Foot-strike Analyzer and Trainer
Zachary Hand
Faculty Advisor: Dr. Barry Grossman, Dept of Electrical Engineering, Florida Institute of Technology

A fitness accessory for runners

**Goal**

The goal of this project is to implement a portable system that is capable of monitoring how a runner's foot interacts with the ground. It provides real-time feedback allowing the user to modify their running form to fit their fitness goals.

**Project Overview**

The project consists of three subsystems:

- Piezoelectric Sensor Array
- Data acquisition/processing/TX
- RX and User Interface (Smartphone Application)

The sensor array is embedded in a running shoe, and the conditioned output connects to a Bluetooth equipped microcontroller. When the shoe makes contact with the ground, the impact is detected and localized, and this information is transmitted via Bluetooth to an Android smartphone.

**Sensor Array**

- Consists of two piezoelectric sensing elements
- Discriminates between heel-strikes and forefoot-strikes

**Data Acquisition**

- RFduino microcontroller transmits sensor data and receives commands via onboard Bluetooth LE
- Auto calibration code sample:

  ```java
  //Auto Calibrate
  if (cal == 1) {
    delay(200);
    //Reads analog input port
    sensorReading = analogRead(sensor1);
    //Initializes values
    threshold1 = sensorReading;
    threshold2 = sensorReading;
    //Counts calibrations
    for (int i = 0; i < 200; i++) {
      //Takes 200 samples from sensor1
      sensorReading = analogRead(sensor1);
      //Finds high value
      if (sensorReading > threshold1) threshold1 = sensorReading;
      //Finds low value
      else if (sensorReading < threshold2) threshold2 = sensorReading;
    }
    //Wait 10ms between readings
    delay(100); //Wait 10ms between readings
    //Paging added to reduce false positives
    threshold1 = threshold1 + padding;
    threshold2 = threshold2 - padding;
  }
  ```

**Android Application**

- Displays and records foot-strike and cadence (foot-strikes per minute) data
- Wireless Control:
  - Start/Stop data collection
  - Auto or Manual sensor calibrate
- Optional user feedback:
  - Cadence: Audible alarm if value deviates from user defined range
  - Heel-strikes: Vibrational feedback
- Sample of cadence alarm code:

  ```java
  if (cadence_enable) //Option selected
  {
    if (cadence_upper && (cadence_upper > cadence_upper && cadenceViolation > 4)) {
      //Checks if an upper limit has been entered, the limit has been exceeded
      if (out of limit for at least 5 events
        cadenceViolation = 0; //reset counter
        //Load sound file
        mp = MediaPlayer.create(this, R.raw.sound1);
        mp.setOnCompletionListener(new MediaPlayer.OnCompletionListener()
        {
            @Override
            public void onCompletion(MediaPlayer mp)
            {
                Toast.makeText(getApplicationContext(), "Self destruct", Toast.LENGTH_SHORT).show();
                mp.reset();
                mp.release();
                mp = null; // free all
            }
        });
    } //End if condition
  }
  ```

---

**Northrop Grumman**

Engineering & Science Student Design Showcase at Florida Institute of Technology