Torque Measurements in Knee and Hip Joints in High Bar, Low Bar, and Front Squats

Jacob Warshauer

November 12, 2020

Abstract

Using Arduinos, I aim to measure the forward shift of the knees, the backward shift of the hips, the angle of the back, and the angle of the shin during the motion of several styles of squat: high bar, low bar, and front squat. These measurements can be used to calculate the torque on the knees and hips in the full range of the motions. Using several GY-521/MPU-6050 modules which contain a 3-axis gyroscope and accelerometer, the relevant data can be collected; combined with measurements of the bar height, there are numerous models of the squat in which these observations can be applied, giving estimated torques on the joints during the entire range of the movement. Current literature has made related calculations of these torques during the high bar squat, which I can use to compare my results to before drawing conclusions from the other two styles of squat I will be researching.

Introduction

There has been a large degree of controversy in sports science and biomechanics research in regards to optimal squat style, stance, and form. Studies have looked at hip and knee torque and stress separately through a high bar squat [1], the anterior chain to posterior chain stress distribution at different movement restrictions on the knee [2], the comparison of high and low bar squat muscle activations [3], and many other aspects of this widely used movement [4]. Despite the numerous experimental and theoretical results in regards to this topic, the many parameters at play lead to heavily debated conclusions.

A particular hole in this research that I have noticed is a direct comparison of torques in the three most popular styles of squat with similar conditions and methods of measurement. The position of the bar in relation to the torso varies heavily between high bar, low bar, and front squats. I hope that by taking specific measurements, I may be able to contribute to the knowledge surrounding one of the oldest and most used strength training and physical rehabilitation movements.

Technical Concepts

I plan on using five GY-521 modules positioned on the shin, back, and barbell [Fig 1]. These modules can measure the angle of the shin and back, which can be used to find torques given the weight of the barbell (this is the simplest model we can use, though these measurements can be applied to much more complex models). Adding a module to the barbell allows us to track the acceleration and bar-speed through the movement.

The GY-521 is extremely capable, and these sensors are relatively easy to communicate with. The early success of this project with this sensor and the accuracy of just one sensor positioned on the body is shown later in the section *Initial GY-521 Testing*.

The calculations take different form based on the model, but the technical challenge lies in determining angle and joint position from the sensor read-outs. Angle should not be difficult, but keeping the modules from confusing themselves when calculating displacements may prove more difficult, particularly without regular calibration.

Wiring for a single GY-521 is quite simple [Fig 2]. I currently use a button to initiate recording of the movement, and I will soon add a second button to calibrate the module.

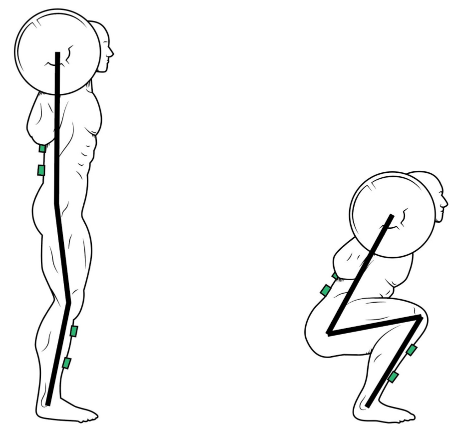


Figure 1: Sensors, in green, positioned on squatting subject. The thick lines run between the joints we are most concerned with.

I intend to have a display that will update through the motion with calculated position values so that the person performing the movement can have immediate feedback. I will cover the technical details of this once I have worked out the measurements and calculations. I currently have live data sent to serial, so I don't have an urgent need for a separate display, though it would be a nice quality-of-life feature.

Project Stages

1. Effectively measure angle with a live serial output and high rate of sampling using one GY-521 module (this has already been completed in the section *Initial GY-521 Testing*).
2. Employ the full set of sensors to measure shin angle.
3. Construct the full set of sensors to measure the back angle and bar height.
4. Test measurements with various squat forms.
5. Apply these measurements to quasi-static models.
6. Allow angular intake of angular displacement for all 3 axes so that precision in device placement can be relaxed.
7. Add bar speed and acceleration to be applied to more complex models.
8. Implement LCD or other display for live feedback without a need for the Arduino to be connected to a computer.
9. Attempt live plotting of data for display.

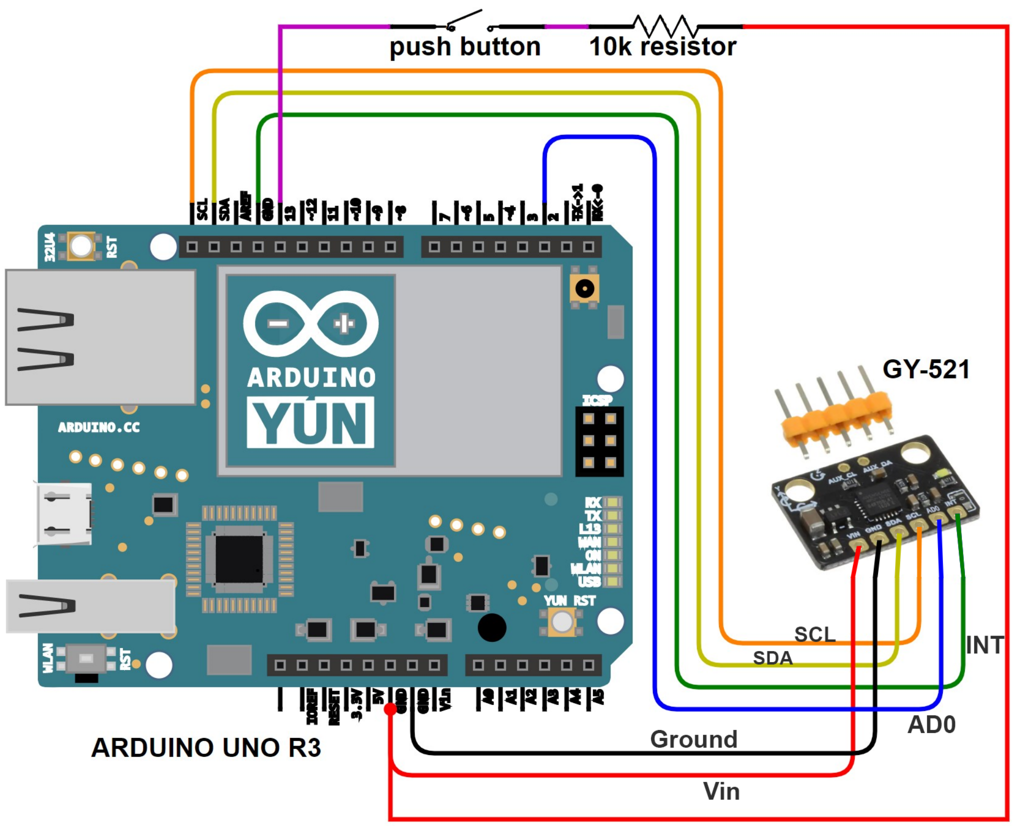


Figure 2: Schematic for single GY-521 with pushbutton for initiating recording of measurements. This is the exact set up used in the first project state detailed in *Initial GY-521 Testing*.

Initial Ultrasonic Testing

I have started this project, testing the capabilities of ultrasonic sensors in regards to detection and live updating, using a simple two sensor setup [Fig 5]. This pair of sensors was set to look at only the shin angle portion of the project: the first stage as outlined above. This code does not become complex yet, as it just uses a fit to the distances measured between the sensors to calculate the angle [Fig 3].

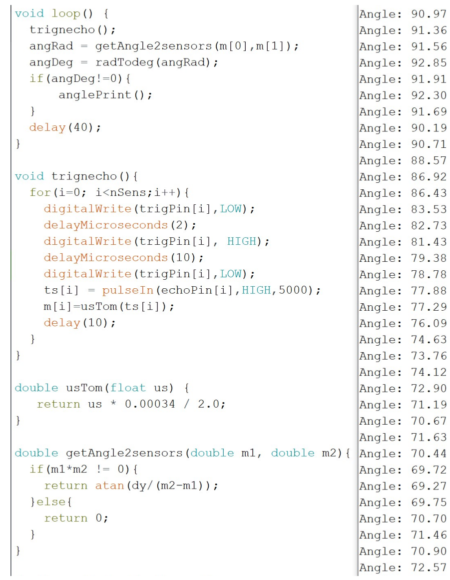
The sensors were positioned by attaching them to a cardboard box, as this is just a simple prototype [Fig 4].

The distance that the sensors will measure is limited by the maximum waiting time in the pulseIn() command. This will keep the code from detecting pulses that are not from the shin.

Issues which have arisen are the following: many zero inputs into the sensors, poor echo off soft materials, and occasional wildly incorrect measurements.

The first of these issues I anticipated. I believe this can be addressed by implementing the other two sensors in the set. In this way, when one detector does not find its pulse echo, a fit can still be found using the other detectors.

The second of these issues does not have a simple fix as this was not expected. Pants and even my bare leg proved to disperse the pressure waves too much for consistent measurements.



```
void loop() {
  trignecho();
  angRad = getAngle2sensors(m[0],m[1]);
  angDeg = radTodeg(angRad);
  if(angDeg!=0){
      anglePrint();
  }
  delay(40);
}

void trignecho(){
  for(i=0; i<nSens;i++){
    digitalWrite(trigPin[i],LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin[i], HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin[i],LOW);
    ts[i] = pulseIn(echoPin[i],HIGH,5000);
    m[i]=usTom(ts[i]);
    delay(10);
  }
}

double usTom(float us) {
   return us * 0.00034 / 2.0;
}

double getAngle2sensors(double m1, double m2){
  if(m1*m2 != 0){
    return atan(dy/(m2-m1));
  }else{
    return 0;
  }
}
```

```
Angle: 90.97
Angle: 91.36
Angle: 91.56
Angle: 92.85
Angle: 91.91
Angle: 92.30
Angle: 91.69
Angle: 90.19
Angle: 90.71
Angle: 88.57
Angle: 86.92
Angle: 86.43
Angle: 83.53
Angle: 82.73
Angle: 81.43
Angle: 79.38
Angle: 78.78
Angle: 77.88
Angle: 77.29
Angle: 76.09
Angle: 74.63
Angle: 73.76
Angle: 74.12
Angle: 72.90
Angle: 71.19
Angle: 70.67
Angle: 71.63
Angle: 70.44
Angle: 69.72
Angle: 69.27
Angle: 69.75
Angle: 70.70
Angle: 71.46
Angle: 70.90
Angle: 72.57
```

Figure 3: Ultrasonic testing angle code on the left, serial output on the right.

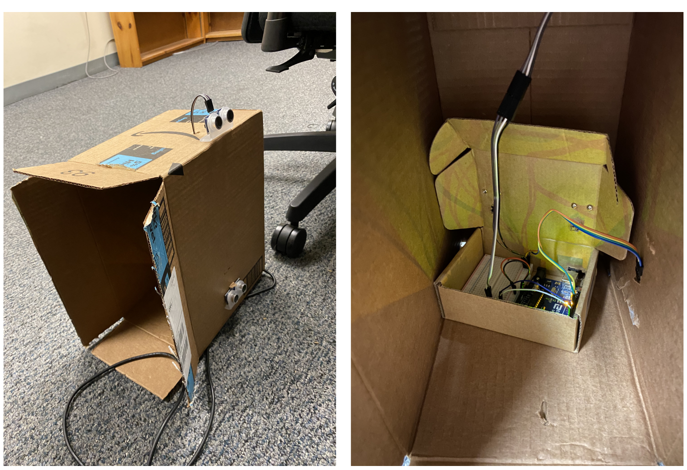


Figure 4: Ultrasonic setup for initial testing section.

The soft material of the pants is understandable, but I did not expect the hair of my leg to cause this issue as well. I addressed this in my tests by wrapping my leg in packaging, which can be seen in the video on the site I have made for this project [5].

The final issue can be seen in the plots below [Fig 6]. Occasionally, there would be very large jumps measured by the sensors. These cause issues for the angle measurements, and thus will cause problems for our models. As a temporary fix, I have written simple code to remove outliers [Fig 7].

In conclusion, the initial tests have been illuminating as far as the limitations of the sensors. Though ultrasonic sensors are promising, I believe I will change my plan to use GY-521 sensors instead and use my ultrasonic sensors for the calibration of my GY-521 modules.

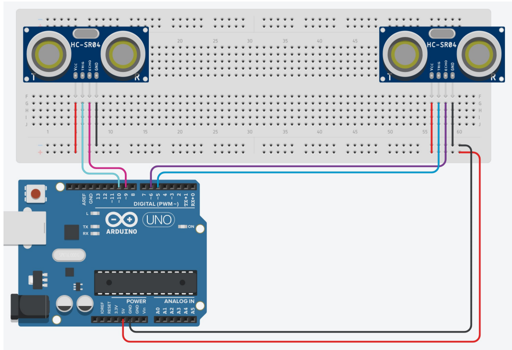


Figure 5: Tinkercad schematic of simple two ultrasonic setup used for initial ultrasonic testing.

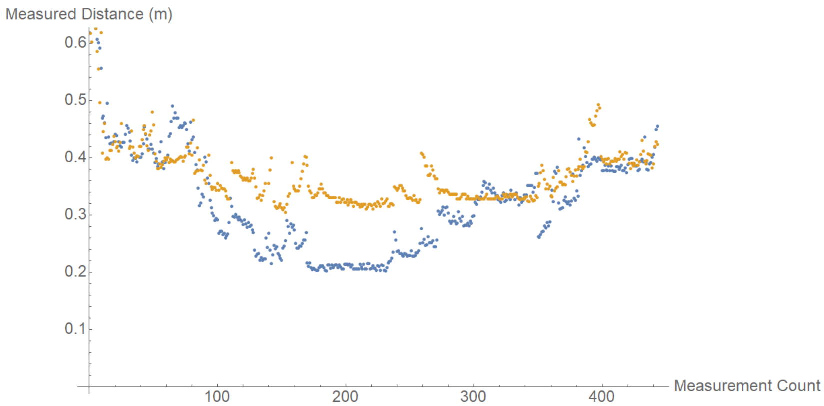


Figure 6: Sensor output vs measurement count for initial ultrasonic test 624. Blue dots are the top sensor, orange dots are the bottom sensor.

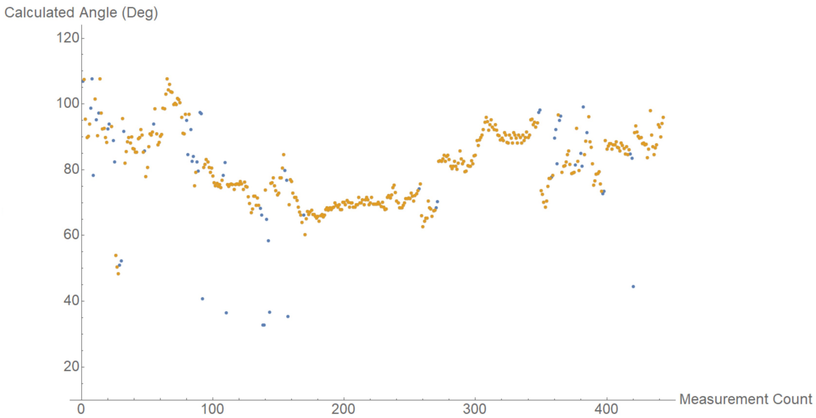


Figure 7: Calculated angle vs measurement count for initial ultrasonic test 624. Blue dots are raw data, orange dots are remaining data after outliers have been removed.

Stage One: Initial GY-521 Testing

This last week I stumbled across GY-521 modules. Looking into them, I found they are actually quite accurate, and I decided to see how effective these will be for my study. I used one connected to my breadboard with a button to control the recording of measurements. The code for the GY-521 is certainly more complex than the ultrasonic sensors; however, it is certainly not as difficult as I had predicted [Fig 9].

The device is attached to the subject's shin, oriented so that the angle change is entirely taken into account on by a single axis measurement [Fig 8]. I do plan on using all of the 3-axis measurement capabilities to allow positioning on the shin to be less precise; however, I will leave this to later stages of the project.

Promisingly, at the time this is being written, I have encountered no major issues in my usage of the GY-521. I am able to connect only two GY-521 modules to one Arduino Uno, so I will require three Arduino Unos to allow for the five planned GY-521 modules.

I used no delay in my first test so that I could observe the exact time for the entire measurement to run one loop. I ran my single sensor code for $64.74s$ and 6951 angle measurements were taken. I compared the rate of measurements at the start of the code running to the end, and I found no significant variation in the speed. With no delay, the current configuration takes 107.36 measurements per second, $9.314ms$ per step. Due to this, I applied a $10686\mu s$ delay so that the device reads out at $500Hz$.

I attached the device to the cardboard armor that I created for the ultrasonic sensors [Fig 8]. I tested this device performing a squat in the test I labeled test 207. The plotted data showed extreme consistency [Fig 10], and a video of this test can be seen on the website made for this project [5].

This initial test with the GY-521 serves as a nearly completed Stage One for this project. I've been able to show that this module will be perfectly suited to this study, and I look forward to continuing to design and build this experimental setup.

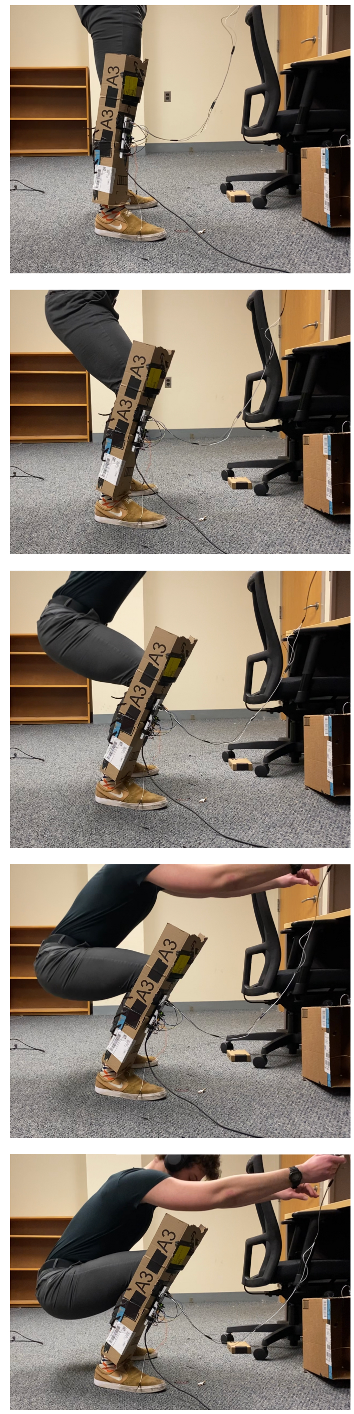


Figure 8: GY-521 Test 207 images through a squat movement.

```
void setup(){
  pinMode(buttPin,INPUT);
  Serial.begin(9600);
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
}

void loop(){
  if(digitalRead(buttPin)==HIGH){
    readGY521();
    Serial.println(calcAngle(accRaw[0],accRaw[1],accRaw[2]),2);
  }
  delayMicroseconds(10686);
}

void readGY521 (){
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true);
  acX = Wire.read()<<8|Wire.read();
  acY = Wire.read()<<8|Wire.read();
  acZ = Wire.read()<<8|Wire.read();
  accRaw[0]=acX;
  accRaw[1]=acY;
  accRaw[2]=acZ;
}

double calcAngle (int16_t xAc, int16_t yAc, int16_t zAc){
  int yAng = map(yAc,minVal,maxVal,lmap,hmap);
  int zAng = map(zAc,minVal,maxVal,lmap,hmap);
  return RAD_TO_DEG * (-atan2(-yAng, -zAng));;
}
```

```
78.25
76.77
74.72
75.22
77.91
77.75
78.50
76.58
76.23
76.87
77.65
74.31
68.45
64.98
65.37
70.41
70.86
67.54
64.24
57.39
55.24
57.11
58.62
57.01
53.66
47.22
44.03
43.12
42.04
42.98
44.72
48.18
48.37
48.01
46.16
46.10
```

Figure 9: Angle measurement code for single GY-521 on left, serial output on right.

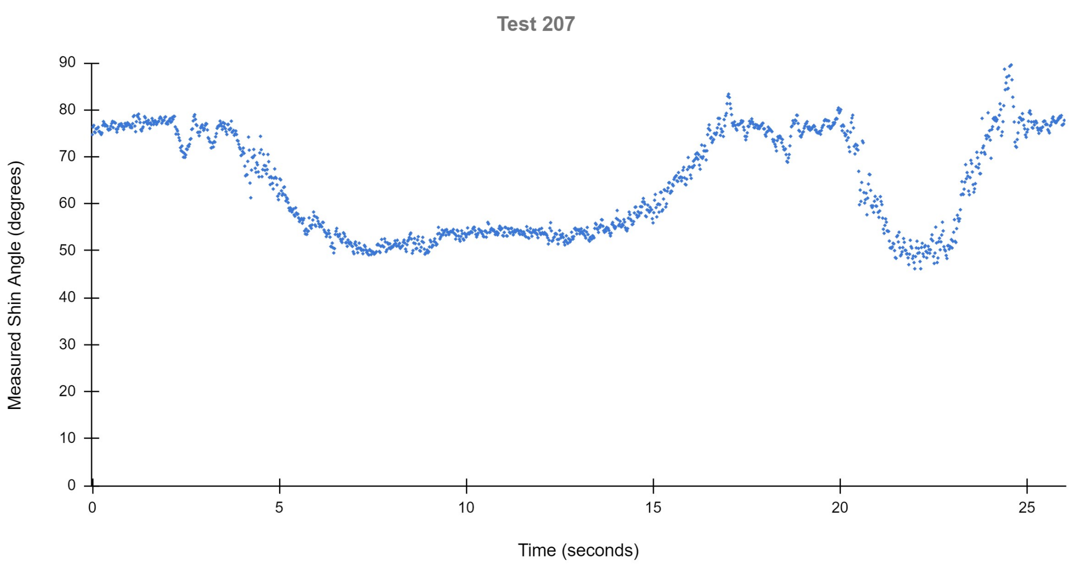


Figure 10: Plot of measured angles during test 207. One slow squat and one fast squat movement were performed during this test. One can see the measurements in this setup are significantly better than those in Figure 7.

Components

The measurements do not require specialized electronics. I will require:

- 3x Arduino Uno R3.
- 5x GY-521 Module.
- Materials to make the device wearable.

Proposal Conclusion

I believe that this is an excellent project in order to present novel measurements which will contribute to the understanding of squat mechanics. This type of data can be applied and compared to some of the numerous models generated in other studies on this topic. The measurements are simple in nature, and the final electronics will also have use in monitoring form consistency for an athlete. Live feedback through a movement will allow the user to actively adjust their form during a session.

References

[1] KC Ng, G Rouhi, M Lamontagne, PE Beaulé. *"Finite Element Analysis Examining the Effects of Cam FAI on Hip Joint Mechanical Loading Using Subject-Specific Geometries During Standing and Maximum Squat."* HSS J. 2012, 8(3):206-12.

[2] A Fry, J Smith, B Schilling. *"Effect of Knee Position on Hip and Knee Torques During the Barbell Squat."* JSCR 2003, 17(4), 629.

[3] J Logar , M Kleva , U Marusic , M Supej, M Gerzevic. *"Differences in the Knee Torque Between High - and Low - Bar Back Squat Techniques: A Pilot Study."* Annales Kinesiologiae 2014, 5 ,2.

[4] P Comfort, P Kasim. *"Optimizing Squat Technique."* SCJ 2007, 29, 6.

[5] J Warshauer, *"Torque Measurements in Knee and Hip Joints in High Bar, Low Bar, and Front Squats."* BU Google Site, 2020.