EE/CprE/SE 492 BI-WEEKLY REPORT 4

2/29 - 3/12

Group number: 32

Project title: Development of a Smart Sensing System for Road Performance Data Collection

Client &/Advisor: PROSPER - Bo Yang & Halil Ceylan

Team Members/Role: Victor Guerra, Ethan Young, Michael Petersen, Shlok Singh

o **Weekly Summary:**

In the last two weeks, we've been working on identifying and solving issues that we weren't anticipating. For one, the way we now plan on filtering some vertical acceleration data will be done on the Arduino itself. Another problem we're facing is timing the Arduino to send HTTP requests in reasonable time to continue polling the accelerometer. With the high rate we need to poll the accelerometer, we will first focus on achieving this goal while writing to the SD card. From the SD card we will calculate the IRI and still put it in the database. If we are able to get a fast enough transmission, we will work on the HTTP requests again.

o **Past week accomplishments**

- Arduino Device successfully polls Accelerometer data and writes data to the SD card.
- Physics derivation for raw accelerometer data into road profile completed
- Modularized the back-end javascript code
- Got the Arduino fully soldered together

o **Pending issues**

- Previous calculations suggested 120Hz polling rate, implementation of vertical acceleration filtering at 600Hz is still required.
- Exact filtering parameters for raw data need to be determined from literature and trial
- Embedded design must depend on constant forward speed and ignore turns

- **Individual contributions**

Name	Individual Contributions	Hours this week	Total Hours
Victor Guerra	IRI Calculation	7	108
Ethan Young	IRI Calculation, Arduino Testing	12	117
Michael Petersen	Arduino Testing	10	106
Shlok Singh	Server Preparation/Development	4	102

- Victor
    - I completed the derivation for accelerometer data to profile. The simple math backing for this is a double integral to get from acceleration to position, but there are additional considerations with using the accelerometer. For one, the accelerometer reports value in g-forces, so unit management was important. Then, since this is a discrete series of points, it made more sense to me to use riemann sums than a more tight integration function.
    - Then, the interval for the integrals needed to be considered. The range is supposed to be over the course of about 300mm, but since we'll be integrating with respect to time and not distance, I needed a way to convert distance into time, which is dependent on velocity. The accuracy of the integrals relies on some pretty big assumptions about the continuity of the road, so it should be interesting to see the accuracy of the programmatic solution versus a hand-solved result.
- Ethan Young
    - I helped Victor out with the completion of the accelerometer data to profile. More of the specifics are in his contribution notes. There still is an issue though with how we will format the stored data on the SD card. Once I know how we plan to do that, getting the rest of the calculations will be easier. Along with this, I translated most of it to javascript and plan to test it through unit test cases over break.
    - I also soldered the rest of the Arduino together and tested it by testing each component alone, and then all together. This process whas a lot faster than the first time we tested the components as we had all of the test cases handy.
    - I also extracted all of the backend javascript into separate modules for easier readability and unit testing.
- Michael Petersen

    This week, after group discussion, we determined the accelerometer data needed to have a sampling rate of around 600Hz for IRI calculations. The hardware we have been using supports upto 400Hz. This means that our IRI calculation will lose accuracy after a certain speed. Previously, we were polling the accelerometer using system interrupts, I reformatted the code to continuously poll the accelerometer at our desired sampling rate (changes with vehicle speed). The interrupt service routine will instead be used to retrieve GPS data at a different interval. After changing the operations of our code, I tested acceleration data retrieval which works at the new sampling rate.
- Shlok Sing
    - Played with GSM signals. Investigating HTTP POST and TCP. Considering the use of websockets to save time.

- **<u>Plans for the upcoming week</u>**
  - Slight rewrite of Arduino code to implement the higher polling rate previously described + further implementation of transmission of data to web server.
  - Continue implementation of accelerometer to profile physics and testing it with unit test cases
  - Implementation of some filtering technique for acceleration and velocity data

- **<u>Summary of weekly advisor meeting</u>**
  - Did not meet with advisor previous week