07/30/2023
Wesley Cooke

BNOOSS Relative Heading Observation

When testing the autonomous boat, we noticed an undesired change in direction and resultant spiral out of control.

Figure 7
Gps data of the waterCraft in autonomous mode. Note that the path is good during the beginning phases, but begins acting more and more weird near the half way point.

BNO055 Relative Heading Observation

07/30/2023 Wesley Cocke

If we observe the GPS and the heading coming from the BNO055, we begin to see an issue.

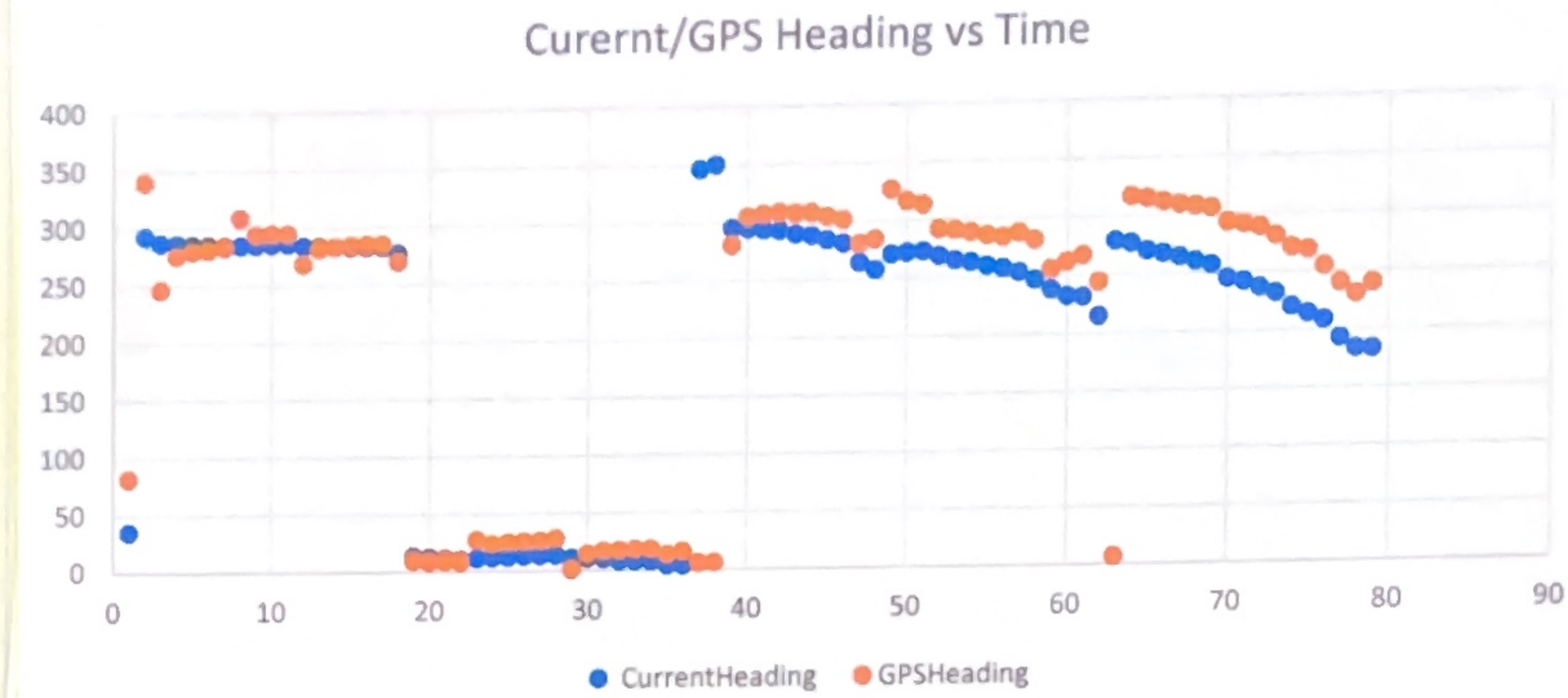

Figure 2

At the start of our program, the GPS and BNO tend to agree on what the heading is, It begins to drift apart though.

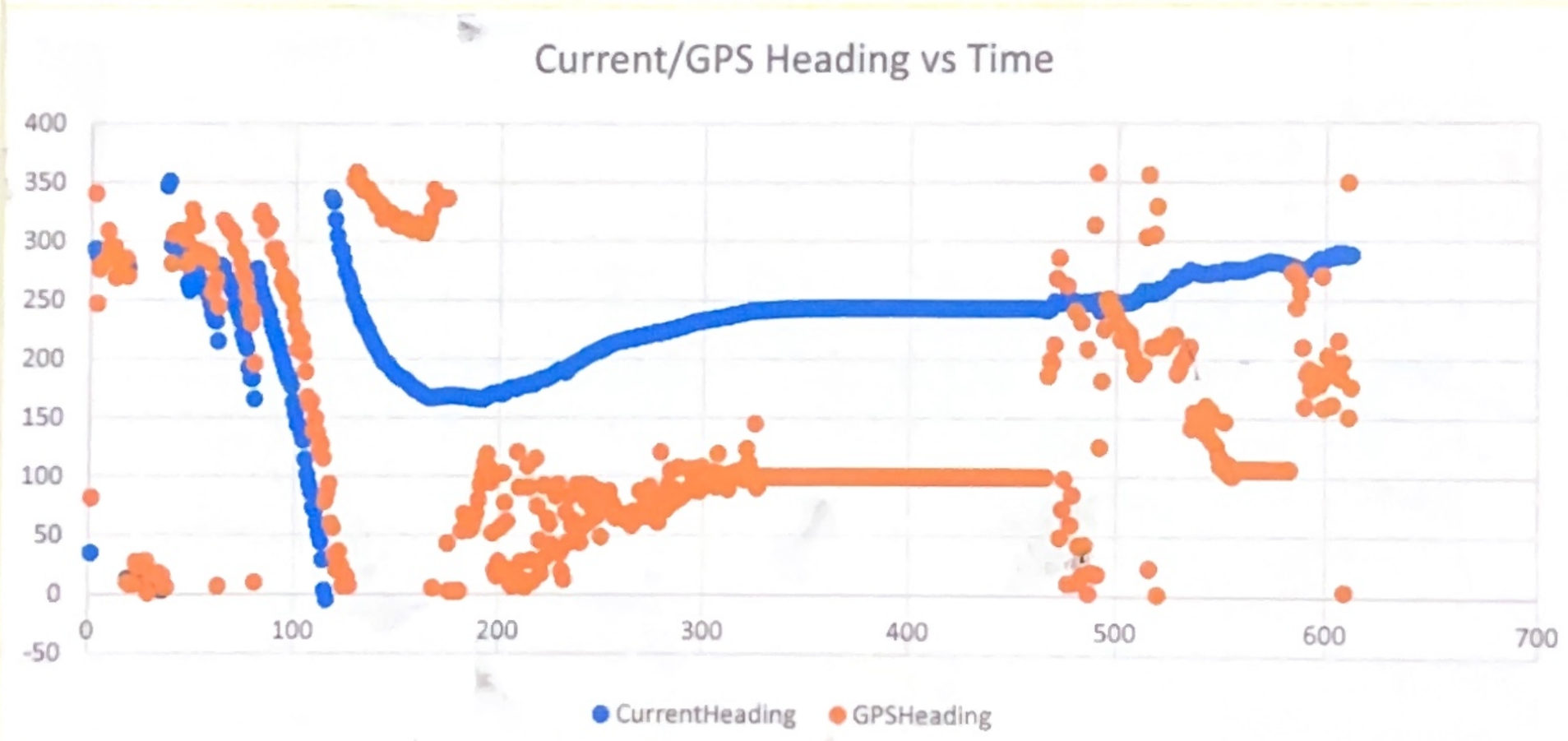

Figure 3

At sompoint the difference between the GPS and the BNO is very large.

What caused this difference?

Gyro drift may account for some of the error. However, gyro drift seems to be a small error compounded over time. Not a large jump.

Could this be gimbal Lock?...

Observations of the BNO055 by itself:

using the following program we can observe the heading reported by the BNO055 on the serial monitor.

bno_relative_test

```
#include <math.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imumaths.h>

Adafruit_BNO055 myIMU = Adafruit_BNO055();
imu::Vector<3> euler;
imu::Vector<3> gyro;
imu::Quaternion quat;
float currentHeading;
uint8_t systemCal, gyroCal, accelCal, mgCal;

void setup()
{
  Serial.begin(115200);

  if (!myIMU.begin())
  {
    Serial.println("Failed to init BNO.");
    while (1);
  }
  //  Serial.println("BNO Connected.");
  Serial.print("Heading, GyroX, GyroY, GyroZ, gyroCal, accelCal, systemCal");

  delay(1000);
  myIMU.setExtCrystalUse(true);
  myIMU.setMode(adafruit_bno055_opmode_t::OPERATION_MODE_IMUPLUS);

}
```

90 | BNO055 Relative Heading Observation | 07/30/2023

```
void loop()
{
  gyro = myIMU.getVector(Adafruit_BNO055::VECTOR_GYROSCOPE);
  quat = myIMU.getQuat();
  euler = quat.toEuler();
  myIMU.getCalibration(&systemCal, &gyroCal, &accelCal, &mgCal);

  Serial.print(euler.x() * (180/3.14), 2);
  Serial.print(", ");
  Serial.print(gyro.x(), 2);
  Serial.print(", ");
  Serial.print(gyro.y(), 2);
  Serial.print(", ");
  Serial.print(gyro.z(), 2);
  Serial.print(", ");
  Serial.print(gyroCal);
  Serial.print(", ");
  Serial.print(accelCal);
  Serial.print(", ");
  Serial.println(systemCal);

  delay(100);
}
```

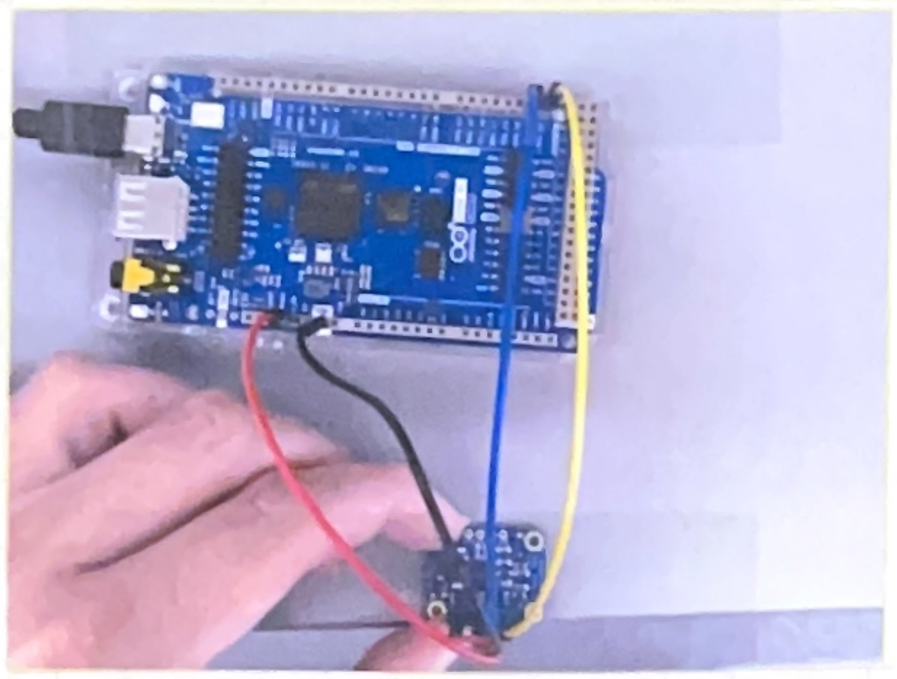

Figure 4
ORientation 1, When you power on your device, make sure the BNO 055 is in this orientation. This ensures that 0° is aligned with the table.

Procedure

Rotate the Chip about the Z axis. Bring it back to 0.
Rotate about the Y axis. Bring it back to 0.
Rotate about the X axis. Bring it back to 0.

Observe the output.

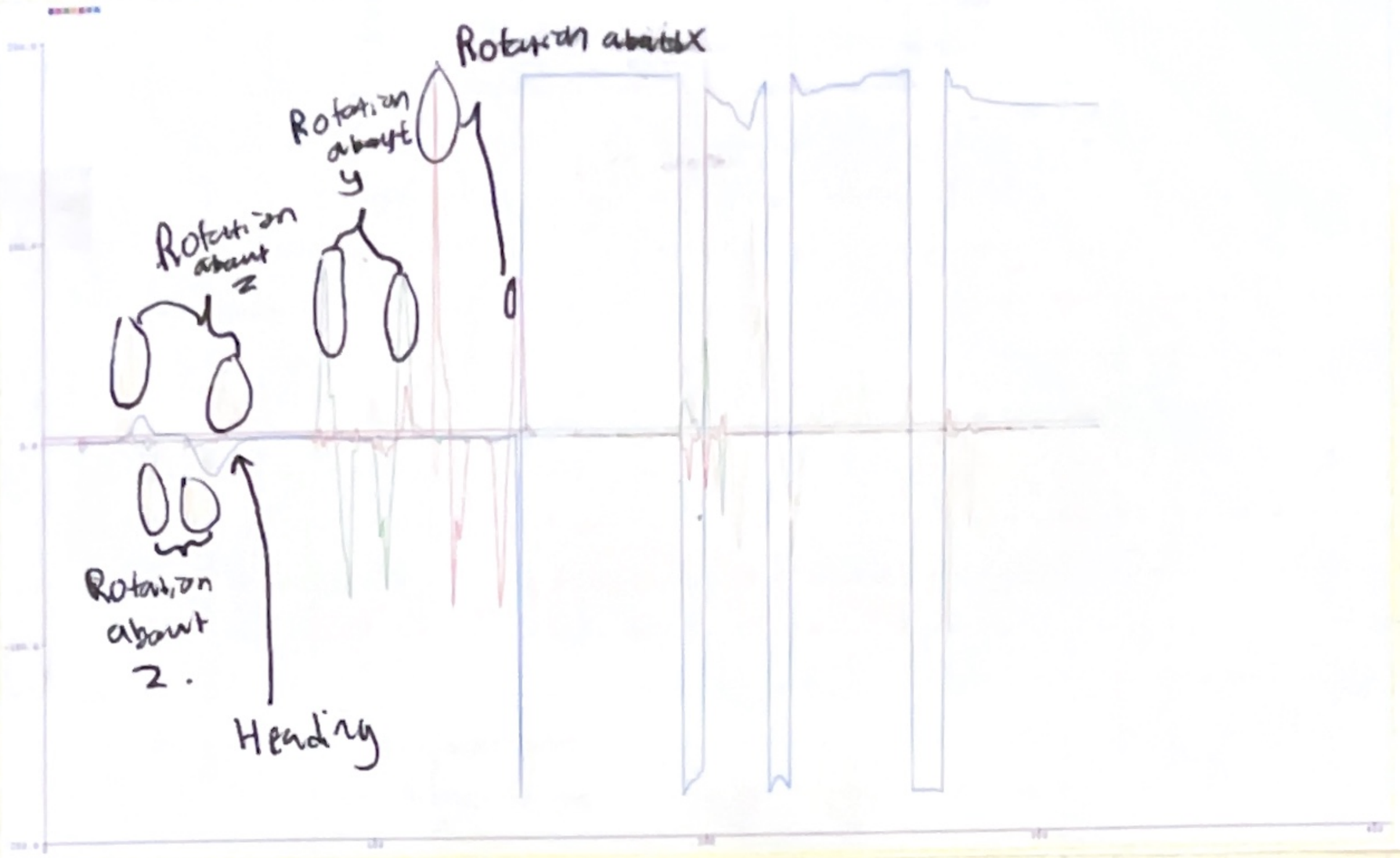

Figure 5

Starting in orientation 1 from figure 4 and following the procedure. This is the collected data.

Notice the blue line that represents heading starts around 0.

The yellow lines represent rotation about the Z-axis.
The green lines represent rotation about the y-axis.
The Red lines represent rotation about the x-axis.

Notice that at some ~~point~~ point after/during the rotation about the x-axis, our heading jumps from 0 to -180/180. This is even when we have returned it to Orientation 1.

BNO055 Relative Heading observation

07/30/2023

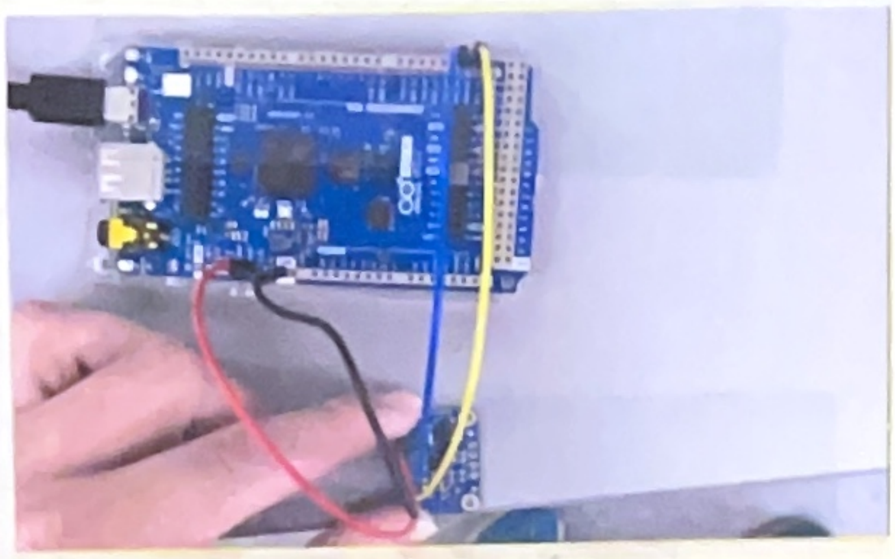

Figure 6
orientation 2.

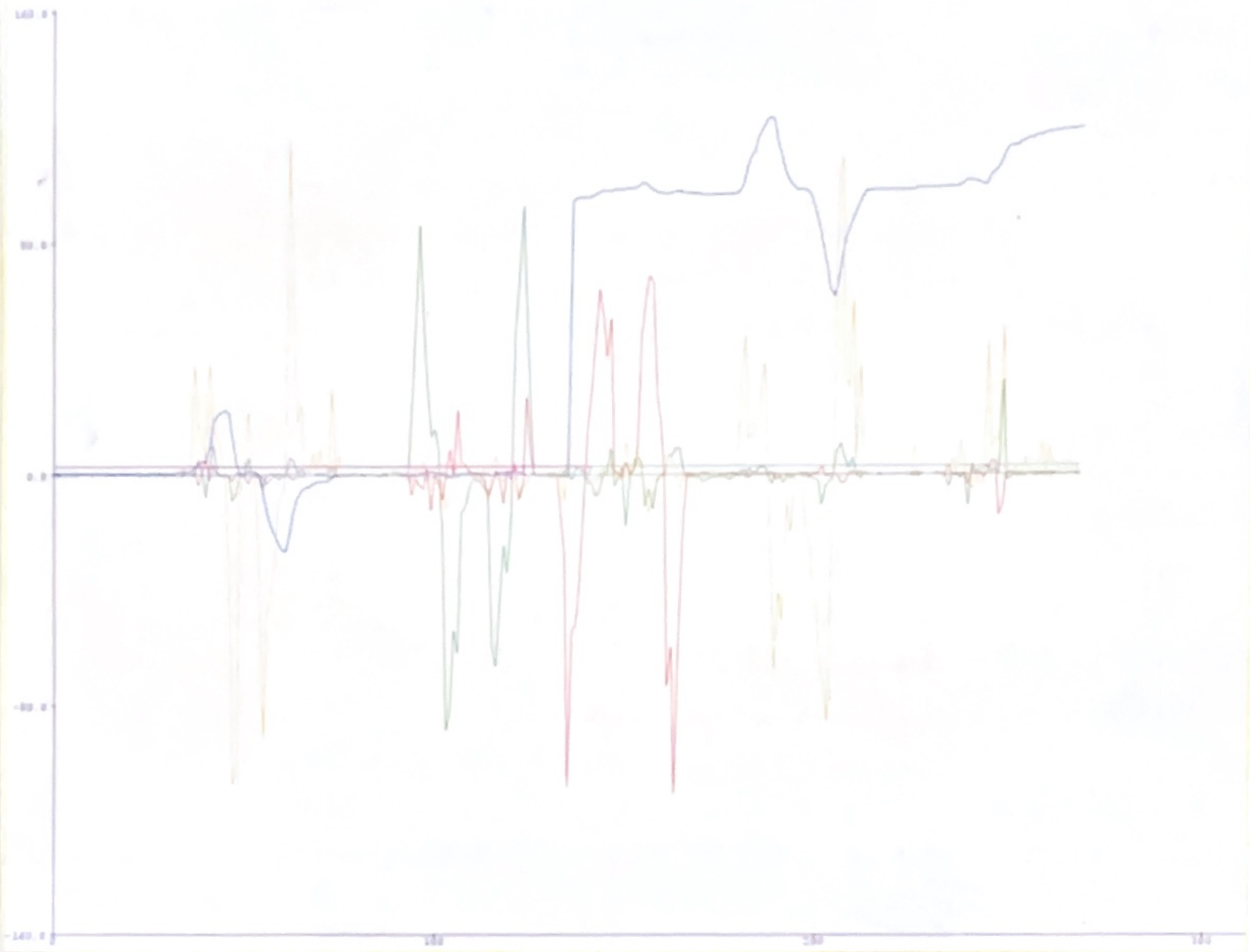

Figure 7
Data from orientation 2 in Figure 6.

Notice this time the 0° point jumps to 90°.

You can try this in all orientations.