Using an Accelerometer to Measure Force and Acceleration

Software Design Document

Jamis Moy & Peter Duy Tran

December 18, 2015

Table of Contents

1) Introduction

a. Purpose

This document is meant to accompany to software that will be used to help measure the acceleration, velocity and displacement if a wii controller.

b. Scope

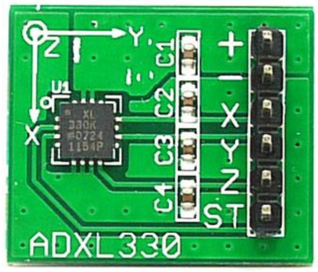
This Software is meant to wirelessly analyze the acceleration so that the user can essentially track the position of the wii controller in a three dimensional plot.

2) System Overview

The main purpose of this project is to be able to track the wii controller in a 3 dimensional plot in the form of a line graph. It is well known that the acceleration of an object can be integrated once to measure velocity, then integrated again to measure displacement. Therefore by making use of the accelerometers, we can translate the acceleration into displacement data. There will also be a MATLAB script that will allow you to compensate for the g values by essentially "removing gravity" from your acceleration data.

The type of accelerometer that will be used is a Wii controller. The main reason for using this device is because it is widely available (a lot people already own a Wii), it has the capability to measure acceleration in all three axis, and has Bluetooth connectivity which makes wireless data acquisition possible.

The acceleration data will be integrated using the trapezoidal approximation technique to obtain velocity data. It will again be integrated again using the same technique to obtain displacement data. Although, there are some error handling techniques that must be used in order to get any useful data. When the program is running, some fluxuations in the data may occur because of some error handling techniques. I will discuss this in the last section of this document. In order to remove the effects of gravity, the MATLAB script differentiates the acceleration data, then re-integrates it. What this does is remove any constants from your acceleration data, which is a good technique to compensate for g.



The wii uses an AXDL accelerometer, as shown on the right. This accelerometer is able to measure acceleration in all three directions by measuring a change in capacitance. Essentially, there is a cantilevered mass that rests above a thing film of silicon. When the mass isn't moving a baseline capacitance is set and used for steady state conditions. But when you move the accelerometer, the mass will move, deflecting a "plate" on the accelerometer, and this change in capacitance is noted to determine the acceleration value.

3) System Architecture

a. Design

The main components used in our systems design is an accelerometer with wireless capabilities (Wii controller), a portable laptop with Bluetooth capabilities and the program LabVIEW, and separate software that allows the computer to recognize and "talk to" the Wii controller. Instructions and examples on how to interface with a Wii controller can be found at: https://decibel.ni.com/content/docs/DOC-1353. Also, if you would like to do further data analysis, you will need to acquire MATLAB, although that part is optional.

The Wii controller is responsible for collecting acceleration data and then sending that data via bluetooth to the computer. The program LabVIEW is then responsible for taking that acceleration data and building an array that can be plotted on an acceleration vs time graph. The program LabVIEW also is responsible for numerically integrating the data and displaying the 3 dimensional plot of position. You can these use the program MATLAB to do further data analysis that may prove difficult to do in LabVIEW.

Rationale

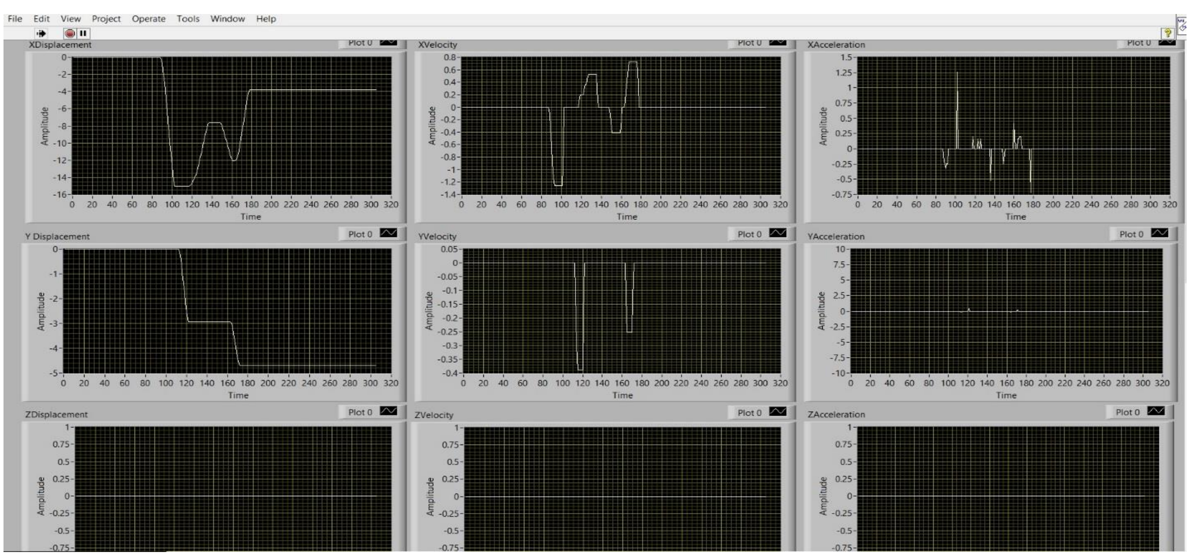
One of the main points of this system is to make it wireless and portable. That is why we chose our components to have Bluetooth connectivity and the ability to function on battery power. The whole point is so that you can take this system to different locations, attach the wii controller to different objects if you wish, and be able to obtain a three dimensional plot of the position. For example, you can strap the Wii controller onto your dog, and plot its position as he runs around the room.

4) GUI Design

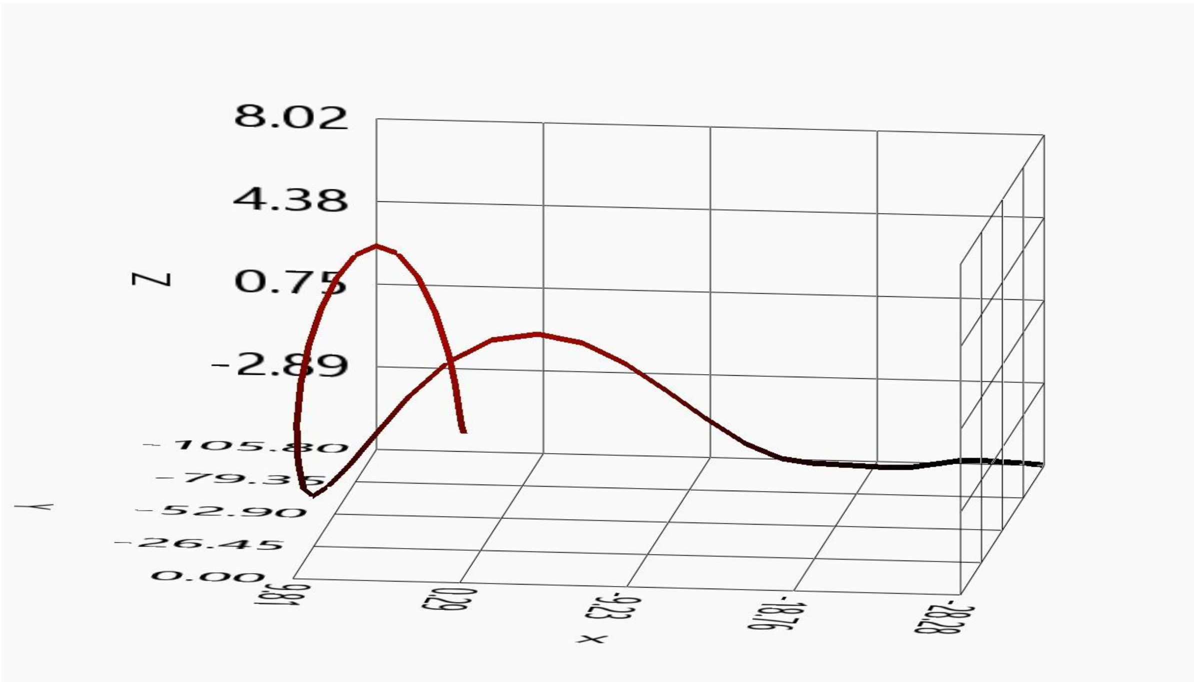
a. Overview

There will be a total of 9 graphs that are displayed. The first row from the top will be data in the x direction, the second row the y direction, and the third row z direction. The first column from the left will be displacement data, the second column will be velocity data, and the third column will be acceleration data. There will also be a 3-d plot underneath this section of 9 graphs that will track the objects position through space.

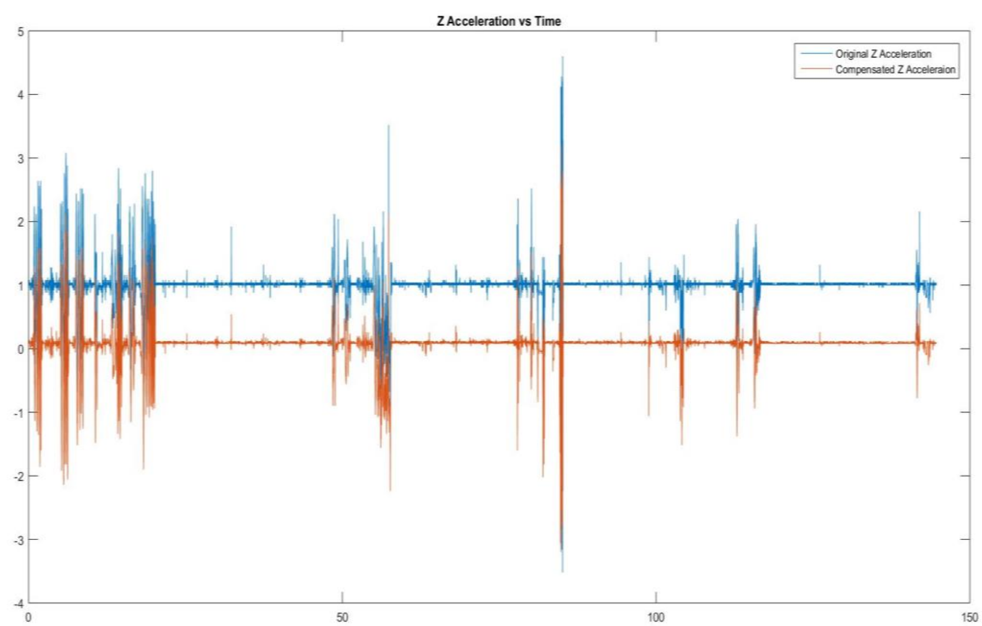
b. Screen shots and examples



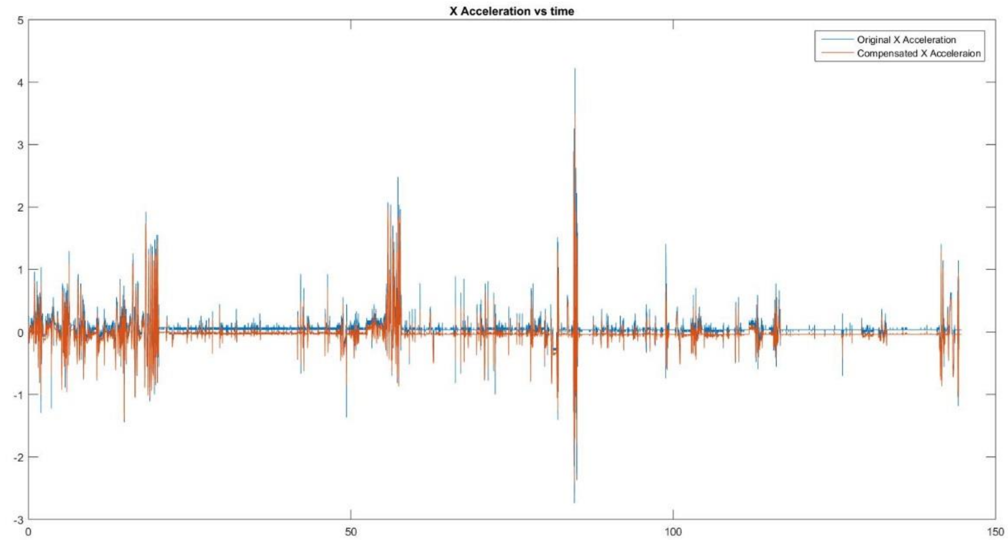
This part of the VI shown on the previous page individually displays the acceleration, velocity, and displacement in all three axis



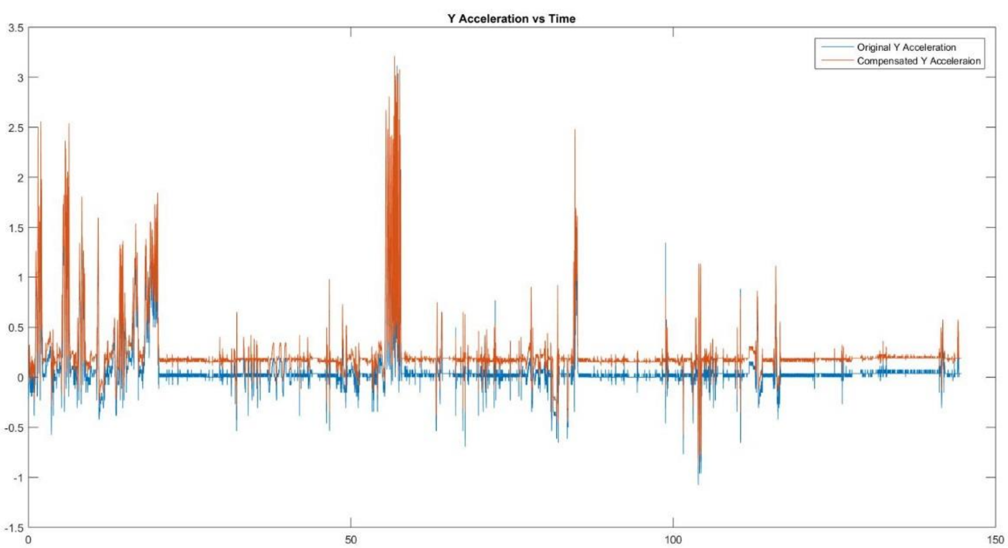
This part of the VI plots the position of the wii controller



This is a plot of the z acceleration vs time with the original acceleration data shown in blue, and the gravity compensated acceleration data shown in orange. This data analysis was done in MATLAB.



Plot of the x acceleration vs time with the original acceleration data shown in blue, and the gravity compensated acceleration data shown in orange. This data analysis was done in MATLAB.



Plot of the y acceleration vs time with the original acceleration data shown in blue, and the gravity compensated acceleration data shown in orange. This data analysis was done in MATLAB.

5) Future Improvements

An additional feature that would be nice to add is to make it so force measurements can be taken. This should not be too difficult because Force is simply the product of your mass and acceleration. Also, it would be interesting to plot the work done as well. The reason these features could be useful is because it gives you the possibility to strap the wii controller to an object, and measure how much force it takes to move the object, or measure the work it takes to move the object.

Another improvement that can be made to the system is to compensate for gravity in the VI. Right now you can only display compensated g values via the math script, but if this was done in the VI, perhaps it would generate more accurate plots.

6) Drawbacks / error handling techniques

One of the main drawbacks of this system is that it is not very accurate. This comes from the fact that the numerical integration techniques are only approximations, and not exact. In addition, the acceleration data cannot be gathered at a fast enough rate where it would give you accurate enough data to integrate twice over. Both of these factors causes the velocity to integrate to some finite value even when the wii controller is not moving, and then that causes the displacement to drift. Therefore, some error handling techniques and assumptions must be made.

This system assumes that whoever is moving this wii controller will not be moving this at a constant, finite velocity for an extended period of time. This means that every movement will be an acceleration, followed by a deceleration with hardly any time spend moving at a constant velocity. With this assumption, the system then is able to set the velocity back down to zero after it detects that there is no acceleration for a certain amount of time. This handles any data drift that might occur from having the velocity remain at a finite value for an extended period of time. After the velocity is set back down to zero, the system then needs to sets the area under the acceleration curve back down to zero. This is done by making the next point having a value of negative of whatever the integral was up to that point. This is why you may see fluxuations in your acceleration data.

Even with all of these error handling techniques, it still doesn’t gather very accurate displacement data. Perhaps making use of the infrared sensor may prove to be a better way to track position.