Tutting Dance with Flex Sensors and IMU

Lucas Lin, Amr Mohamed
EE267 - Virtual Reality

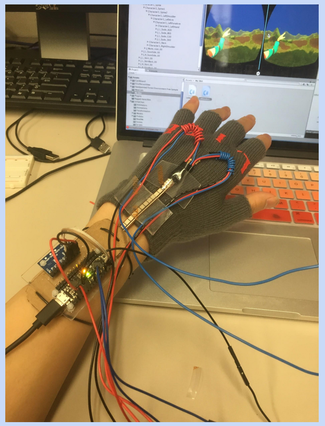
Motivation

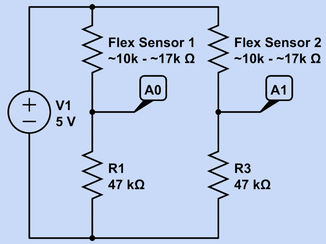
Tutting is a form of dance that involves the use of the body--usually the arms and hands--to create geometric shapes and usually with right angles. This style of dance is usually focused on making sure the shapes created by the limbs are as precise as possible. Additionally, these motions generally are done to the beat and each position quickly flows from one position to another. The goal of our VR demo is to develop a hand/arm tracking program to teach the dance style of tutting by mimicry.

Related Work

Previous work has seen the use of inertial (and magnetic) measurement units for hand tracking and rendering. One example of this work has used a glove outfitted with six IMUs to track finger and hand positions.[1] It has been noted, however, that use of IMUs to track position leads to drift, so we decided to design our approach on hand and arm angles rather than absolute position. Additionally, we use flex sensors for further disambiguation of wrist angle vs. hand angles. These flex sensors vary their resistance based on their bend and have also been used in a glove construct to track fingers.[2]

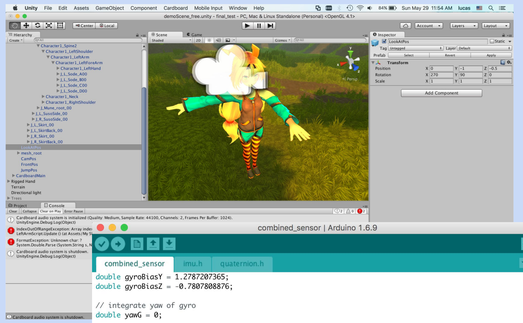
Hand Angle Tracking

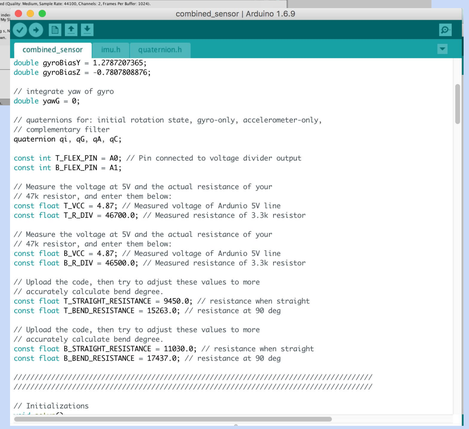




Voltage divider for flex sensors

We use the Spectra Symbol flex sensors to keep track of the wrist angle and the IMU on the back of the forearm to calculate the angle of the elbow length-wise rotation of the forearm. Since flex sensors only vary resistance reliably in one bend direction, we use two flex sensors in conjunction on the back of the wrist to obtain both the forward bend as well as backwards bend. The circuit used for the flex sensors is a simple voltage divider and we use pre-recorded calibrating values to estimate the degree of bend from the input voltage from the divider. As for the elbow angle and forearm rotation, we transformed the quaternion from the IMU into Euler angles and then calibrated the values based on the maximum and minimum rotation of the arm/wrist.






```
double gyroBiasY = 1.2787207365;
double gyroBiasZ = -0.7807808876;

// integrate yaw of gyro
double yawG = 0;

// quaternions for: initial rotation state, gyro-only, accelerometer-only,
// complementary filter
quaternion qi, qG, qA, qC;

const int T_FLEX_PIN = A0; // Pin connected to voltage divider output
const int B_FLEX_PIN = A1;

// Measure the voltage at 5V and the actual resistance of your
// 47k resistor, and enter them below:
const float T_VCC = 4.87; // Measured voltage of Ardunio 5V line
const float T_R_DIV = 46700.0; // Measured resistance of 3.3k resistor

// Measure the voltage at 5V and the actual resistance of your
// 47k resistor, and enter them below:
const float B_VCC = 4.87; // Measured voltage of Ardunio 5V line
const float B_R_DIV = 46500.0; // Measured resistance of 3.3k resistor

// Upload the code, then try to adjust these values to more
// accurately calculate bend degree.
const float T_STRAIGHT_RESISTANCE = 9450.0; // resistance when straight
const float T_BEND_RESISTANCE = 15263.0; // resistance at 90 deg

// Upload the code, then try to adjust these values to more
// accurately calculate bend degree.
const float B_STRAIGHT_RESISTANCE = 11030.0; // resistance when straight
const float B_BEND_RESISTANCE = 17437.0; // resistance at 90 deg

////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////

// Initializations
```

Results

We were able to successfully use the data from the flex sensors and the IMU to control a rigged arm model in Unity. In order to prevent some snapping of the model arm due to potential fluctuations of incoming data--like a value of 360 degrees wrapping around to 0 or 1 degree--we limit the range of motion on the arm to what is normally physically possible and typically used in a tutting routine. One major concern about the current algorithm is that it is fairly dependent on the manual calibration being correct as the degree bends of the various angles are estimated using base values that we measured, so a future improvement could be the automation of calibration at the beginning of each demo round.

Due to a lack of free but still properly rigged models of the arm and hand, we used the free Unity asset “Unity-chan” pictured to the right as the user avatar. By locking the skeleton of this avatar with the first person controller and warping the arm in front of the camera, we were able to create a first person perspective onto this model.

In addition to having “Unity-chan” as the user avatar, the virtual environment we created for the demo also includes a recording of a simple tutting routine that is designed for the user to follow along in virtual reality. The user is to follow the routine as best as possible and receive a metric on how accurate they were at the end.

References

[1] Lisini Baldi, T., Mohammadi, M., Scheggi, S., & Prattichizzo, D. (2015, June). Using inertial and magnetic sensors for hand tracking and rendering in wearable haptics. In World Haptics Conference (WHC), 2015 IEEE (pp. 381-387). IEEE.

[2] Soral, L., & Ambikapathy, M. Hand-talk Gloves with Flex Sensor: A Review. *International Journal of Engineering Science Invention*, *2*(4). 43-46.