Feasibility Testing Report – Stair Detection with filtering

Team: P15001: Active Ankle Foot Orthotic
Engineer: Jared Green– Mechanical Engineer

Test Date: October 9, 2014 – October 15, 2014

Related System: Sense Conditions

This test was an analysis of sensor position to help sense changes in terrain. When stairs get detected there should be a jump in sensor voltage which represents and object approaching.

Testing Procedure

The first part was involved connecting a GP2Y0A02YK0F distance sensor to the Arduino and running a data acquisition code. Connection from the distance sensor to the Arduino is shown in Figure 1.

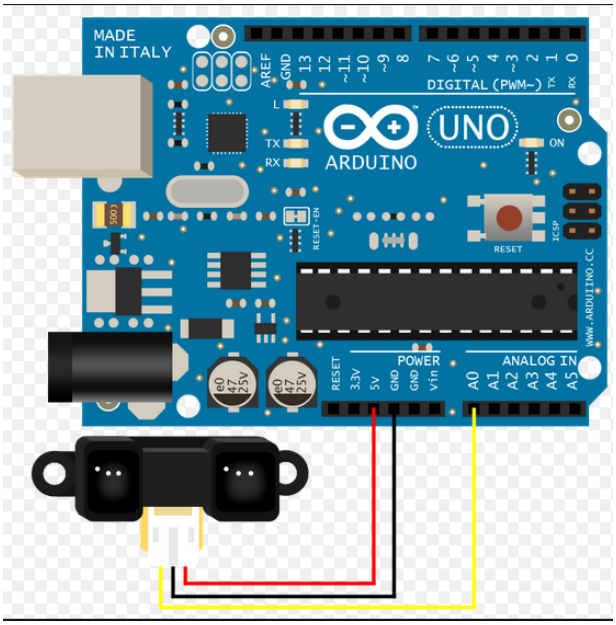

Figure 1: Distance sensor, Arduino connection

Once the sensor started working, several gait cycle trials were performed. These trials included walking up to a wall on levelground as illustrated below.

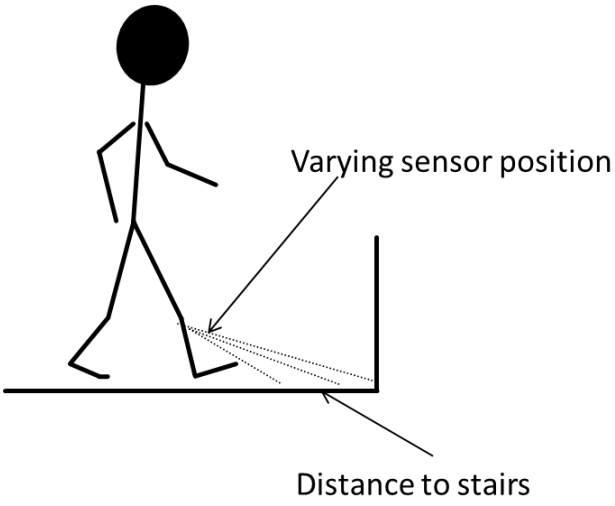

Figure 2: Illustration of testing

Results

A few preliminary trials were performed using the current gait monitor as currently assembled as is.

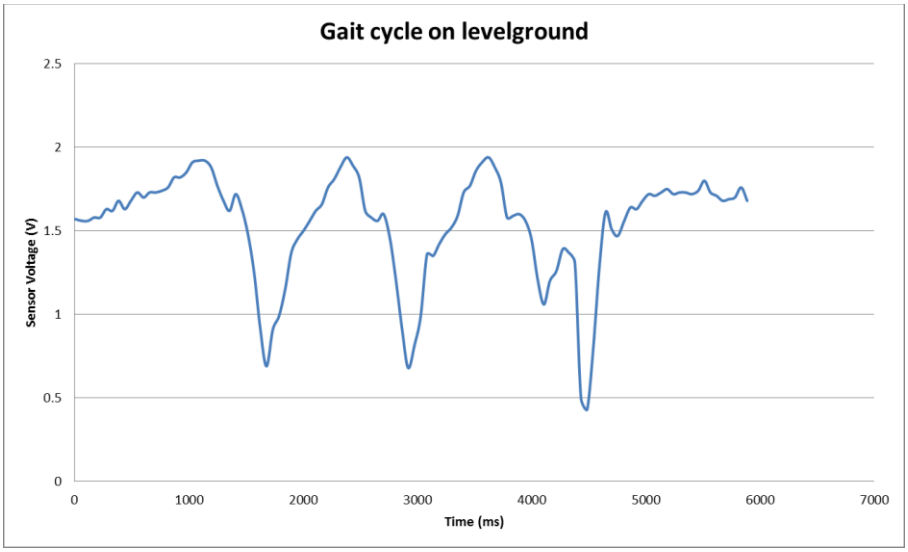

Figure 3: Preliminary trial gait cycle

Although a typical gait cycle was coming out reasonable, the sensor would not pick up a stair approaching until just a few inches from the wall. In addition, there was some occasional spike occurring so an Arduino smoothing function was necessary to smooth out data.

In order to improve the system to detecting stairs, sensor position would have to be measured and adjusted so that it can sense stairs from a farther distance. Tilt angles were measured from the vertical line to the base of the sensor. A set of small step trials were performed while varying the angles to minimize changes in amplitudes.

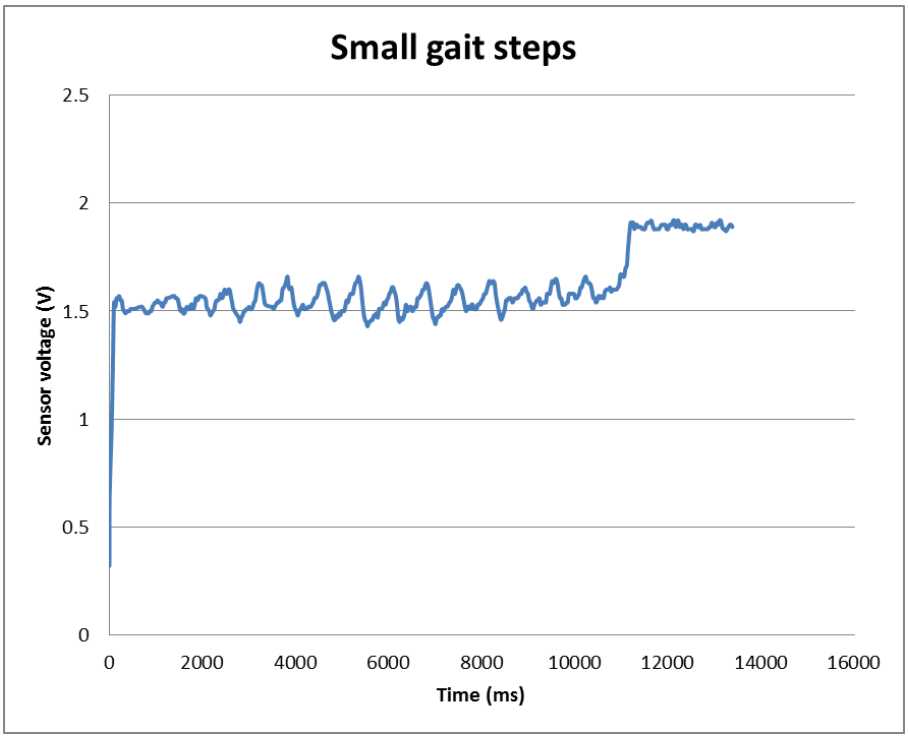

Figure 4: Stair detecting taking small steps with a 36 degree tilt angle

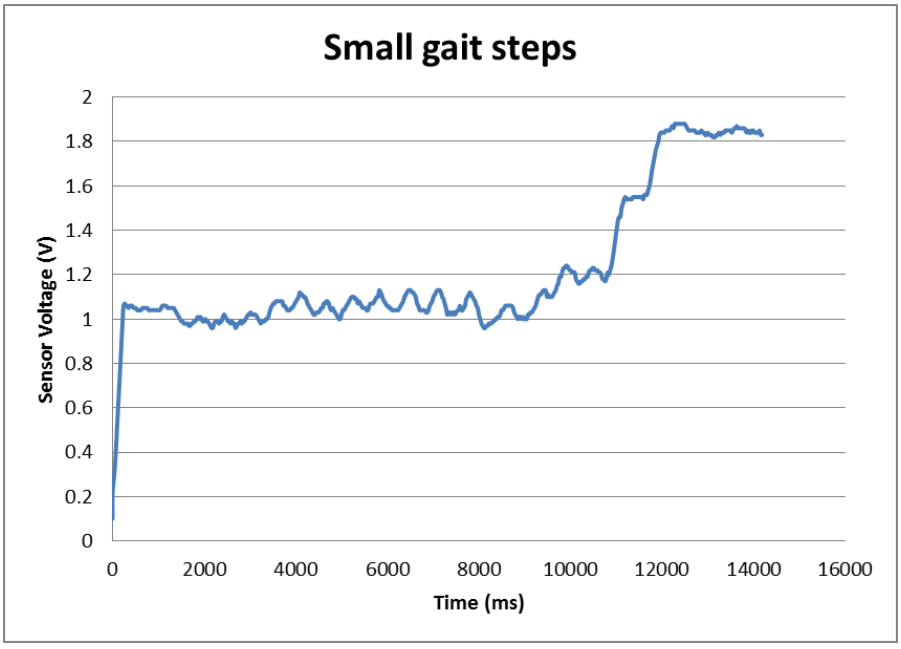

Figure 5: Stair detecting taking small steps with a 23 degree tilt angle

From the small step trials, the forward distance sensor definitely needed some repositioning in order to help sense stairs from a much farther distance as before. Also, a smoothing function was added to the Arduino code to help eliminate spikes. This smoothing function takes every reading per delay in between reads and puts them in an array of 5 readings. Every new reading replaces the oldest reading on the array and the array is then averaged to give the output for a gait cycle. Finally using the typical 23 degree tilt, a few standard gait cycles were performed on levelground and up a couple stairs.

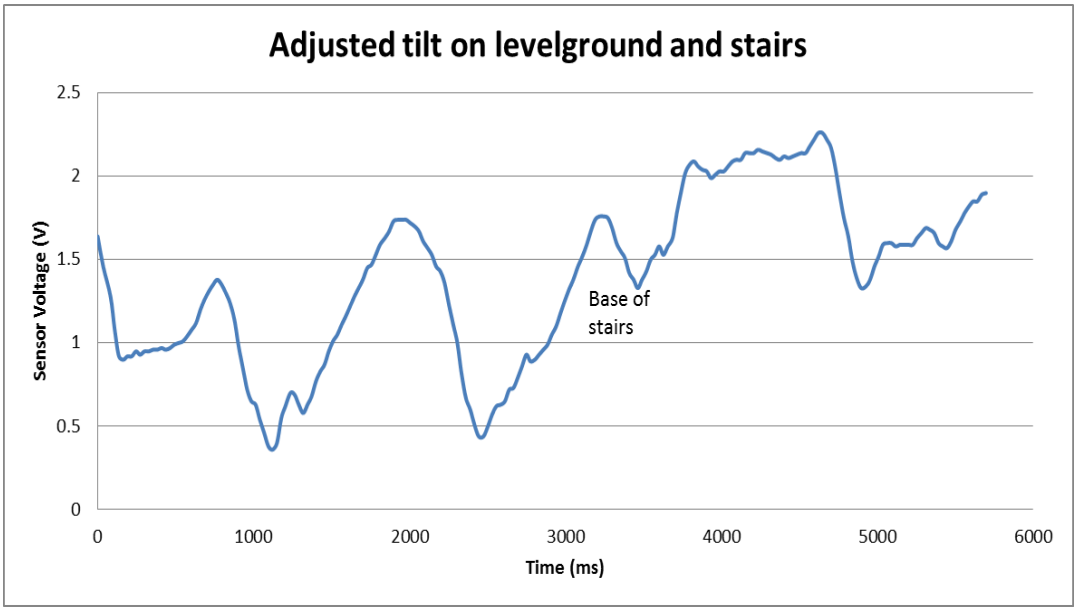

Figure 6: Levelground with stair detection and a stair cycle

From these test results, the sensor were detecting stairs from around 1 foot which is a much better distance at sensing stairs that before the sensor was adjusted.

With a typical sensor position and a built in software smoothing function set, a hardware feature filter schematic was added in order to assist the software filtration process. This hardware feature includes a 10 µF capacitor connected from output of the filter to ground. This capacitor value was chosen based off of a response time test that the sensors feature.

A few gait trials were then performed and their results are shown below.

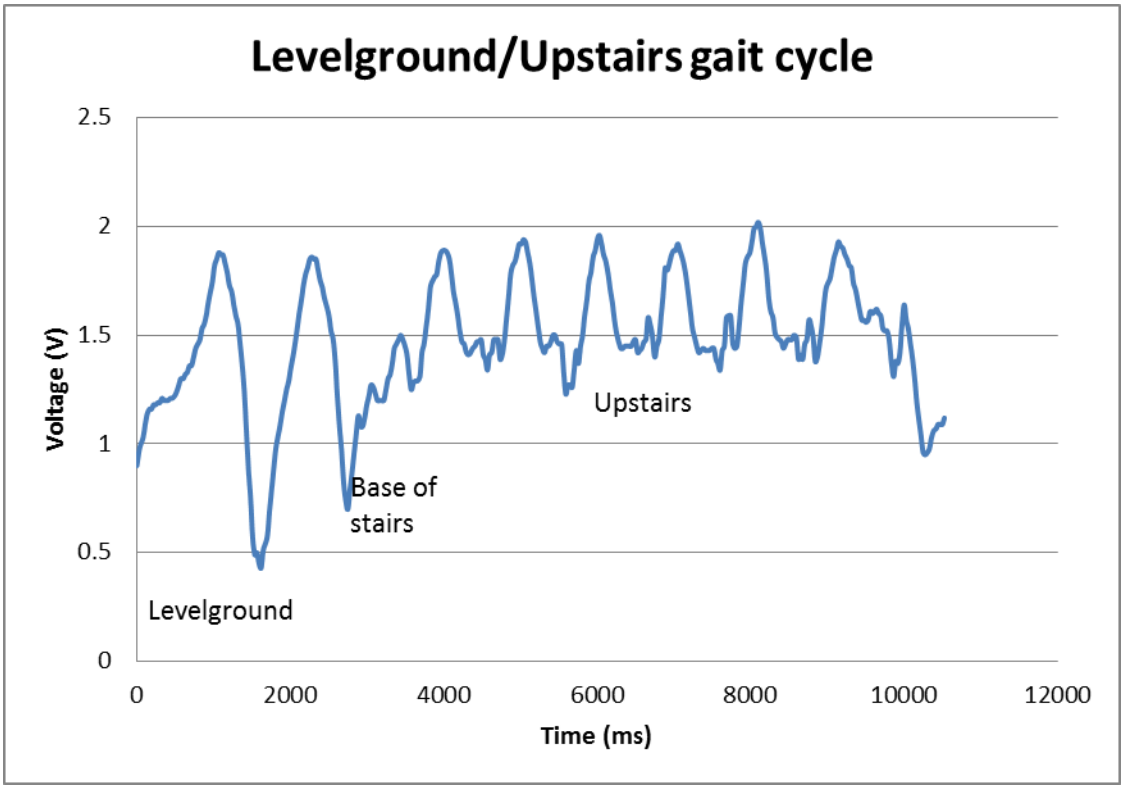

Figure 7: Hardware filtered cycle: Levelground with stair detection and upstairs

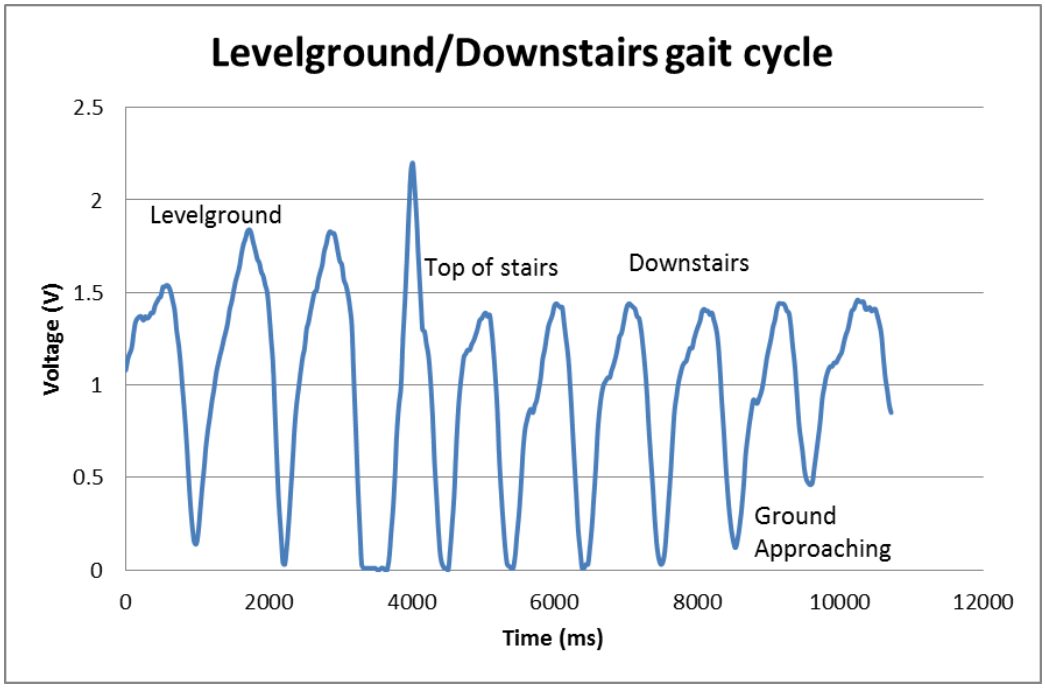

Figure 8: Hardware filtered cycle: Levelground with ledge detection and downstairs

Applying the hardware filter really helped filter out more spikes especially when ascending and descending stairs. Notice that on the downstairs cycles, the base of the gait cycles get higher. This is due to the sensors sensing the terrain that's being approached

Conclusions

The results tell us that the GP2Y0A02YK0F distance sensor can sense changes in terrain when a test subject is walking at a standard gait cycle. When the sensor is positioned correctly, changes can be detected which can ultimately predict the terrain that a test subject is walking and going to be walking on. Both hardware and software filtration features can help filter out spikes which can result in data coming out smoothly.

Next Steps

The next step would be to analyze timing for both the distance sensing and the compression sensing.

Reference

http://www.digikey.com/product-detail/en/GP2Y0A02YK0F/425-2062-ND/720167?WT.srch=1&WT.medium=cpc&WT.mc_id=IQ66882679-VQ2-g-VQ6-53963506155-VQ15-1t1-VQ16-c

http://bildr.org/2011/03/various-proximity-sensors-arduino/