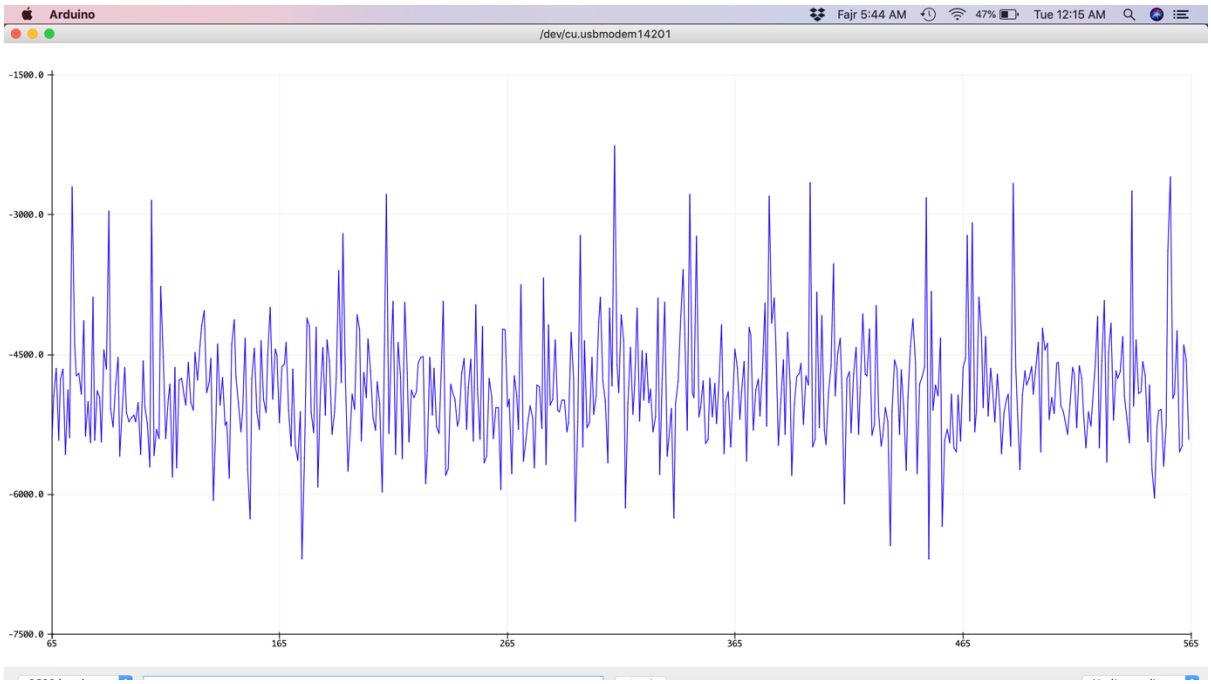
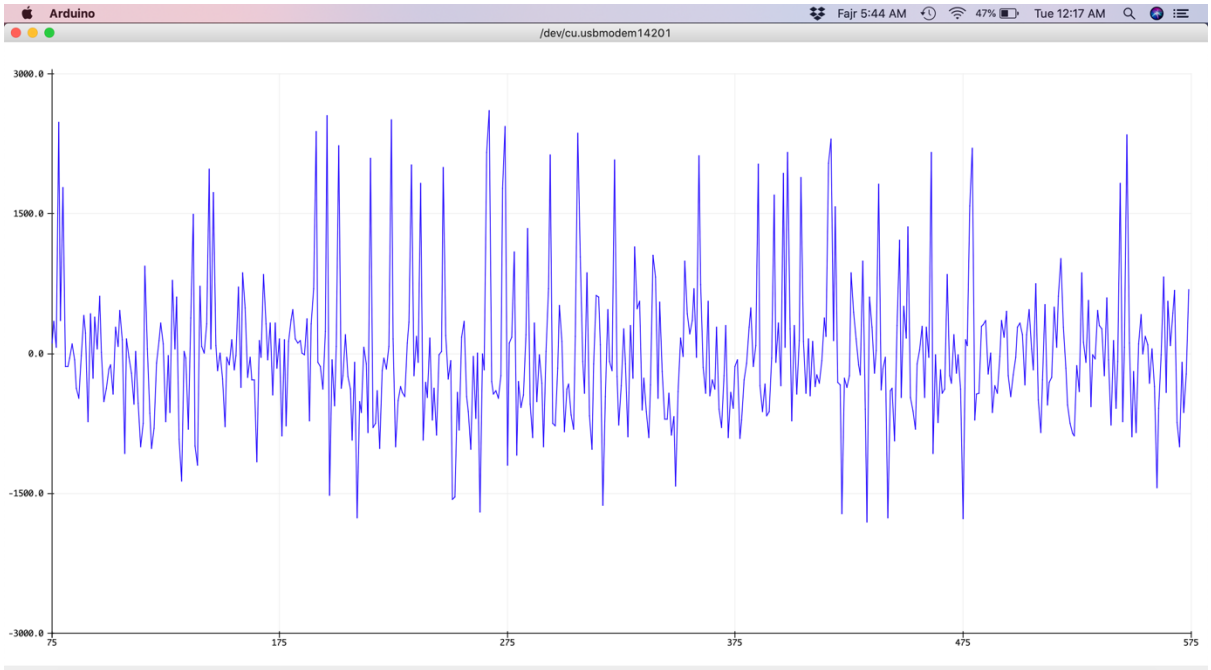
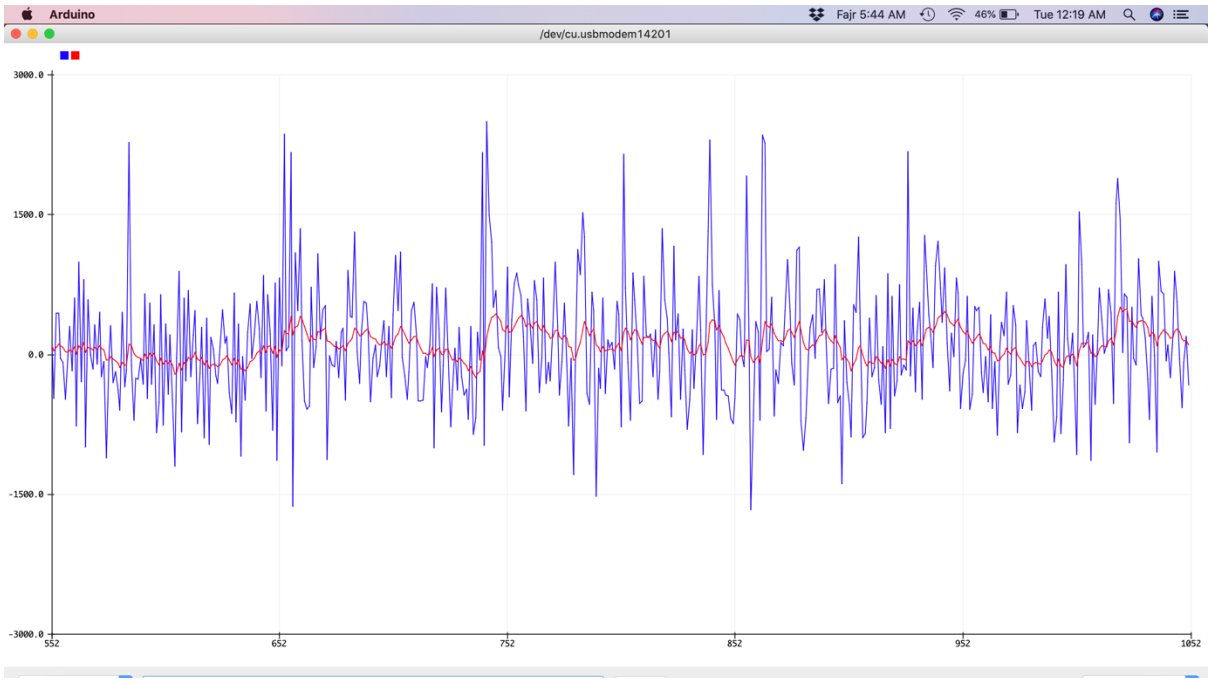
Sensor Data from Z axis of Gyro Kalman filtering



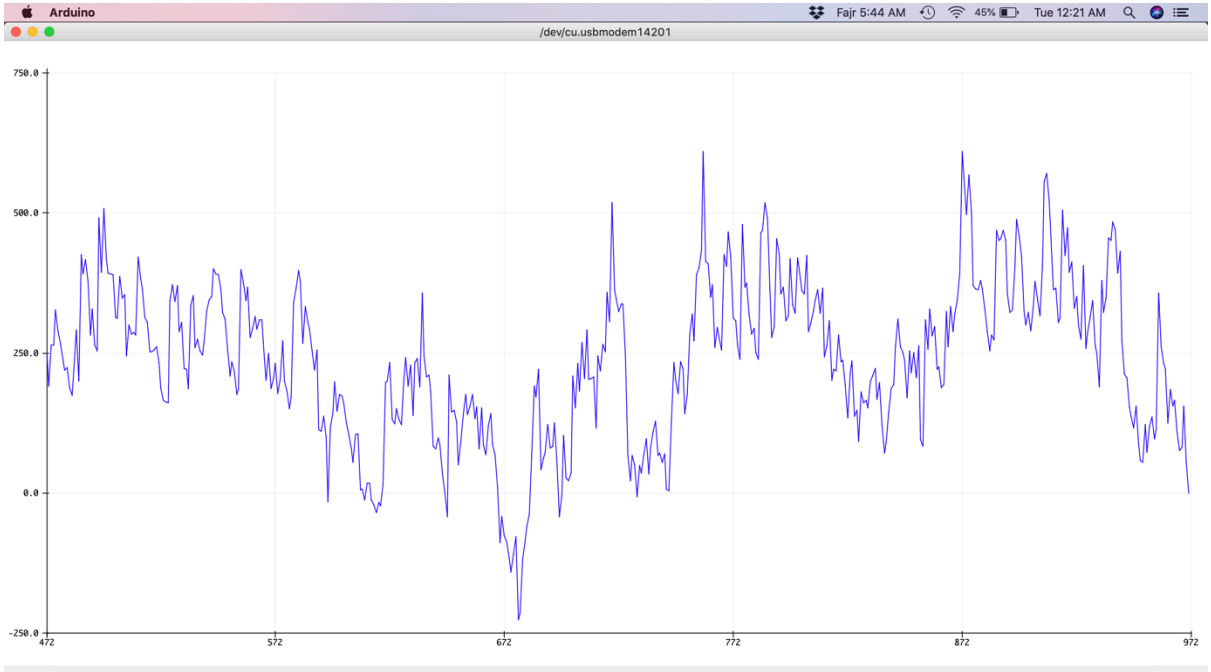
Raw Data from range 7000 md centered -4500



Bias corrected Data from range 7000 md but centered around 500 md



Both Bias corrected data in blue and filtered data in red.



Kalman filtered data with average noise range of 700 md only

Code:

```
#include <Wire.h>
#include <LSM6.h>
#define FS_2 0.061
#define FS_245 8.75

float SensorData, KalmanFilterData;
float Xt, Xt_update, Xt_prev;
float Pt, Pt_update, Pt_prev;
float Kt, R, Q;


float bias;


LSM6 imu;

void setup()
{
 Serial.begin(9600);
 Wire.begin();

 if (!imu.init())
 {
  Serial.println("Failed to detect and initialize IMU!");
  while (1);
 }

 imu.enableDefault();
 //imu.writeReg(LSM6::CTRL1_XL, 0b01011000);

 float c=0;
 float sum;
 float reading;

 while ( c<100 )
 {
  imu.read();
 reading = imu.g.z * 8.75;
 sum = reading+sum;
 c=c+1;
 }
 bias = sum/c;

 Serial.print("The sum of readings is: ");
 Serial.println(sum);
 Serial.print("The bias of readings is: ");
 Serial.println(bias);
```

```
  delay(500);

  R = 100;
  Q = 1;
  Pt_prev = 1;

  imu.read();


  SensorData=(imu.g.z * 8.75)-bias;

  Xt_prev=SensorData;


}

void loop()
{
  //read_g_z(); //for raw data only
  kalman_s();  // for filtered data



}

void kalman_s()
{
  imu.read();
  float reading = imu.g.z * 8.75;

  SensorData=reading-bias;

  Xt_update = Xt_prev;
  Pt_update = Pt_prev + Q;
  Kt = Pt_update / (Pt_update + R);
  Xt = Xt_update + ( Kt * (SensorData - Xt_update));
  Pt = (1 - Kt) * Pt_update;
  Xt_prev = Xt;
  Pt_prev = Pt;
  KalmanFilterData = Xt;
  Serial.print(SensorData);
  Serial.print(",");
  Serial.println(KalmanFilterData);
  Serial.println();
  delay(1);
}

void read_g_z()
{
  imu.read();
  Serial.println(imu.g.z*FS_245-bias);

  delay(100);
}
```