Main Pressurant Valves Documentation

Background:

The main pressurant valves consist of the main fuel valve and the main oxidizer valves. Both valves are responsible for the delivery of two pressurized liquids. The main fuel valve will be located on the fuel side of the test stand which carries Isopropyl alcohol. The main oxidizer valve will be located on the oxidizer side of the test stand which carries liquid oxygen. Both valves are required to have a system that would allow for manually controlled on/off actuators that will turn the ball valves at pressures of up to ~1000 psi. Since liquid oxygen is a cryogenic fluid, the LOX valve will need to have a system designed to prevent frost from accumulating and freezing the valve shut.

The Design:

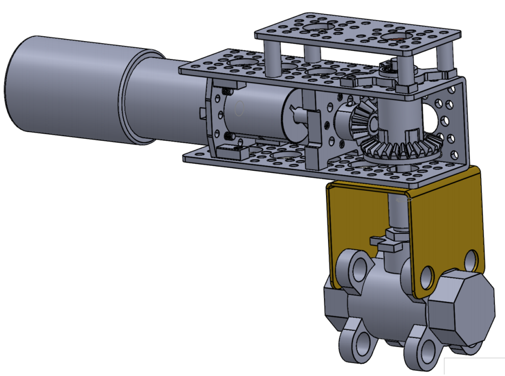

Figure 1: Main Fuel Valve

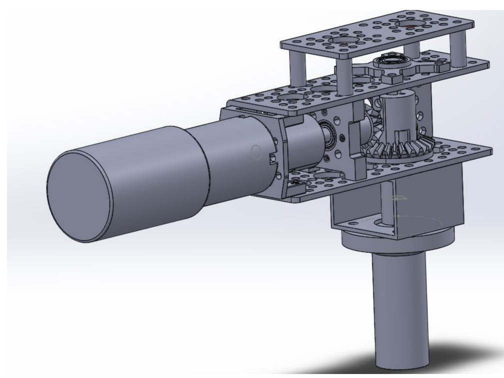

Figure 2: Main Oxidizer Valve

The configuration of the actuator went through many iterations and ended with the elbow design shown in Figure 1 and 2. The purpose of the elbow is to provide a method of gathering information of the ball valves position. In essence the shaft coming from the valve goes through a coupler and a bevel gear set where the shaft continues to protrude from the top. A special magnet is placed on the shaft with a sensor encoder located on top. With this setup the valve both gets its torque from the bevel gear set and the exact position at which the valve is oriented at. Thus, the controller will know when the valve is in the open or closed position. Both designs for the LOX and fuel valve actuators are very similar in nature. The only difference between them are the valves themselves and the types of brackets used to connect the actuators to the valves. To crack open and close the valves, a 12V 23 rpm DC motor is used, which will get the valve to open in about 1.25 seconds. The frame, motors, and supports are all parts found on the web, specifically from servocity.com. The brackets on the other hand were made in house using sheet metal for the MFV and a square tube for the LOX valve.

Current progress:

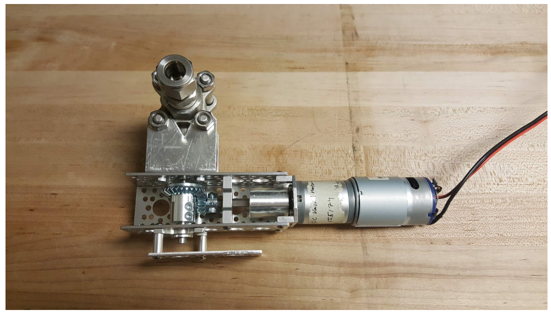

Figure 3: Built MFV. minus the TSAR components

Both the main fuel and LOX valves are built and assembled, minus the sensor portion of the actuator but the structure is there to implement the encoder (seen in figure 3). Integrating with TSAR on the side of controls is still necessary. As previously mentioned, the LOX valve freezing shut was a concern. A test was conducted to gather information on any torque changes that happen when exposed to cryogenic liquids, and if a greater torque motor was required. The test consisted of using liquid nitrogen, which is much safer to handle than liquid oxygen, to simulate cryogenic flow. Another benefit to using liquid nitrogen is its freezing point which is slightly lower than liquid oxygen. Shown in figure 4, the MFV was dipped in liquid nitrogen for a certain amount of time. After taking it out, the valve was measured by a force meter attached to a lever arm, which provided us with a reading of torque. Figure 5 shows the torque graph we gathered. After observing the torque required to open the valve at room temperature and then at cryogenic exposure, both graphs were identical and a conclusion was made that the cryogenic flow was irrelevant. Upon further notice, it was found that the experiment conducted was invalid because we did not let the valve collect enough frost. When given enough time, the humidity in the air will collect as frost when there is cryogenic flow. Given we will be operating the test stand outside, with prolonged cryogenic fluid pushed against the valve, there is a significant possibility that the valve could freeze shut. Further testing is needed.

Figure 4

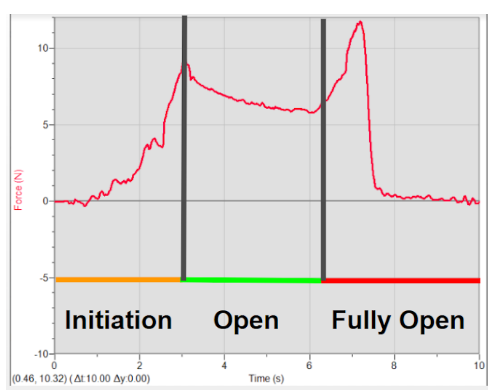

Figure 5

What needs to be done/recommendation:

- Wire the motor driver and verify operation via test code on an arduino
- Mount magnetic sensor pcb
- Test and possibly iterate a magnet attachment solution to affix the magnet to the top of the gear shaft
- Complete arduino-based test code to read sensors and actuate valve on a button press
- Integrate sensor and motor driver into TSAR actuator controller code
- Design an anti-freezing system for lox valve
    - The basic technique for this design is to bleed nitrogen into a plastic bag that covers the main LOX valve. In theory the nitrogen will push out the air surrounding the valve, essentially dehumidifying the system. No humidity means no frost collection on the valve.
- Test the main fuel valve under pressure
- Test lox valve with pressurized liquid nitrogen with anti-freezing system

Figure 6: Potential Main Fuel Valve and LOX valve mini test stand

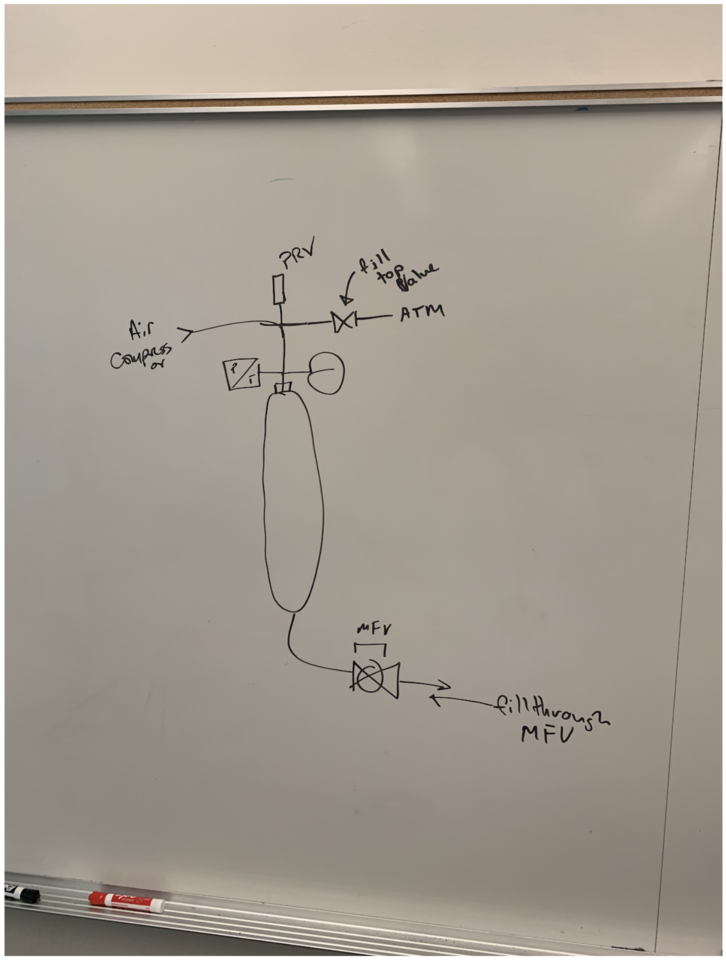

Figure 7: Schematic for a test run of the valves

Basic Procedure for Main Fuel valve: Main idea of this test is to see if the main fuel valve can be run under pressure.

- Open manual valve to atm.
- Fill pressure cylinder through main fuel valve with water.
- Close manual valve
- Pressurize pressure cylinder with air compressor till approximately 60 psi
- Actuate Valve

PRV: Pressure relief valve
Air Compressor: Provides pressure of up to 100 psi
Pressure transducer: Data acquisition of pressure that was used
MFV: Main Fuel Valve
Analog pressure gauge: Used for visual aid of pressure used
Manual valve: Manually turn valve for ATM vent during water filling

Basic procedure for Main LOX Valve: Main idea for this experiment is to test a cryogenic fluid under pressure. As mentioned before, a common mistake amateur rocketeers make is neglecting the fact that the main LOX valve WILL FREEZE SHUT when exposed at a prolonged duration with liquid oxygen. This experiment will also test the technique used to mitigate the frost collection on the valve by introducing a flow of nitrogen gas to purge the air around the valve and prevent any humidity from accumulating.

- Procedure will follow the same principle as in the MFV test.
- Incorporate N2 bag system around main LOX valve
    - In essence, this technique uses a bag to cover the main LOX valve while leaving to openings. Inlet will supply a steady flow of nitrogen while the outlet will purge all air/humidity surrounding the valve.

Magnetic rotator encoder example code:

The sensor and motor will be run on an arduino uno. The following code is just an example of how it can be setup. Iterations can be made to adjust to the ongoing changes.

```
#include <SPI.h>

//Set Slave Select Pin (MOSI, MISO, CLK are handled automatically)
int CSN = 10;
int SO = 12;
int SI = 11;
int CLK = 13 ;
unsigned int angle;
int pos;

int ledPin = 9; // choose the pin for the LED
volatile int store = 0; // variable for storing angle at moment of button press
int motor = 4; // choose motor output pin
int motordir = 5; // choose motor direction pin
volatile int actuate = 0; // should motor run
int delta = 0; //error b/w measured & desired

void setup() {

  Serial.begin(9600);

  //Set Pin Modes
  pinMode(CSN, OUTPUT);
  pinMode(SI, OUTPUT);
  pinMode(SO, INPUT);
  pinMode(CLK, OUTPUT);
  //Set Slave Select High to Start i.e disable chip
  digitalWrite(CSN, HIGH);
  //Initialize SPI
  SPI.begin();
```

```
  pinMode(ledPin, OUTPUT);  // declare LED as output
  attachInterrupt(digitalPinToInterrupt(2), buttonPress, FALLING); //interrupt for button press -
unsure if should be RISING or FALLING trigger
}

void loop() {

  SPI.beginTransaction(SPISettings(10000000, MSBFIRST, SPI_MODE1));

  //Send the Command Frame
  digitalWrite(CSN, LOW);
  delayMicroseconds(1);
  SPI.transfer16(0xFFFF);
  digitalWrite(CSN,HIGH);

  //Read data frame
  digitalWrite(CSN, LOW);
  delayMicroseconds(1);
  angle = SPI.transfer16(0xC000);
  digitalWrite(CSN, HIGH);
  SPI.endTransaction;

  angle = (angle & (0x3FFF));

  int pos = ( (unsigned long) angle)*360UL/16384UL; //convert 14 bit output to angle

  //Motor handling
  if (actuate = 1) {
    delta = abs(store - pos);
      if (delta > 180) {
        delta = abs(delta - 360);   //Ugly rotational handling to produce smaller angle b/w values
      }
      if (delta >= 90) {      //If it already went 90 degrees, stop!
        actuate = 0; //turn off motor trigger
        analogWrite(motor,0); //turn off motor
        digitalWrite(ledPin,LOW); //turn off led
      }
  }
  if (actuate = 1) {
    digitalWrite(motordir,HIGH); //Need to determine correct rotation direction, HIGH or LOW
    analogWrite(motor,255); //turn on motor at full speed
    digitalWrite(ledPin,HIGH); //turn on LED
  }

 // Serial.println(pos);    //Option to print current position
  Serial.println(delta);

  delay(100); //small delay
```

```
}

void buttonPress()  //Interrupt function, ostensibly
{
  store = pos;
  actuate = 1;
}
```