Design of an Integrated System for Measuring Road Surface Roughness

Talha Alvi, Natnael Alemu and Iven H. Liu
Advisor: Dr. Raman Paranjape

Abstract*— *This paper presents the design process for the work carried out in the development of a system built for measuring road roughness and road surface conditions. The system employed the use of accelerometer sensors and a camera in order to get data on the road surface quality. Physical prototypes were built and tested to meet the required design specifications. The system was then mounted on a car and bump testing was performed in order to prove out the system. Afterwards the system was used to collect real world data, which was then displayed on a website.

I. INTRODUCTION

Saskatchewan Ministry of Highways and Infrastructure currently spends close to half a million dollars annually to gather surface roughness and overall condition data of over 20,000 km of highways. The City of Regina also uses a similar system to gather data from major city roads every three years [1]. A new approach minimizes the overall cost of such systems by building a low cost and modular system that can be used to take measurements of the road roughness and can be used to supplement the more expensive equipment to map all of the roadways. This system will also be able to support the measurement of road surface roughness where data is not currently being collected such as smaller rural municipalities roads and residential city roads. The measurement of road roughness along with images that provide visual reference are used to quantify road surface conditions.

The system uses three roughness sensors (accelerometers) with two mounted over the back wheels in the car and a third inside the car that transmit real time roughness data wirelessly to a central terminal situated in the car. The car position is tracked with a GPS sensor that organizes the data transmitted to the central terminal with location coordinates of the car. At the same time a camera, attached to the central terminal, captures images of the road surface and the surroundings as the car is moving. The captured roughness and image data are stored in the central terminal. Afterwards, automated scripts process the data using digital signal processing tools to analyze the accelerometer and image data to extract usable information from the data. The required information is processed on the central terminal, which uploads the data to a server. A website is used to display the information regarding road roughness and conditions in an intuitive and accessible format.

II. SYSTEM DETAILS

A. Background

Based on literature review, and talks with Ministry of Infrastructure and Highways, the International Roughness Index (IRI) is the most widely accepted method for gauging road surface roughness. There are accepted approaches for deriving IRI, which is measured in m/m or m/km of vertical displacement over a section of the road. For accelerometer based systems, the two approaches generally suggested either use a numerical integration approach to find the vertical displacement or use the Power Spectral Density from the accelerometer data to derive the measurement [2][3]. For simplicity, the integration approach was used to derive the IRI as a part of this project.

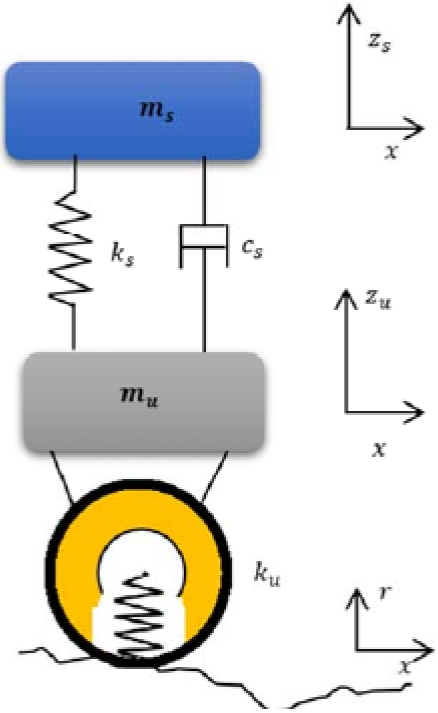

Fig. 1. Quarter Car Model showing sprung and unsprung masses in a car [2]

From the quarter car model above, the differential equations modeling the system can be derived as such.

$$m_s\ddot{z}_s + c_s(\dot{z}_s - \dot{z}_u) + k_s(z_s - z_u) = 0 \quad (1)$$

$$m_u\ddot{z}_u + c_s(\dot{z}_u - \dot{z}_s) + k_s(z_u - z_s) + k_u z_u = k_u r \quad (2)$$

The system has accelerometers placed on both the sprung and unsprung masses of the car. The value of the vertical displacement required for IRI can then be computed using numerical integration techniques on the accelerometer data being collected. Simpson's method shown below is used to integrate the acceleration data.

$$\dot{z}_{i+1} = \dot{z}_i + \frac{(\ddot{z}_i + 4\ddot{z}_{i+1} + \ddot{z}_{i+2})}{6} \quad (3)$$

With the velocities of the sprung and unsprang masses calculated, the following formula is used to computed the roughness index.

$$Roughness = \frac{1}{N}\sum_{0}^{N} |\dot{z}_s - \dot{z}_u| \quad (4)$$

B. Overall System

The system block diagram can be seen below. The system set up includes three wireless nodes with two mounted on the control arms of the car, representing the unsprung acceleration. One node is mounted inside the car, representing the sprung acceleration. The central processing unit is situated towards the front of the car and is used to record the acceleration data along with GPS and images of the road, while driving.

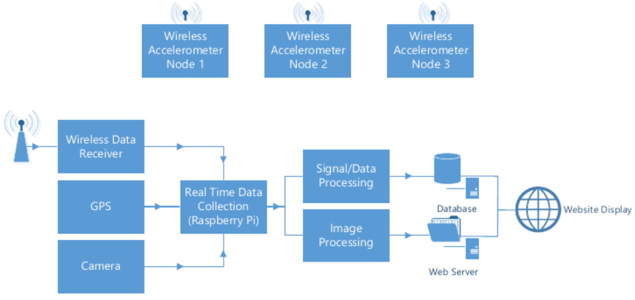
Wireless Accelerometer Node 1
Wireless Accelerometer Node 2
Wireless Accelerometer Node 3
Wireless Data Receiver
GPS
Camera
Real Time Data Collection (Raspberry Pi)
Signal/Data Processing
Image Processing
Database
Web Server
Website Display

Fig. 2. Overall System Block Diagram

C. Design Specifications

The following design specifications were selected for the system.

- Designed to be able to detect and characterize roughness of at least 5 cm vertical displacement over a 30 cm length of the road
- Samples at least every metre long section of the road
- Noise and other variables such as car speed, or car suspension dynamics are either minimized or controlled in order to get a true indication of road surface roughness
- Wireless data transmission rate is able to support the maximum data rate selected for the device (minimum 60 bytes/sec required data rate)
- GPS system takes a minimum of one sample per second
- Positioning system maintains a similar accuracy ($\pm 20\%$) as a cellphone GPS
- Images are captured fast enough to capture the whole road surface: image taken every 10 metres
- Able to provide power to the components for a minimum of one hour

III. Design Process

The design process consisted of two parts which included the hardware design portion to use the appropriate sensors and devices to perform the functions specified and software running on those devices able to fulfill the tasks as required.

A. Hardware Design

The following devices were used to build the wireless accelerometer nodes:

- STM32F100RB Microcontroller
- MPU6050 Accelerometer
- XBee S2C
- 9V Rechargeable Battery
- LP2950 Voltage Regulator

The central processing unit was comprised of the following devices:

- Raspberry Pi
- Raspberry Pi Camera
- Adafruit GPS
- XBee S2C

The build process included testing the operation of the above components on breadboards to ensure that they worked and would perform the required functions. Afterwards, Eagle was used to design the printed circuit boards (PCBs) to layout the components

B. Software Design

The software design portion of the project was much more indepth and involved being able to integrate all the necessary components to perform their required functionality, while meeting the design requirements that were specified.

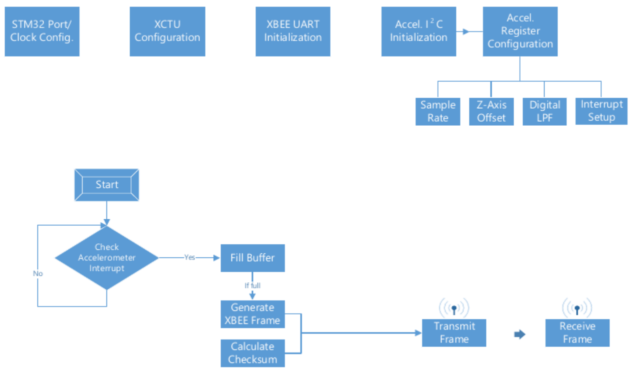

Fig. 3. Wireless node code flow diagram

The above flow diagram shows the general workings of the wireless nodes. The wireless nodes are configured in a star network with the receiving node on the central processing unit acting as the central node. The configuration is performed using XCTU tool for all four Xbees used in the project. With the wireless network configured, the STM32 MCU is configured to communicate with the MPU6050 accelerometer using I2C protocol. The appropriate registers are configured as shown. The program essentially runs on an interrupt and fills a buffer containing 50 samples obtained from the accelerometer. The samples are used to build an API frame 100 bytes long, which is then transmitted to the receiving node. The sampling rate for the accelerometer is set to 200 samples/sec, which means each wireless node sends 4 frames/sec as each frame contains 50 samples. Every second the receving node is getting 12 frames containing accelerometer information.

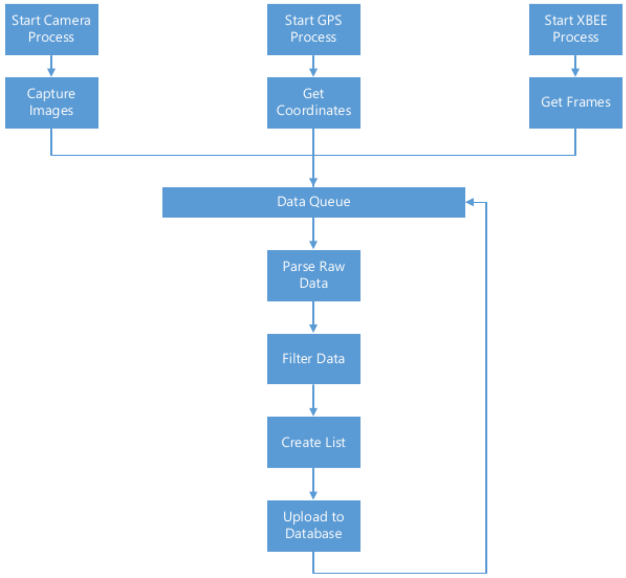

Fig. 4. Central data collection code flow diagram

The flow diagram above shows the general process used to get the data, process it and save it. Multiple processes are used in a python process to read serial data from GPS and Xbee continuously, while the camera is taking pictures. The GPS coordinate data, the accelerometer data from Xbee and the name of the picture files are put into a FIFO queue, which is then used to process the data. The processed data is then uploaded to the database to be displayed on the website. The website uses Mapbox API, with GeoJSON used to process the GPS coordinates, which are then displayed on the map with the roughness index. The images taken are linked as well and displayed side by side with the roughness data.

C. Signal Processing

The following block diagram describes the signal processing done on the accelerometer data to get the roughness index. It follows the math described in the background section of the report. The raw accelerometer data is initially filtered with a DC block filtering modeled by the difference equation $y[n] = x[n] \;\; x[n-1] + 0.9995y[n-1]$. This provides a sharp cut off that only removes the 0 Hz component, which is necessary to make the mean of the data zero since all accelerometers are mounted in different orientations. Afterwards a bandpass Butterworth IIR filter is used with a band pass of 3-30 Hz. This filter helps to filter out high frequency noise and low frequency vibrations from the car. Acceleration is then integrated to velocity, which is then used to compute the roughness index for the left and right wheel paths by summation of the difference between sprung and unsprung velocities [2].

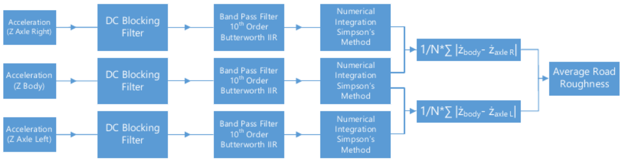
Acceleration (Z Axle Right)
Acceleration (Z Body)
Acceleration (Z Axle Left)
DC Blocking Filter
DC Blocking Filter
DC Blocking Filter
Band Pass Filter 10th Order Butterworth IIR
Band Pass Filter 10th Order Butterworth IIR
Band Pass Filter 10th Order Butterworth IIR
Numerical Integration Simpson's Method
Numerical Integration Simpson's Method
Numerical Integration Simpson's Method
$1/N^*\sum |\dot{z}_{body} - \dot{z}_{axle\ R}|$
$1/N^*\sum |\dot{z}_{body} - \dot{z}_{axle\ L}|$
Average Road Roughness

Fig. 5. Process used to get roughness data from raw accelerometer data

D. Image Processing

The block diagram above shows the process used to analyze the images of the road surface to highlight the potholes and cracks in the image. First a bounded region of interest is created containing only the road surface as a mask. The masked image is then converted to gray scale and a smoothing Gaussian blur filter is used to filter out high frequencies. An edge detection algorithm is used to find the edges in the image. The edges are then dilated. The contours of the dilated edges are identified and a convex hull algorithm is used to find the points outlining the pothole or the crack. The identified crack or pothole area is then overlaid onto the original raw image. Actual images from the process are shown below [4].

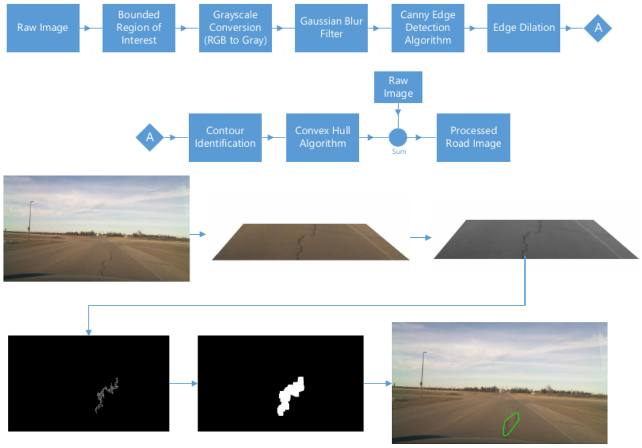

Fig. 6. Details of process to extract cracks and potholes from images using edge detection

IV. Testing and Results

A. Testing

Testing was an important step of the design process in order to ensure that the system met the specified design specifications. The following were the major tests under taken as part of the project. Further detail on testing steps and results can be found in the attached test document.

- Roughness testing using simulated bumps of 2.5 cm and 5 cm over a length of a 15 cm
- Testing using various filters to identify best method to minimize noise

- Sample rate of accelerometer tested up to maximum sampling rate (1000 samples/sec)
- Wireless data transmission/reception tested to determine maximum sampling rate possible (200 samples/sec selected)
- GPS sampled once per second in order to determine car position
- GPS accuracy tested by plotting data against GPS data from cellphone on a map
- Image capture rate tested to determine maximum image sampling rate
- Images visually analyzed for identification of cracks/potholes on road surface
- System tested using battery power for 1 hour to ensure battery meets specifications

B. Results

After proving that the system could work as specified through the testing described above, data was collected to be displayed on the website. As can be seen below, the information on the website combines all the data that was collected in a usable format that is easily accessible. A map showing where the data was collected shows how rough the streets are through a 3D visual overlayed on to the map. By clicking on the different areas where data has been collected, the images from that part of the road can also be seen. Lastly, the actual roughness that is measured for that part of the road is displayed as part of the graph at the bottom to give a quantitative measure. Data can be accessed at *roadquality.ca*.

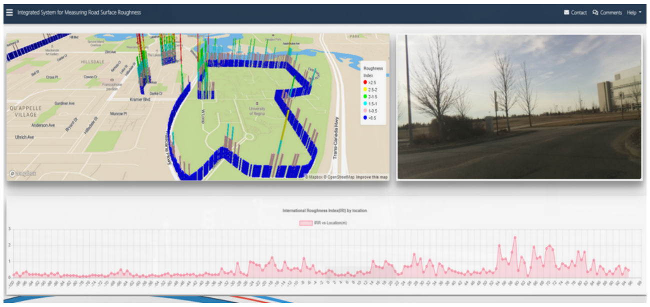

Fig. 7. Final display format of collected data

C. Design Challenges

There were several design challenges that we faced as part of our system build from both a hardware and software perspective. One of the biggest challenges initially was just being able get the microcontrollers to communicate with the Xbee devices. The Xbee devices when powered on would go into a boot loader mode that required special initialization to get back into normal mode.

Another challenge had to do with the physical foot print of the wireless nodes. The boards were designed to be 100 mm x 100 mm, however, we did not keep in mind that it would be good to minimize the size to make it easier to find enclosures and mount them on the car. This made it a challenge to find low cost enclosures and also mount those enclosures.

GPS accuracy was another problem when collecting data continuously. Under some conditions such as driving under bridges or near high rise buildings, GPS signal was lost, which meant that the signal data had to be interpolated when not captured. This limited the accuracy of the system.

Throughout the whole project one challenge that we had to work around was weather. Since our project was based around collecting outdoor information, it was a challenge with the long winter to perform the outdoor testing for our devices. It made it difficult to test devices like GPS as signals could not be obtained indoors. Image processing could not be tested either since roads were covered with snow for most of the time during the build process. This made it difficult to fix the problems discovered with the image processing in time as the project due date was near. Overall, working with the given weather was one of the biggest challenges through the project.

V. CONCLUSIONS

The project was able to achieve the goals outlined initially and prove as a suitable platform for further development. Since the overall goal of the project was to prove that this set up and design was viable, which was accomplished. The next step for this project would be to refine the design. The first step would be to layout a PCB with a custom microcontroller design that has wireless capabilities built in. Using a surface mount accelerometer design instead of a through hole would also be a part of it. This would help save both space on the PCB and make the design more power efficient, helping achieve better battery life for the wireless nodes. For our system, Python was used to build the main data collection framework, however, a more purpose built language for handling concurrency might be another design change. One other big change would be to explore options such as dual cameras or camera augmented with laser to scan the road surface. Further curve fitting and correction factors need to be applied to the data to match it to the accepted IRI standard, as well. With these changes this can help bring the system to a level that it could be worth exploring commercial application of such a system.

ACKNOWLEDGMENT

Thanks to Dr. Raman Paranjape and Doug Wagner for their helpful advice through the length of the project.

REFERENCES

[1] H. Soleymani, Interviewee, *Talk regarding capstone design project requirements for a project on road roughness measurement.* [Interview]. 27 September 2018.

[2] S. Eskhabilov and A. Yunusov, ”Measuring and Assessing Road Profile by Employing Accelerometers and IRI Assessment Tools,” *American Journal of Traffic and Transportation Engineering,* vol. 3(2), pp. 24-40, 2018.

[3] P. Andren, ”Power spectral density approximations of longitudinal road profiles,” *International Journal of Vehicle Design,* 2006.

[4] S. Ninenaber, M. Booysen and R. Kroon, ”Detecting Potholes using Simple Image Processing Techniques and Real World Footage,” *34th Annual Southern African Transport Conference SATC*, 2015.