int forcei;
//Cycles need to be even
#define cycles 4


//Duty Cycle Calculation
int dCycle (int input) {
  int d = (255 / 100) * duty;
  return d;
}

//Position Conversion
float position (int input, int initial) {
  return .0029779 * input - .0029779 * initial;
}

//Velocity Calculation
float velocity (float dt) {
  return abs((p2 - p1) / (dt / 1000));
}

//Load Calculation
float load (float input, int input2) {
  //return 1.1854*(input-508);
  return 1.1854 * (input - input2);
}

void setup() {
  Serial.begin(115200);
  pinMode(13, OUTPUT);
  pinMode(start, INPUT);
  pinMode(comp, OUTPUT);
  pinMode(rebound, OUTPUT);
  t1 = millis();
  t3 = millis();
  p1 = position(linpotValue, initial);
  maxPos = analogRead(linpotPin);
```

```
  //Calibration
  if (digitalRead(start) == HIGH) {
    while (millis() < 5000) {
      linpotValue = analogRead(linpotPin);
      // record the maximum sensor value
      if (linpotValue > sensorMax) {
        sensorMax = linpotValue;
      }
      // record the minimum sensor value
      if (linpotValue < sensorMin) {
        sensorMin = linpotValue;
      }
    }
  }
}


void loop() {
  if (count >= cycles || status == 1) {
    if (s < 1) {
      delay(500);
      Serial.println(F("Press button to restart"));
      s++;
    }
    while (press < 2) {
      if (digitalRead(start) == HIGH) {
        press++;
      }
    }
    count = 0;
    counter = 0;
    status = 0;
    s = 0;
  }
  if (s < 1) {
    Serial.println(F("Press start button for 1.0s to start..."));
    s++;
  }
  if (count == 0 && status == 0) {
    digitalWrite(comp, LOW);
    digitalWrite(rebound, LOW);
```

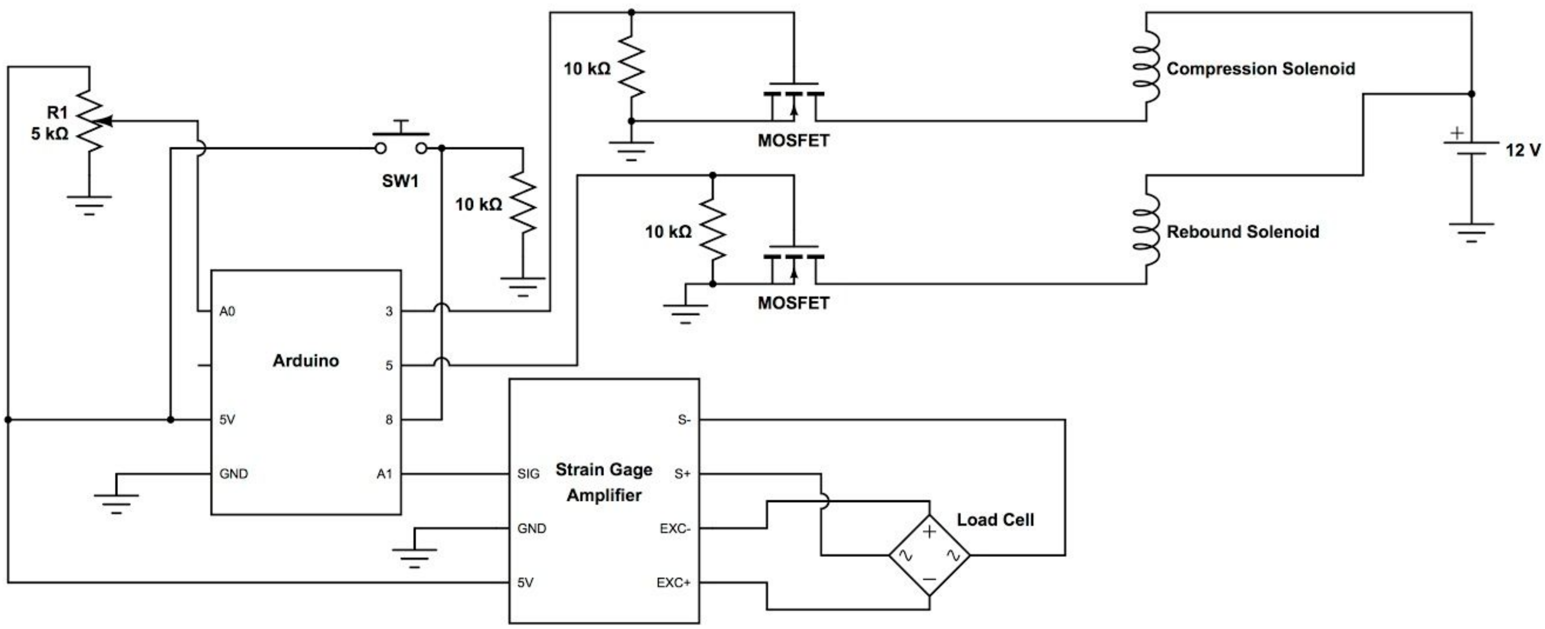
```
//Serial.println("Press start button for 1.5s to start...");
//Serial.println(digitalRead(start));
Button = digitalRead(start);
if (Button == HIGH && status == 0) {
  Serial.println(F("Wait for LED to come on before releasing"));
  delay(1500);
  digitalWrite(13, HIGH);
  Button = digitalRead(start);
  s = 0;
  while (Button == HIGH) {
    Button = digitalRead(start);
    if (s < 1) {
      Serial.println(F("Release Start Button to Start"));
      s++;
    }
    //      Serial.print("Button Pressed: ");
    //      Serial.println(digitalRead(start));
    //      Serial.print("Count: ");
    //      Serial.println(count);
  }
  initial = analogRead(linpotPin);
  forcei = analogRead(loadPin);
  status = 0;
  Serial.println(F("Initialize"));
  while (counter < datapts && status == 0) {
    //      Serial.println("In While Loop");
    //      Serial.print("Status: ");
    //      Serial.println(status);
    //      Serial.print("Count: ");
    //      Serial.println(count);
    // read the value from the sensor:
    linpotValue = analogRead(linpotPin);
    // apply the calibration to the sensor reading
    linpotValue = map(linpotValue, sensorMin, sensorMax, 0, 1023);
    // in case the sensor value is outside the range seen during calibration
    linpotValue = constrain(linpotValue, 0, 1023);
    //Serial.println(position(linpotValue),3);
    loadValue = analogRead(loadPin);
    loadValue = map(loadValue, sensorMin, sensorMax, 0, 1023);
    loadValue = constrain(loadValue, 0, 1023);
    if (linpotValue > maxPos) {
      maxPos = linpotValue;
    }
```

```
t2 = millis();
t4 = millis();
p2 = position(linpotValue, initial);
if (t2 - t1 >= 1) {
  float dt = t2 - t1;
  if (p1 != (float) 0 || p1 != (float) - 0) {
    nonzero++;
  }
  //    Serial.print("Position One = ");
  //    Serial.println(p1);
  //    Serial.print("Position Two = ");
  //    Serial.println(p2);
  //    Serial.print("dt = ");
  //    Serial.println(dt);
  //                        Serial.print(",");
  //          //Serial.print("Velocity = ");
  //Serial.print(velocity(dt), 3);
  //          //Serial.println("in/sec");
  //          Serial.print(",");
  //          Serial.print(analogRead(loadPin));
  //          Serial.println();
  //        Serial.print("Compression: ");
  //        Serial.println(digitalRead(comp));
  //        Serial.print("Rebound: ");
  //        Serial.println(digitalRead(rebound));
  //        Serial.print("State: ");
  //        Serial.println(state);
  //        Serial.println();
  if (counter < datapts & nonzero > 3) {
    //dataArray[counter][0] = micros();
    //dataArray[counter][0] = p1;
    dataArray[counter][1] = velocity(dt);
    dataArray[counter][2] = load(loadValue, forcei);
    counter++;
  }
  if (state == -1) {
    digitalWrite(comp, HIGH);
    digitalWrite(rebound, LOW);
    //analogWrite(comp, dCycle(duty));
  }

  else {
    digitalWrite(rebound, HIGH);
```

```
  //analogWrite(comp, dCycle(duty));
  digitalWrite(comp, LOW);
}
if (abs(velocity(dt)) > 10 && t4 - t3 > 40) {
  //digitalWrite(13, HIGH);
  state = -state;
  count++;
  t3 = t4;
}
else {
  //digitalWrite(13, LOW);
}
if (counter >= datapts) {
  digitalWrite(comp, LOW);
  digitalWrite(rebound, LOW);
  for (int j = 0; j < datapts; j++) {
    //Serial.println(dataArray [j] [0]);
    //Serial.print(", ");
    Serial.print(dataArray [j] [1]);
    //          Serial.print(", ");
    Serial.println(dataArray [j] [2], 3);
  }
  Serial.println(F("Successfully Completed"));
  s = 0;
  digitalWrite(comp, LOW);
  digitalWrite(rebound, LOW);

}
p1 = p2;
t1 = t2;
Button = digitalRead(start);
if (Button == HIGH || position(linpotValue, initial) > 2.2) {
  //if (Button == HIGH || position(linpotValue) < 2.2) {
  //Check for position trigger
  //Serial.println(position(linpotValue));
  status = 1;
  if (Button == HIGH) {
    Serial.println(F("Kill Switch Activated"));
  }
  if (position(linpotValue, initial) > 2.2) {
    Serial.println(F("Position Fault"));
  }
  digitalWrite(comp, LOW);
```

```
          digitalWrite(rebound, LOW);
          s = 0;
          break;
        }
      }
    }
  }
 }
}
```

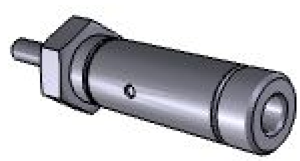
A4.2 Circuit Diagram



APPENDIX 5: Component Specifications

A5.1 Air Cylinder

Your configured part number: **0070.011**
Additional accesories: **(NO ACCESSORIES)**



Features and Benefits

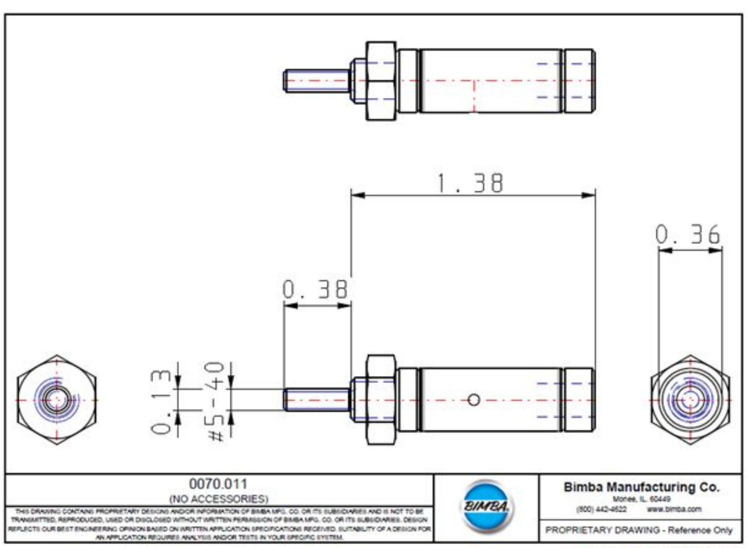
- **Maintenance Free Performance:**
  - Lubed for life w/proprietary synthetic blend grease
  - Blue and improved design doubles previous cylinder life
- **Added Strength:**
  - Piston to rod connection is threaded, sealed, and riveted securely in place
  - Roll formed rod threads on both ends
  - Aluminum alloy piston with blow-by flats
- **Low Breakaway:**
  - Inflatable, wear compensating, U-cup rod and piston seals
  - Breakaway slots on each end cap for fast seal inflation

Engineering Specifications

Pressure Rating: 250 PSI
Temperature Range:
Buna N seals: -20° F (-29° C) to 200° F (90° C)
Fluoroelastomer seals for high temperatures (up to 400° F) available
Magnetic piston option: max. operating temp. is 185° F
Materials:
Rod: High strength carbon steel and stainless steel available
Body: 304 stainless steel
End Caps: 6061 aluminum (RoHS compliant)
Rod Bushings: Oil-impregnated bronze except (5/16", 7/16", 9/16" single acting)

Service and Design Capabilities

- **Catalog models available in 4 days or less**
- **Custom design capabilities for your needs**
- **Common modifications include:**
  - Custom rod threads/lengths
  - Customer logo/part# on cylinder body
  - Additional standard port sizes/locations
  - Special lubricants (FDA approved, etc.)
  - Anodized or nickel plated end caps
  - Clean room design with Krytox lubrication
  - Special spring forces

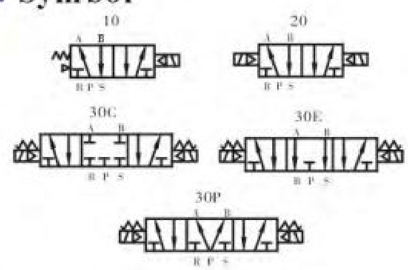


A5.2 Solenoid Valve

SOLENOID VALVE(5/2 5/3 WAY)

4V300 SERIES

Symbol

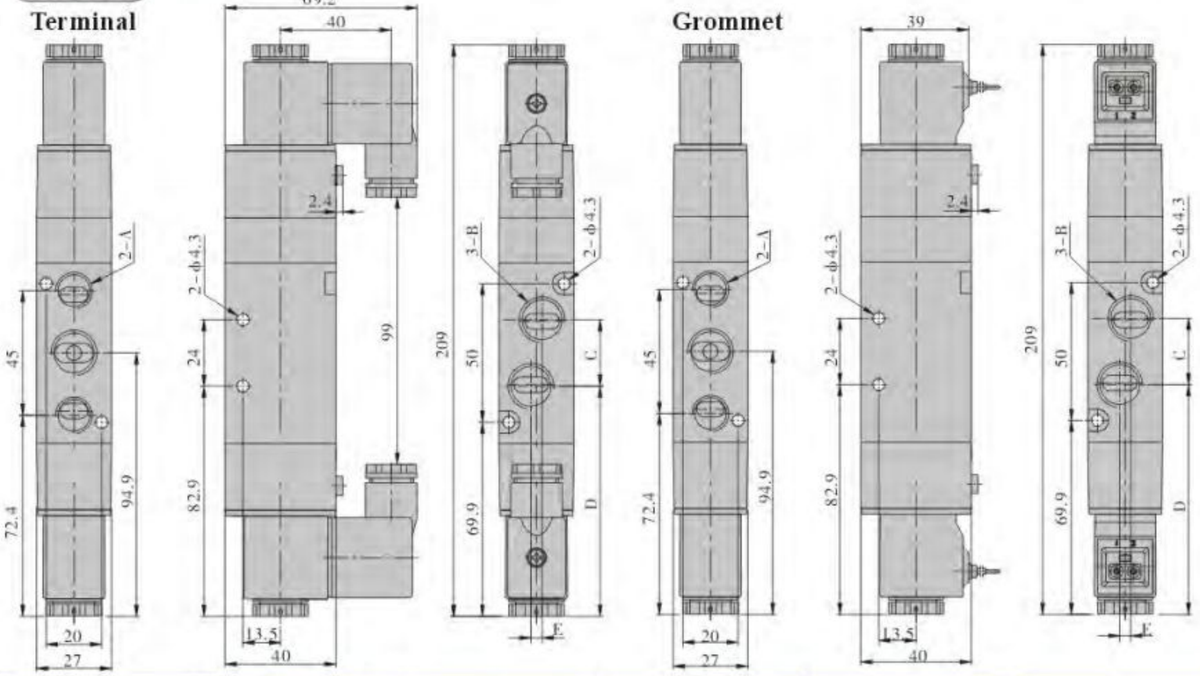


Specification

Item\Type	4V310-08	4V320-08	4V330C-08	4V330E-08	4V330P-08	4V310-10	4V320-10	4V330C-10	4V330E-10	4V330P-10
Fluid	Air (to be filtered by 40 μ filter element)									
Operating	Internally piloted									
Valve type	5 port 2 position		5 port 3 position			5 port 2 position		5 port 3 position		
Orifice size	25mm^2(Cv=1.4)		18mm^2(Cv=1.00)			30mm^2(Cv=1.68)		18mm^2(Cv=1.00)		
Port size	In=Out=Exhaust=1/4″					In=Out=3/8″ Exhaust=1/4″				
Lubrication	Not required									
Pressure range	1.5~8.0 bar (0.15~0.8MPa)(21~114Psi)									
Proof pressure	12.0 bar (1.2MPa)(170Psi)									
Temperature	-5~60℃(23~140℉)									
Voltage range	-15%~+10%									
Power consumption	AC: 3.5VA DC: 2.5W									
Insulation	B class									
Protection	IP65 (DIN40050)									
Connector	DIN Terminal or Grommet									
Max frequency	4 cycle/sec		3 cycle/sec			4 cycle/sec		3 cycle/sec		
Min activating time	0.05 sec									
Weight	310g	400g	540g	540g	540g	310g	400g	540g	540g	540g

*Note: PT、NPT、BSPP Thread are available.

4V330



Model \ Item	A	B	C	D	E
4V330-08	1/4″	1/4″	22	83.9	0
4V330-10	1/4″	3/8″	24	82.9	4

A5.3 Load Cell

FUTEK ADVANCED SENSOR TECHNOLOGY, INC.

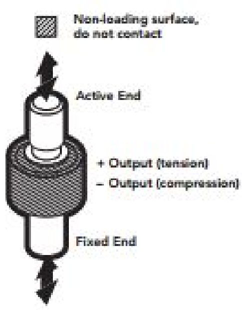
MODEL LCM325

Miniature Threaded In Line Load Cell

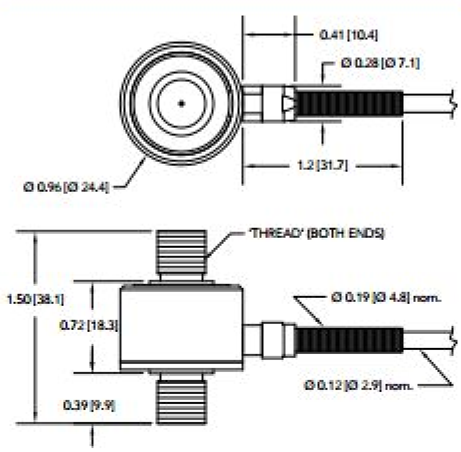
SPECIFICATIONS	
PERFORMANCE	
Nonlinearity	±0.5% of RO
Hysteresis	±0.5% of RO
Nonrepeatability	±0.1% of RO
ELECTRICAL	
Rated Output (RO)	1.3 mV/V (2 klb) 2 mV/V (3 klb)
Excitation (VDC or VAC)	18 max
Bridge Resistance	350 ohm nom
Insulation Resistance	≥500 Mohm @ 50 VDC
Connection	#28 AWG, 4 conductor, braided shielded PVC cable, 10 ft [3 m] long
Wiring/Connector Code	WC1
MECHANICAL	
Weight (approximate)	4 oz [113 g]
Weight (minus cable)	1.8 oz [51 g]
Safe Overload	150% of RO
Deflection	0.001 in [0.05 mm] nom
Material (flexure)	17-4 PH stainless-steel
IP Rating	IP64
TEMPERATURE	
Operating Temperature	-45 to 200°F (-42 to 93°C)
Compensated Temperature	60 to 160°F (15 to 72°C)
Temperature Shift Zero	±0.005% of RO/°F (0.01% of RO/°C)
Temperature Shift Span	±0.02% of Load/°F (0.036% of Load/°C)
CALIBRATION	
Calibration Test Excitation	10 VDC
Calibration (standard)	5-pt Tension
Calibration (available)	Compression
Shunt Calibration Value	100 kohm (2 klb), 60.4 kohm (3 klb)

FEATURES

- Miniature size
- Fast response and low deflection
- Robust cable strain relief
- For use in both tension and compression



DIMENSIONS inches [mm]



WIRING CODE (WC1)	
RED	+ EXCITATION
BLACK	– EXCITATION
GREEN	+ SIGNAL
WHITE	– SIGNAL
SHIELD	FLOATING

CAPACITIES

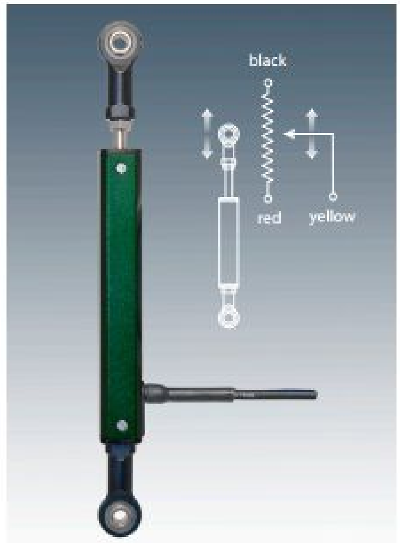
ITEM #	klb	kN	Thread	Natural Frequency
FSH03666	2	8.9	3/8-24	18kHz
FSH03667	2	8.9	M10x1.5	18kHz
FSH00672	3	13.3	3/8-24	18kHz
FSH03658	3	13.3	M10x1.5	18kHz

A5.4 Linear Potentiometer

CLP

Linear Potentiometer

Absolute Linear Position to 10 inches (250 mm)
2.5K - 10K ohms • High Cycle Applications
Factory Automation or Auto Sport Instrumentation
IP65 Protection



GENERAL	
Full Stroke Ranges	0-1 to 0-10 in. (0-25 to 0-250 mm)
Output Signal	voltage divider (potentiometer)
Linearity	see ordering information
Repeatability	0.01 mm
Resolution	essentially infinite
Life Expectancy	> 25 million cycles
Operating Speed	400 inches (10 M) per second max.
Enclosure Material	aluminum
Sensor	conductive plastic linear potentiometer
Weight	see ordering information

ELECTRICAL	
Input Resistance	see ordercode
Recommended Maximum Input Voltage	42 VDC
Recommended Operating Wiper Current	< 1µA

ENVIRONMENTAL	
Enclosure Design	IP65
Environmental Sealing	O-ring and felt shaft seal
Operating Temperature	-40° to 212°F
Vibration	up to 10 g to 2000 Hz maximum

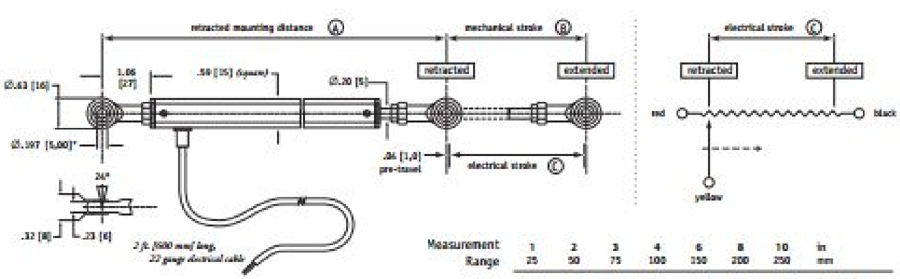
Developed specifically to meet the needs of the auto racing industry and proven in industrial applications, our CLP series position transducers offer unrivalled performance in terms of accuracy, repeatability, life expectancy and ease of mounting.

The combination of individually corrected conductive plastic elements and precious metal wipers provide a cost effective measuring system which can operate effectively without being unduly influenced by external environmental conditions.

Ordering Information:

Item Number	CLP-25	CLP-50	CLP-75	CLP-100	CLP-150	CLP-200	CLP-250
measurement range, in. [mm]:	1[25]	2[50]	3[75]	4[100]	6[150]	8[200]	10[250]
resistance, (±20%):	2.5K	5.0K	5.0K	5.0K	10K	10K	10K
linearity, %:	0.2%	0.2%	0.2%	0.2%	0.1%	0.1%	0.1%
weight, oz. [grams]:	3.0[87]	3.4[97]	3.8[108]	4.1[117]	4.8[138]	5.5[157]	6.2[177]

Outline Drawing:

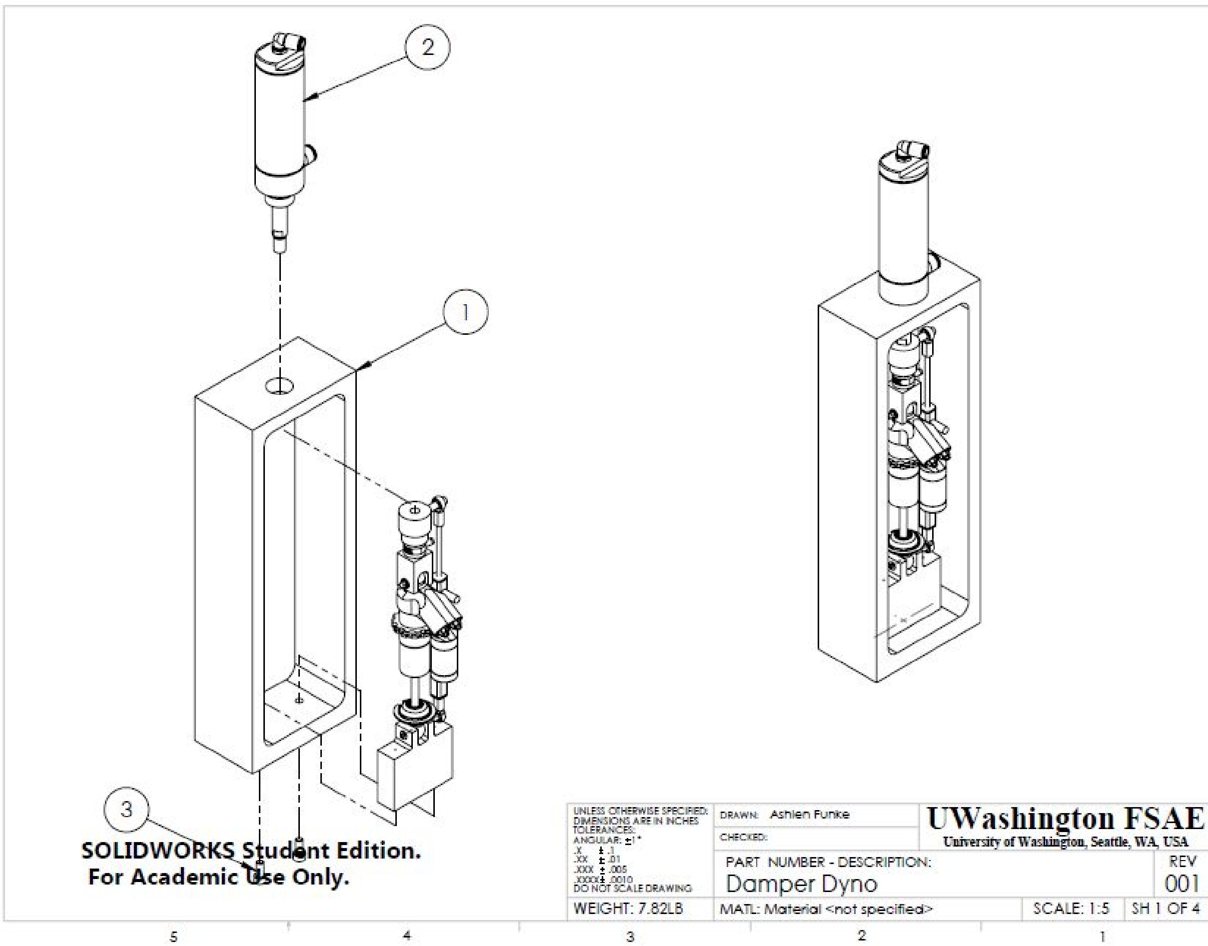


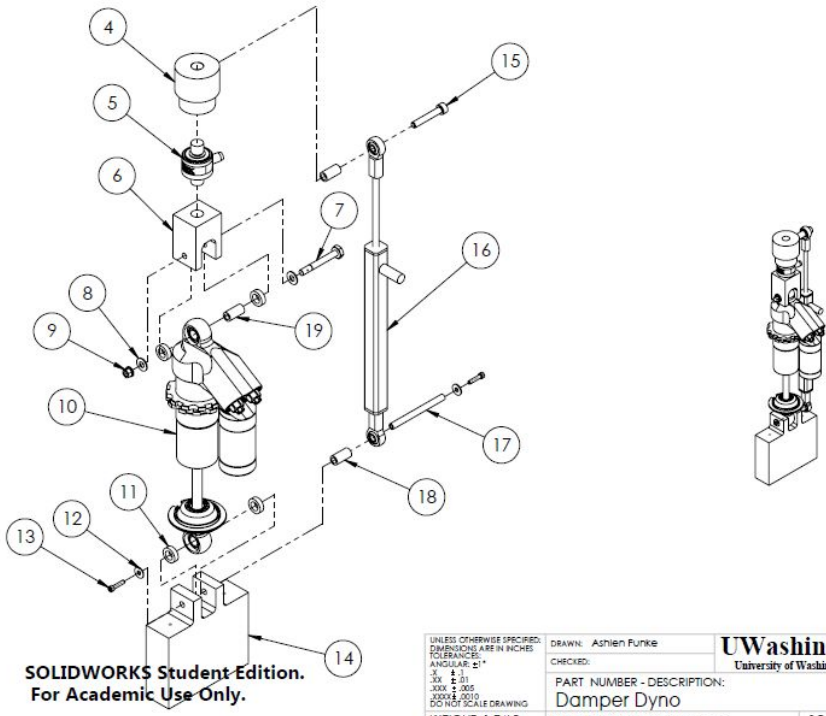
Measurement Range	1 25	2 50	3 75	4 100	6 150	8 200	10 250	in mm
Ⓐ	6.81 173	7.80 198	8.75 222	9.76 248	11.73 298	13.82 351	15.79 401	(± 0.1) in (± 2) mm
Ⓑ	1.12 28.5	2.11 53.5	3.09 78.5	4.08 103.5	6.04 153.5	8.13 206.5	10.10 256.5	(± 0.02) in (± 0.5) mm
Ⓒ	1.06 27.0	2.05 52.0	3.03 77.0	4.02 102.0	5.98 152.0	8.07 205.0	10.04 255.0	in mm

ALL DIMENSIONS ARE IN INCHES [MM]
tolerances are 0.03 IN. [0.5 MM] unless otherwise noted

*tolerance = +.002 -.000 [+0,05 -0,00]

APPENDIX 6: Exploded View





UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR: ±1° .X ±.1 .XX ±.01 .XXX ±.005 .XXXX ±.0010 DO NOT SCALE DRAWING	DRAWN: Ashlen Funke	UWashington FSAE University of Washington, Seattle, WA, USA	
	CHECKED:		
	PART NUMBER - DESCRIPTION: Damper Dyno		REV 001
WEIGHT: 1.74LB	MATL: Material <not specified>	SCALE: 1:3	SH 2 OF 4

5 4 3 2 1

ITEM NO.	PART NUMBER	exploded general/QTY.
1	Dyno_Base	1
2	SR-313_5-D	1
3	91502A194	2
4	Load Cell Adapter	1
5	27-SU-11209-Futek Load Cell	1
6	Damper Interface	1
7	AN3-14	1
8	93286A011	2
9	MS21042-L3	1
10	DB-FSAE	1
11	Damper Spacer	4
12	93286A007	2
13	91251A110	2
14	mounting bracket	1
15	91502A150	1
16	Celesco CLP 75	1
17	Potentiometer Spacer	3
18	5mm rod	1
19	Spacer_AN3	1

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR: ±1° .X ±.1 .XX ±.01 .XXX ±.005 .XXXX ±.0010 DO NOT SCALE DRAWING	DRAWN: DRNBY DRNDT	UWashington FSAE University of Washington, Seattle, WA, USA	
	CHECKED: CHKBY CHKDT		
	PART NUMBER - DESCRIPTION: Damper Dyno		REV 001
WEIGHT: LB	MATL:	SCALE: 1:3	SH 3 OF 4

5 4 3 2 1

APPENDIX 7: Bill of Materials

Part Name	Part No.	Quantity	Cost	Total Cost	Status
Dyno Base		1	$0.00	$0.00	Have stock, machined part
Mounting Bracket		1	$0.00	$0.00	Have stock, machined part
Damper Interface		1	$0.00	$0.00	Have stock, machined part
Load Cell Adapter		1	$0.00	$0.00	Have stock, machined part
Futek Load Cell		1	$0.00	$0.00	Acquired
Damper Spacer		4	$0.00	$0.00	Have stock, machined part
Bimba SR-313-5-D Pneumatic Actuator	SR-313-5-D	1	TBD	TBD	Need to purchase
Potentiometer	CLP-75	1	$0.00	$0.00	Acquired
Class 10.9 M8x40mm Cap Screw	90180A624	1	$6.65	$6.65	Need to purchase
5/16" Washer	93286A015	2	$0.00	$0.00	Acquired
Class 10 M8 Nyloc	94645A210	1	$8.19	$8.19	Need to purchase
No 10 Washer	93286A011	1	$0.00	$0.00	Acquired
Potentiometer Spacer		2	$0.00	$0.00	Have stock, machined part
Class 10 M5 Nyloc	90576A104	1	$4.27	$4.27	Need to purchase
Class 12.9 M5x65 SHCS	91290A270	1	$3.17	$3.17	Need to purchase
Class 12.9 M5x30 SHCS	91502A150	1	$6.07	$6.07	Need to purchase
Class 12.9 M8x1.25x18mm SHCS	91502A194	2	$10.02	$10.02	Need to purchase
1/4" NPT to 3/8" Tube PTC Elbow	5779K158	1	$4.75	$4.75	Need to purchase
1/4" NPT to 3/8" Tube PTC Straight	5148K125	1	$5.19	$5.19	
Total Cost				$48.31	

APPENDIX 8: Timeline

Week	Dates	Timeline
1	3/27 - 4/2	project definition / top level design
2	4/2 - 4/9	top level design / preliminary model
3	4/10 - 4/16	component selection / system analysis
4	4/17 - 4/23	component selection / system analysis
5	4/24 - 4/30	purchase parts / machining
6	5/1 - 5/7	dyno assembly / wiring
7	5/8 - 5/14	controls / troubleshooting
8	5/15 - 5/21	testing / data analysis
9	5/22 - 5/28	testing / data analysis
10	5/29 - 9/4	testing / data analysis