IntelBri.github.io

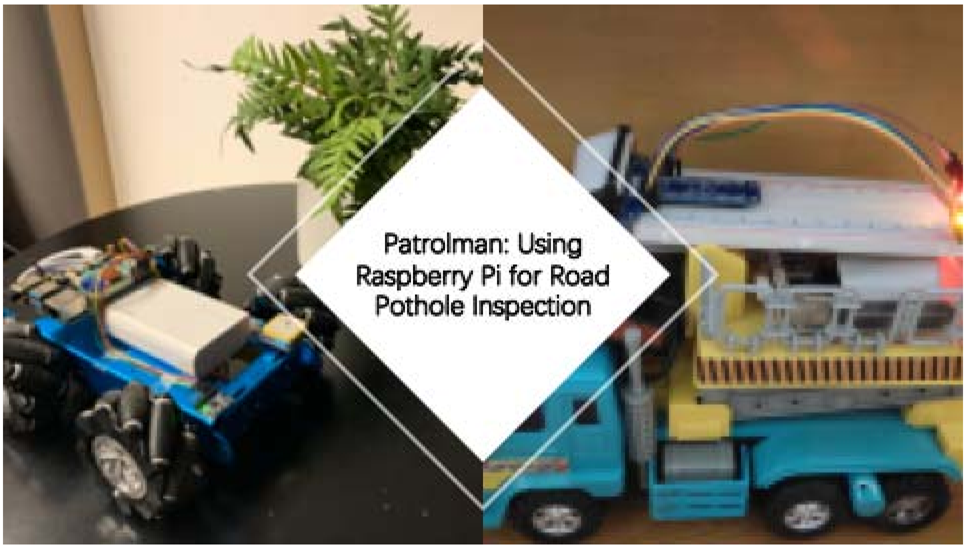
Patrolman: Using Raspberry Pi for Road Pothole Inspection

View on GitHub

Patrolman: Using Raspberry Pi for Road Pothole Inspection

• Pengkun Liu; Ruoxin Xiong

Video link: http://www.youtube.com/watch?v=wE4p8kOEM_Y



1 Introduction

Road potholes are common defects for aging civil infrastructures around the world. For example, in 2017, more than half a million potholes have been reported to local departments in the UK [1]. The massive amounts of road potholes require increased maintenance capital from the management authorities. The Edmonton Administration used $4.8 million to repair 450,000 potholes in 2015 [2]. The bad road conditions also lead to discomfort and even safety hazards to vehicle drivers or commuters, especially under high speed or low visibility conditions [3]. Therefore, this project designs and implements a portable Raspberry Pi system named Patrolman for road pothole inspection.

Image source:Pothole - Wikipedia

1.1 Motivation

Reliable routine monitoring and assessments of road conditions can lead to timely preventive actions. Traditonally, inspectors will visually assess the road conditions and report the road potholes [3]. Due to the significant scale of the roadway networks, this manual method is typically expensive and error-prone. Considering that road conditions can be transmitted and measured in vibrations during a drive [4], the team designs a portable Raspberry Pi system - Patrolman, for road pothole inspection. The Patrolman system makes use of the mobility of vehicles (we use a toy car) to collect vibration signals and identify abnormal road conditions.

1.2 Goals

This project presents a mobile system – Patrolman, using Raspberry Pi to detect and report road potholes with their georeferenced locations. The patrolman system uses the vehicles' mobility, continually collecting the accelerometer and GPS sensors and evaluating road surface conditions the with the sensing signals.

2 For Progress Reports

Progress Report_Oct 5th

Progress Report_Oct 8th

Progress Report_Oct 15th

Progress Report_Oct 17th

3 Methodology

3.1 Phenomena of Interest

The vibration patterns sensed by driving cars actually reflect the road conditions being monitored [4]. The general sensing signals intended to assess road conditions include:

1. Vibration - Acceleration is a common measurement in characterizing vibrations. When the participating vehicles move on smooth roads, the acceleration values (g) are typically steady, while acceleration may show fluctuating patterns when driving on bad road conditions with potholes.

2. Georeferenced locations - We also use the GPS module to record the trajectories of moving vehicles. Therefore, we can report the georeferenced locations for each identified road pothole.

3.2 Sensor(s) Used

In this project, the team leverages and collects signals from sensors mounted on a testing vehicle (we use a toy car instead). The Patrolman system consists of a Raspberry Pi, an MPU-6050 six-axis accelerometers, a GPS, and a power bank.

Accelerometer: MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS

MPU6050 sensor is a 6-axis motion tracking module, which has both a 3-axis Accelerometer and Gyroscope. The features and specifications of this sensor (descriptions refer to MPU6050 Sensor Module Wiki are shown below:

- Features and Specifications ```markdown
  - Power voltage: 2.375 – 3.46V
  - 3-axis Gyroscope values in degrees per second (dps)
  - 3-axis Accelerometer values in gravity force (g)
  - 3-axis Gyroscope full-scale range: ±250, ±500, ±1000, and ±2000dps
  - 3-axis Accelerometer full-scale range: ±2, ±4, ±8, and ±16g ```

The detailed reference materials are from MPU6050 Sensor Module Wiki.

GPS sensor: L76X GPS Module

L76X GPS Module, as a common Global Navigation Satellite System (GNSS) module, supports Multi-GNSS systems, including Global Positioning System (GPS), Quasi-Zenith Satellite System (QZSS), and BeiDou Navigation Satellite System (BDS) [5]. The features and specifications of this sensor (descriptions refer to L76X GPS Module User Manual) are shown below:

- General Features and Specifications ```markdown
  - Update rate: 1Hz (default), 10Hz (max)
  - Power voltage: 3.3V / 5V
  - Horizontal position accuracy: <2.5m circular error probable(CEP)
  - Dynamic performance:

    - Velocity (max): 515m/s
    - Acceleration (max): 4g
  - 2x working status LEDs
  - Dimensions: 32.5mm x 25.5mm ```

The detailed reference materials are from L76X GPS Module User Manual.

3.3 Signal Conditioning and Processing

3.3.1 Sampling frequency

The current operating system of Raspberry Pi uses Linux kernels, which do not support real-time clock. So the Raspberry Pi typically cannot generate real-time pulses to control high-frequency sensors in the experiments. To determine the limitation of the Raspberry Pi system, the team uses different sampling frequencies and conducts a host of pre-experiments.

- Accelerometer: The accelerator sensor is installed on the moving cars to test its vibration on the Z-axis. The team counts the actual sampling duration for receiving 300 signals and compares it with the theoretical values.

In the experiments, the sampling frequencies of the Accelerometer are tested for 1Hz, 2Hz, 5Hz, 10Hz, 20Hz, 50Hz, 100Hz and 200Hz.

Sampling frequency	1Hz	2Hz	5Hz	10Hz	20Hz	50Hz	100Hz	200Hz
Theoretical Signal received	300	300	300	300	300	300	300	300
Theoretical sampling time (s)	300	150	60	30	15	6	3	1.5
Actual sampling time (s)	301	151	62	32	17	8	5	3
Error rate (%)	0.33	0.67	3.33	6.67	13.33	33.33	66.67	100

In particular, when the sampling frequency of the accelerometer reaches 200Hz, some data points are collected with zero values with an error rate of 100%. Here, the team uses 10Hz as the sampling frequency in the experiments.

Index	3	4	5	6	7	8
118		0.0151367	-0.0786133	0.97998	-4.58586	-0.882084
119		0.0180664	-0.0908203	0.977539	-5.30705	-1.05425
120		0.00341797	-0.0771484	0.968506	-4.55437	-0.201564
121		0.0310059	-0.0795898	0.977295	-4.6535	-1.81118
122		0.013916	-0.0737305	0.97998	-4.30221	-0.81127
123		0	-0.0751953	0.982422	-4.37693	-0
124		0.0283203	-0.0927734	0.980957	-5.40041	-1.64633
125		0.0168457	-0.0793457	0.97583	-4.64786	-0.985743
126		0.015625	-0.079834	0.980225	-4.65557	-0.910217
127		0	-0.0900879	0.969971	-5.30623	-0
128		0.015625	-0.0800781	0.986084	-4.64212	-0.904827
129		0.0134277	-0.0793457	0.97876	-4.63426	-0.783429
130		0.0175781	-0.0744629	0.986328	-4.31667	-1.01811

- GPS sensor: The GPS module's max frequency is 10Hz, and the default frequency is 1Hz. Here, the team uses the default frequency - 1Hz as the sampling frequency due to the low speed of the mobile toy car.

3.3.2 Absent values of sensors

The sensors, including the GPS module and accelerometer, often miss some signals or receive zero readings during the experiments. For example, when passing through bridges, closed buildings, or tunnels, the GPS module may lose its connections.

In our experiments, we use linear interpolation to the missing values between GPS readings due to the testing cars' low speed. Similarly, for the accelerometer's zero readings, the team also fills the missing values with linear interpolation.

3.3.3 Signal smoothing

To uncover and identify the signal patterns of acceleration data, the team applies the moving average method to smooth the accelerometer signals and remove noisy values to reflect the overall trends.

3.4 Pavement defect patterns

The team focuses on three main road defects [6]:

- Smooth road (SM): Good road conditions with smooth road surface.
- Potholes (PH): Missing chunks of the road surface.
- Upheaval (UH): Localized upward swelling on the roads.
- Cracking (CR): Road cracks often result from frequent vehicle movements and temperature changes.

Cracking

Potholes

Upheaval

Image source: https://www.pavemanpro.com/article/identifying_asphalt_pavement_defects/

Pot Holes

Cracking

Upheaval

Outdoor testing environments

4 Experiments and Results

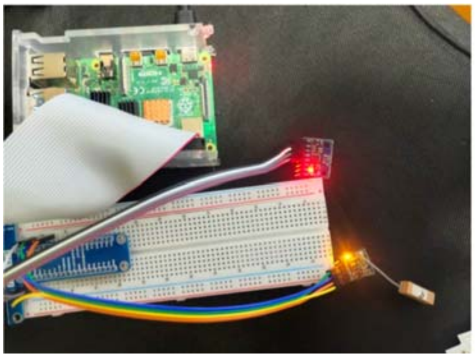
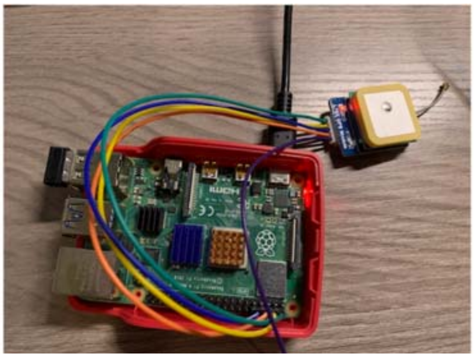
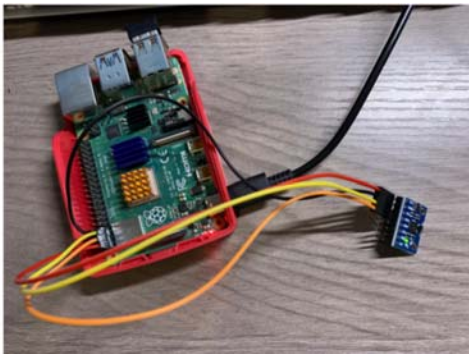
4.1 Data collection

- Getting hardware

We have purchased Raspberry Pi and all required sensors (GPS module and MPU-6050 accelerometer) from Taobao.

- Set up the test environment

We have built all sensors and written Python codes to monitor the parameters based on the MPU6050 Sensor Module Wiki and L76X GPS Module User Manual. The sampling frequencies of the MPU-6050 accelerometer and GPS module are 10 Hz and 1 Hz, respectively.

- GPS module test results
    - Hardware connection

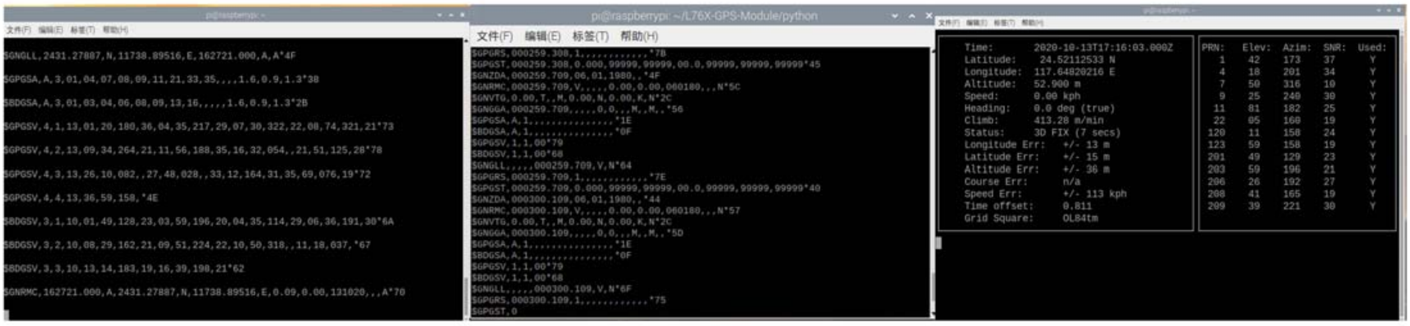
Connect L76X GPS module to the Raspberry Pi board. Four pins are used: VCC, GND, RX, and TX [5].

PIN	Raspberry Pi Board	Raspberry Pi BCM
VCC	5v	5v
GND	GND	GND
RX	8	P14
TX	10	P15

The detailed interface connections refer to L76X GPS Module User Manual.

- Running code

Run the test code GPS.py. The test results are shown as follows:



- MPU 6050 sensor test results
    - Hardware connection

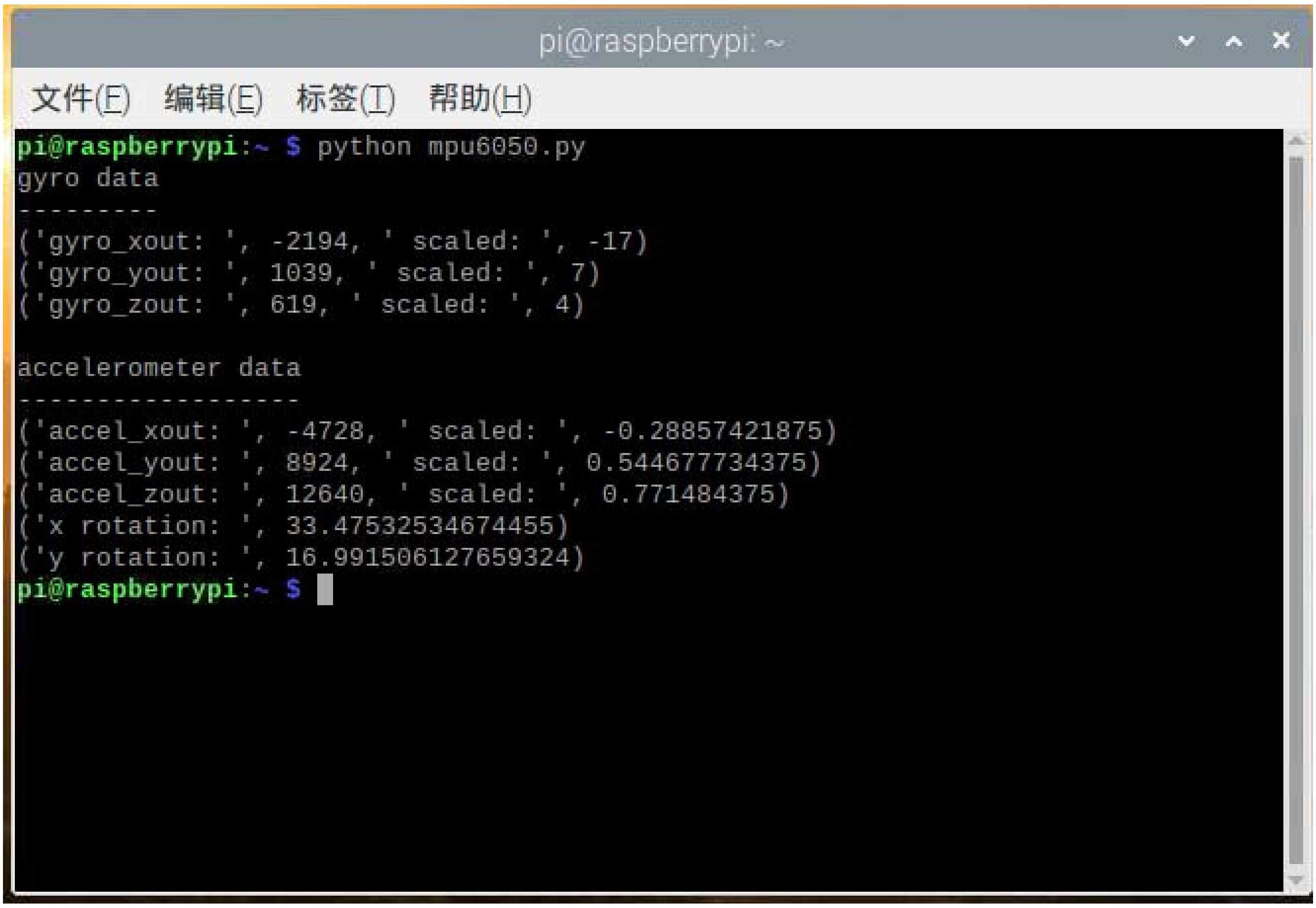
Connect the MPU 6050 sensor to the board. Four pins are available for use: VCC, GND, SDA, and SCL [8].

PIN	Raspberry Pi BCM
VCC	3.3v
GND	GND
SDA	P3
SCL	P5

The detailed interface connections refer to MPU6050 Sensor Module Wiki.

- Running code

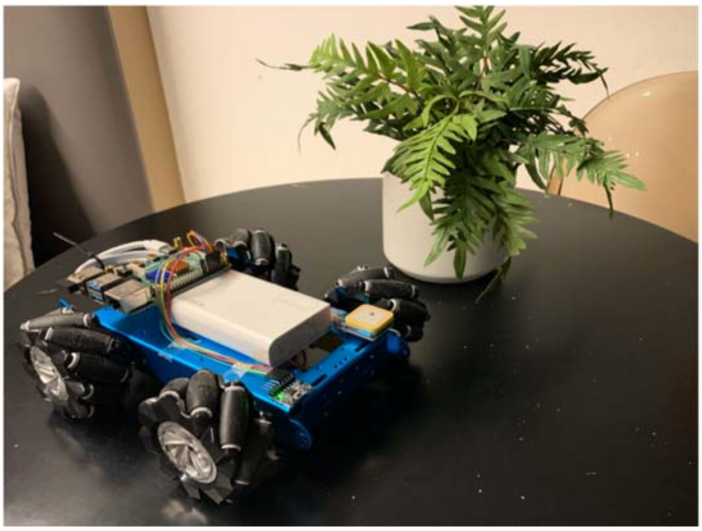
Run the test code Acceleration.py. The test results are shown as follows:

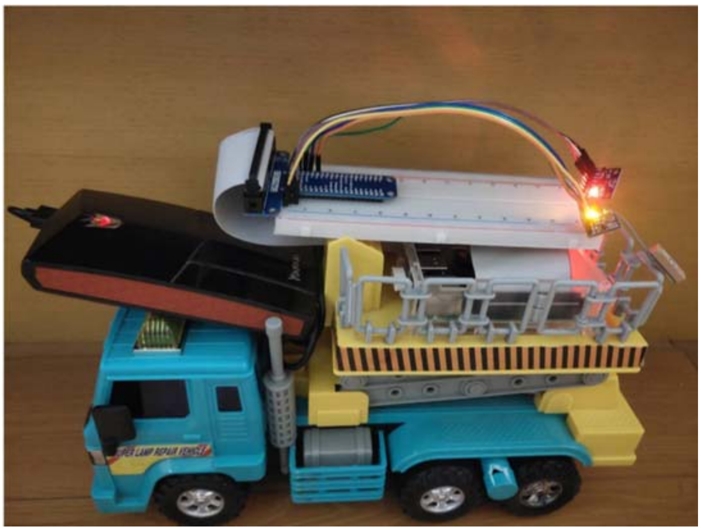


4.2 Indoor experiments

4.2.1 Patrolman system desgin

We design the Patrolman system to detect and report road potholes with their georeferenced locations. The system consists of a mobile platform (we use the toy car in the experiments), an MPU-6050 accelerometer, and a GPS sensor. Also, we use the power bank as the power supply to the whole system.

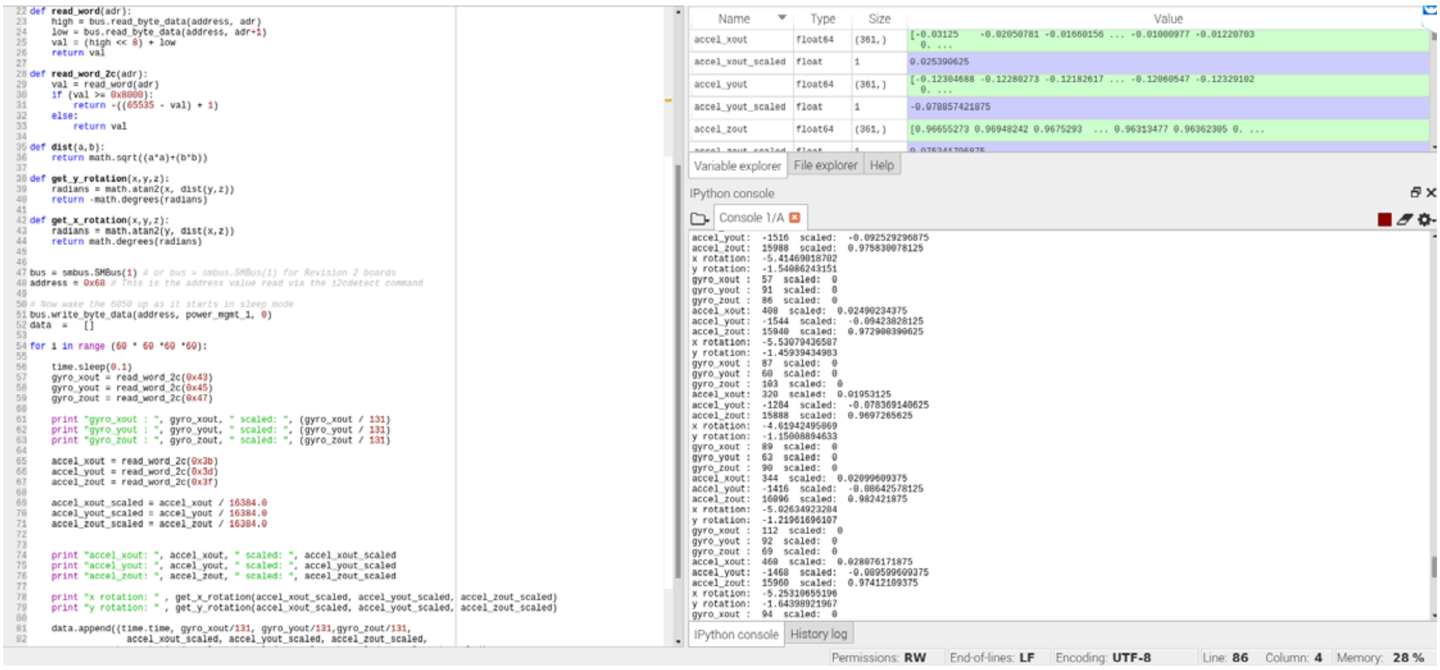


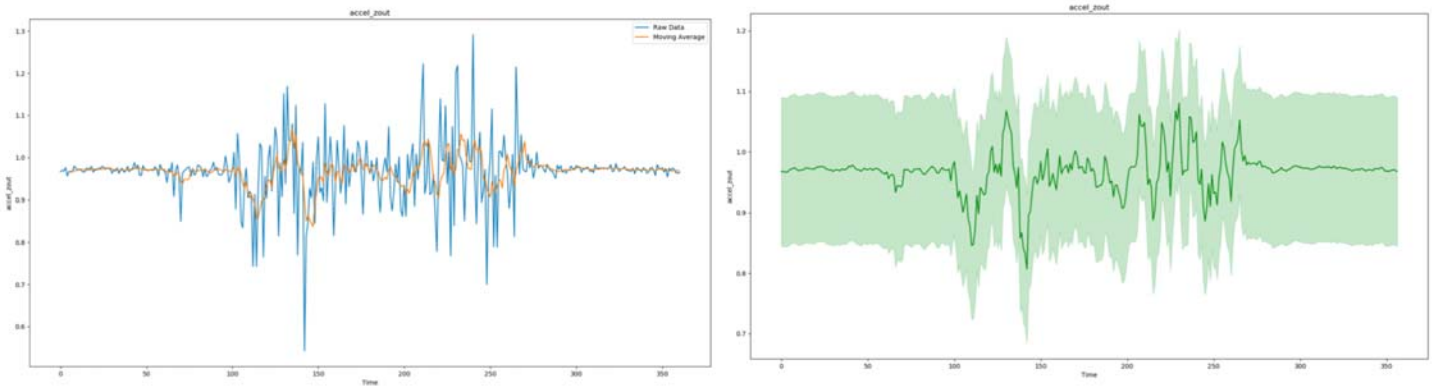


4.2.2 Indoor testing results

For the indoor experiments, the adopted sensors communicate the captured accelerometer signals to the computer through Wi-Fi.

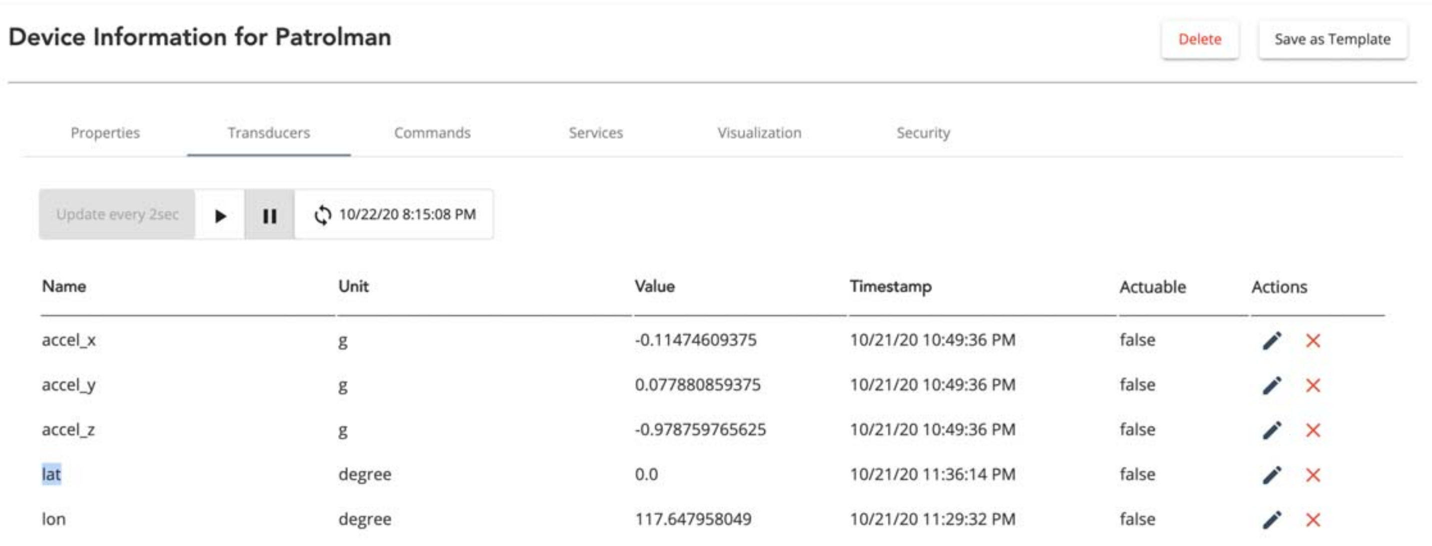
The Z-axis acceleration shows significant patterns when the car crosses over the road obstacle contrasted to the smooth road surface.





4.3 Set up IoT device

Following the tuturial, we create a device named “Patrolman” on OpenChirp.

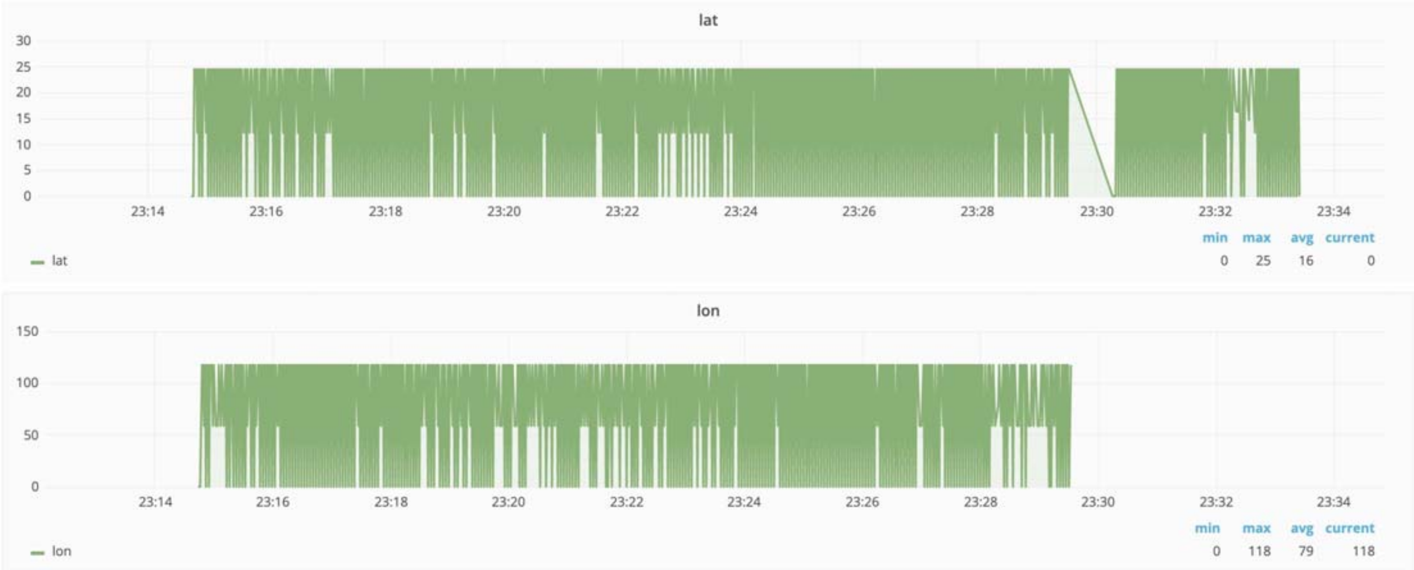


Transducers

Name	Unit	Actuable
accel_x	g	false
accel_y	g	false
accel_z	g	false
lat	degree	false
lon	degree	false

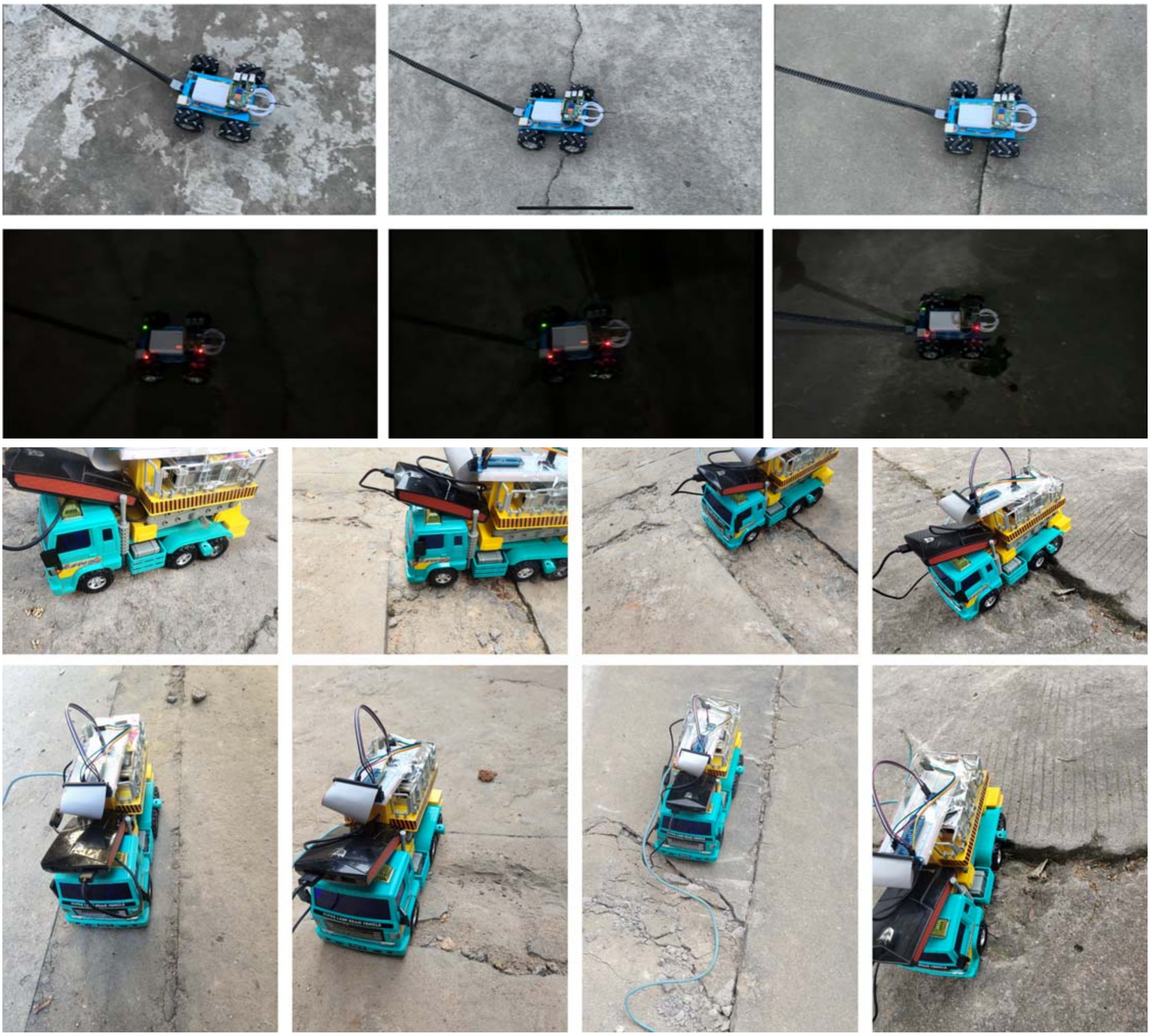
We then release sensor values onto OpenChirp, which can timely report and visualize data on the website.

See acc_openchirp.py and gps_openchirp.py for the code.



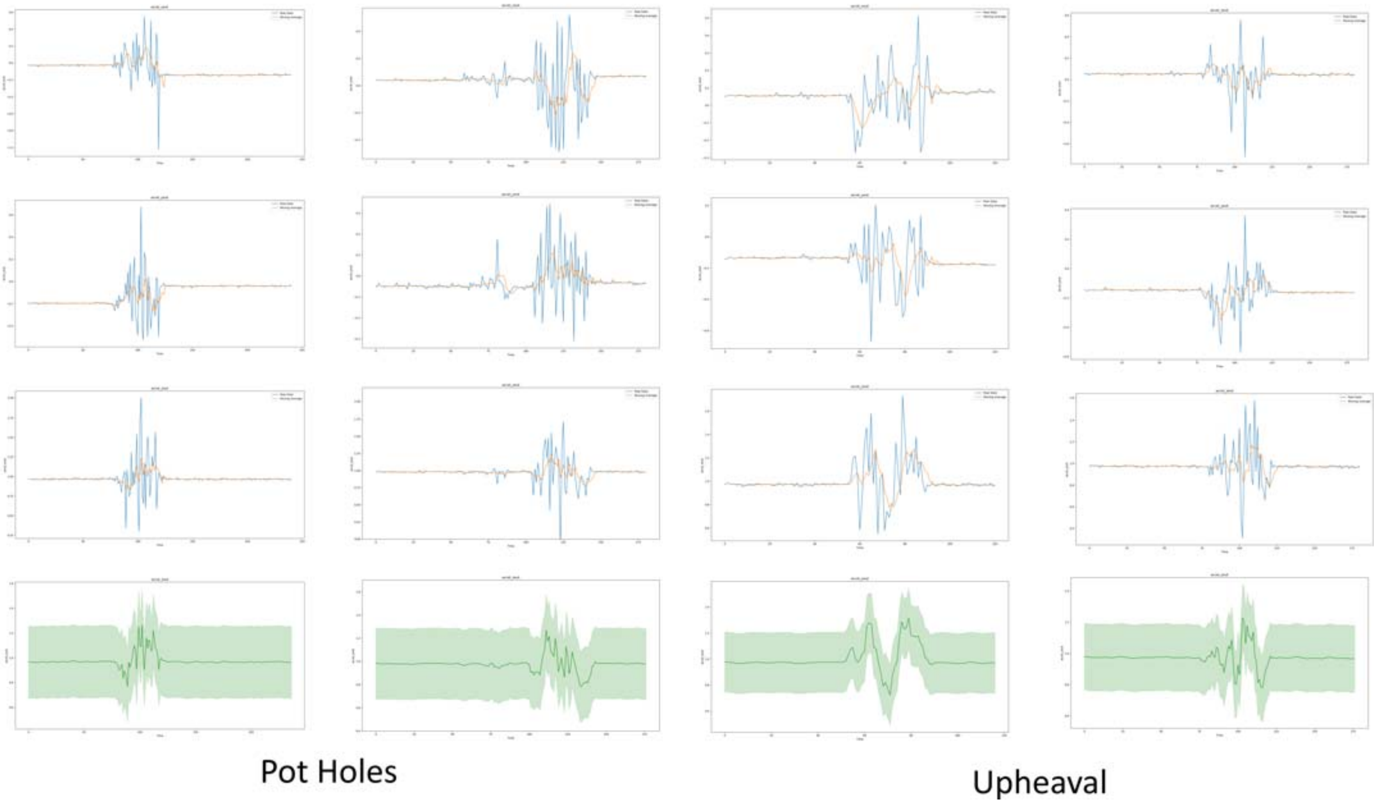
4.4 Outdoor experiments

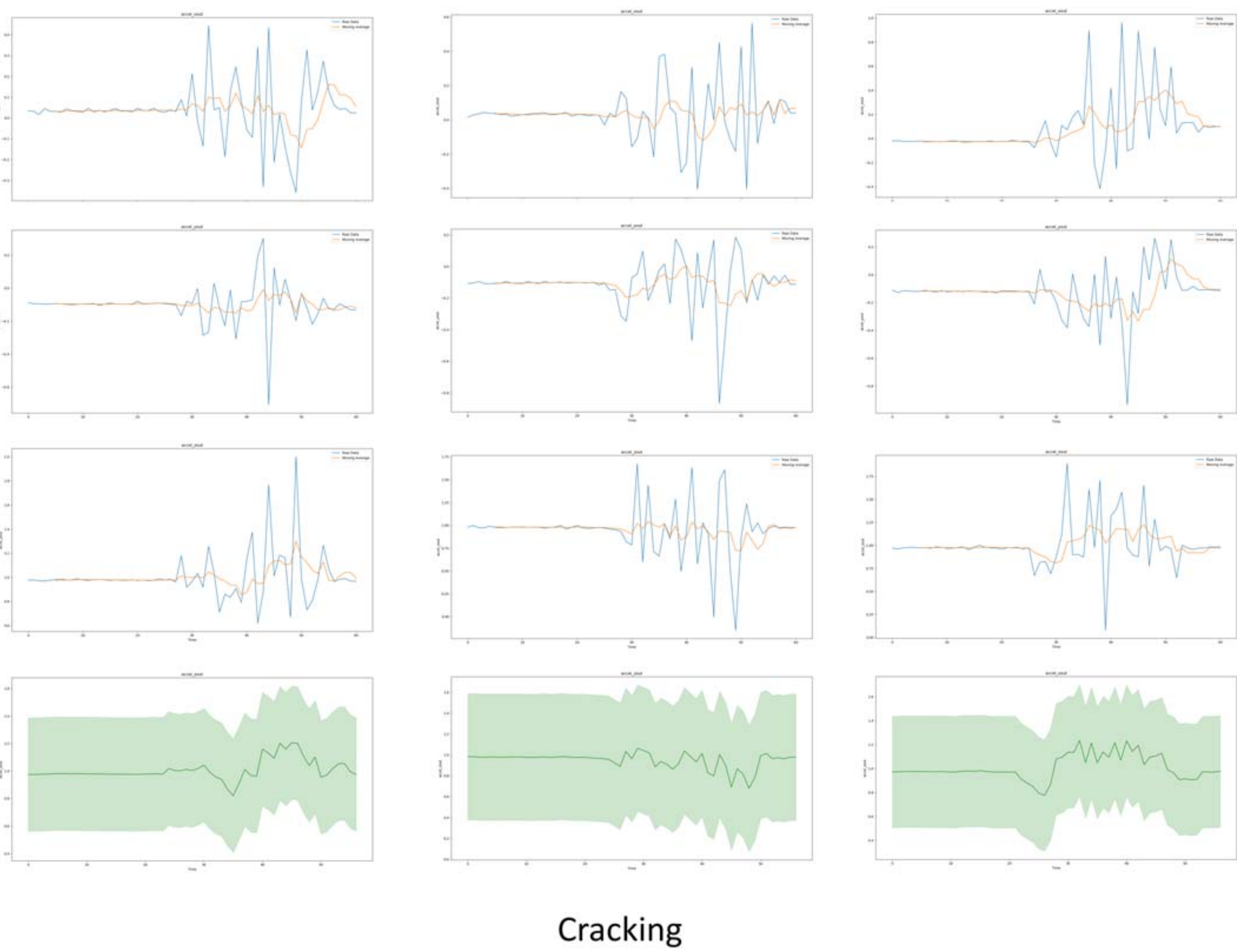
Our experimental works were conducted independently in Zhangzhou and Nanjing, China. The team collects the hand-labeled signals by consistently moving several known roads and recording accelerometer and GPS data. Using mobile phone's network hotspots, we control the Raspberry Pi with a remote-control app Anydeck and communicate the captured accelerometer and GPS signals to OpenChirp.



4.5 Pothole detection

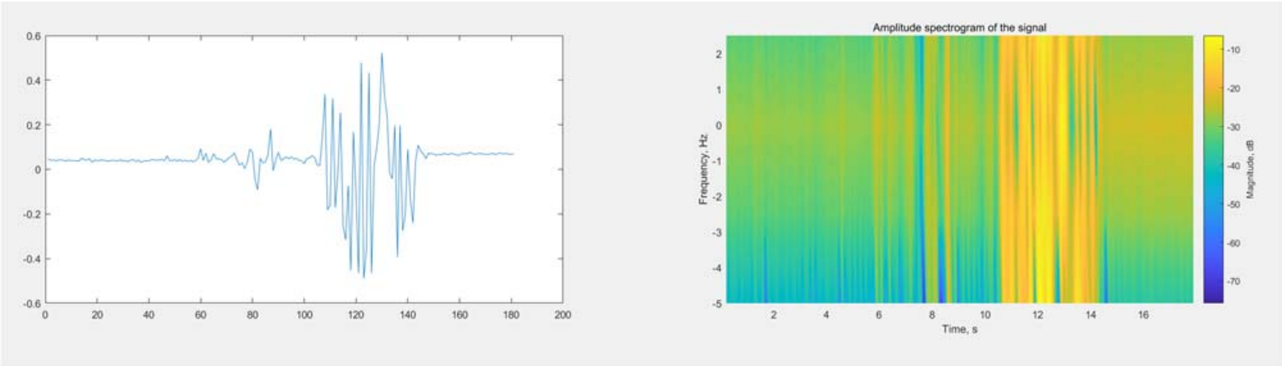
First, we fill the missing values with linear interpolation and then use the moving average method to smooth the accelerometer signals.The X, Y, Z axis acceleration of all defects are shown below. 95% confidence interval of the Z-axis acceleration of three kinds of defects are also drawn. Different defects have different data distributions and patterns. As for the acceleration duration, the Potholes have the shortest period, especially with dramatic changes in amplitude in a short time. The durations of acceleration of the Cracking last for the longest time, but the variation range of acceleration is more mild compared to the Pot Holes. The changing patterns of acceleration for the Upheaval are moderate between the Potholes and Cracking.



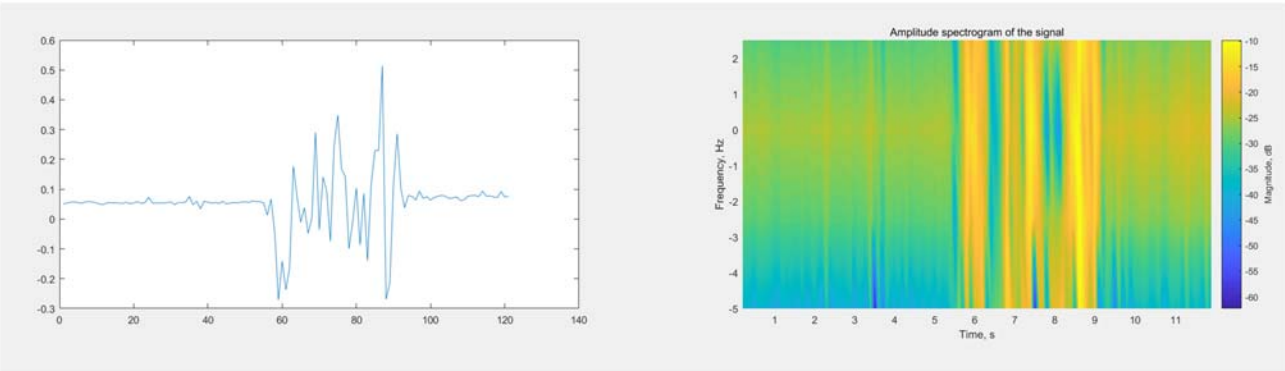


Short-time Fourier transform (STFT) analysis

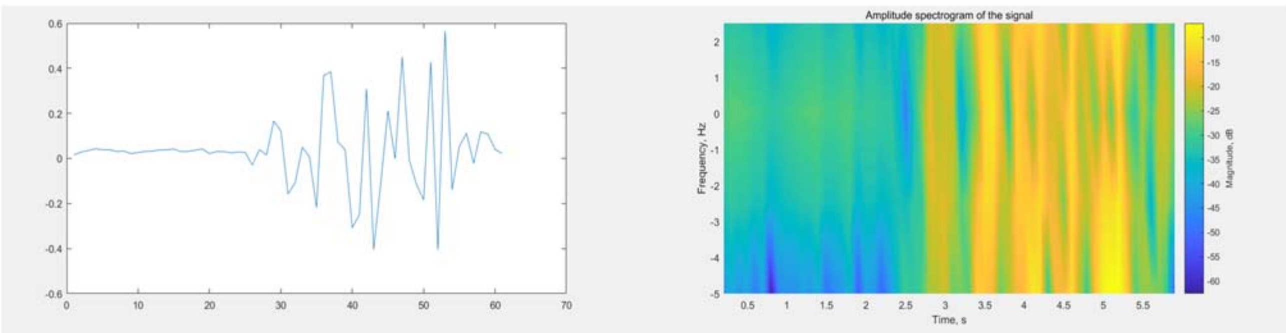
To analyze the patterns of different signals, Short-time Fourier transform (STFT) [9] is used to divide the long Z-axis acceleration signal into short parts and calculate the Fourier transform on each part to get the complex frequency, amplitude, and phase content of the Z-axis acceleration signal developing with time. As shown in the figure, the Y-axis is the frequency (Hz) of the Z-axis acceleration signal, and the X-axis is time (s). The time-frequency distributions (amplitude spectrogram) of the Z-axis acceleration signal vary between the Upheaval, Potholes, and Cracking.



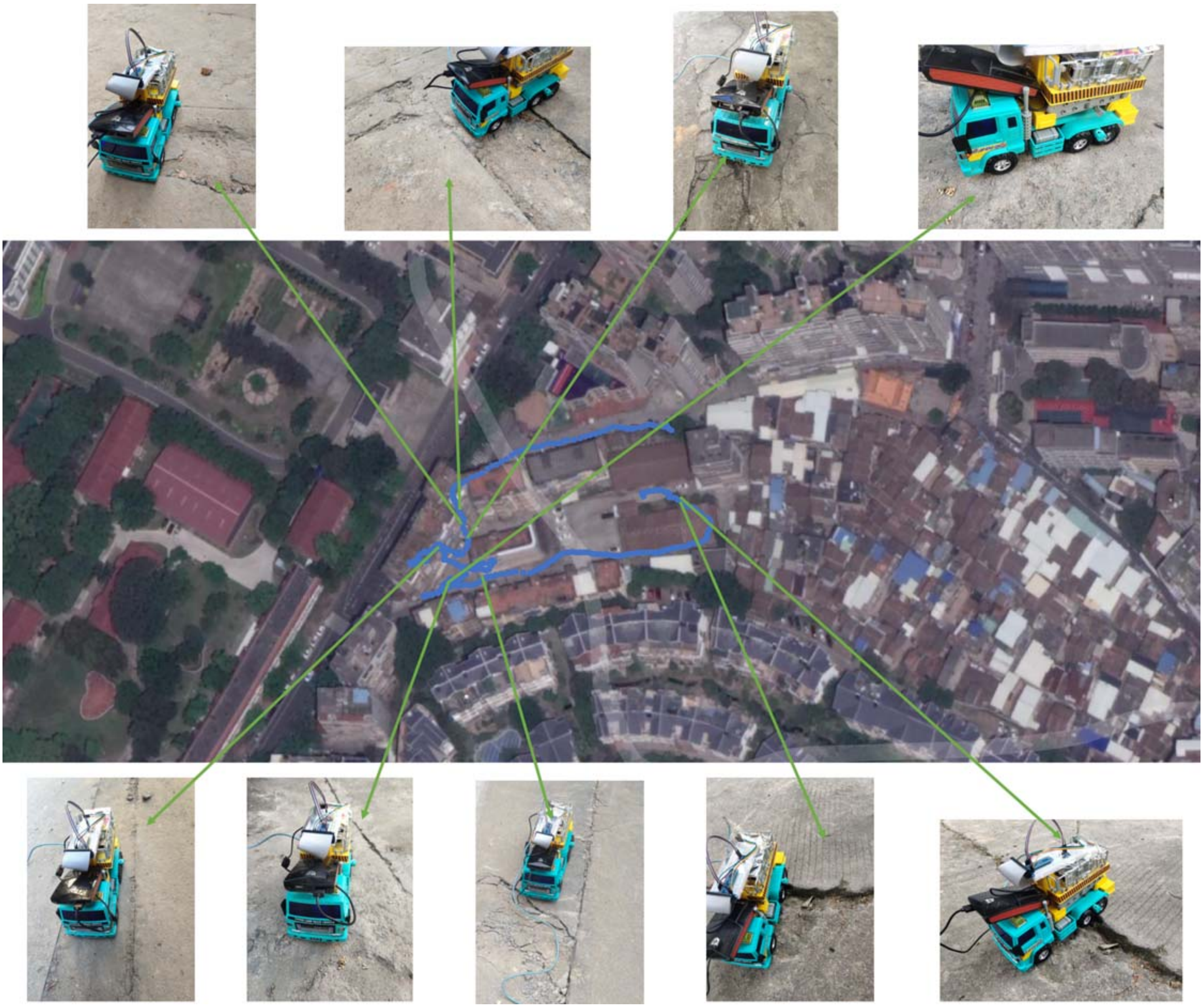
Pot Holes



Upheaval



Cracking



5 Discussion

- This project designs and implements the patrolman system, using Raspberry Pi for road pothole inspection. The system consists of a mobile platform (we use the toy car in the experiments), an MPU-6050 accelerometer, a GPS sensor, and a power bank.
- By tracking data from the accelerometer and GPS sensors, the team uses the captured signals to evaluate road surface conditions.
- We aim to analyze three main types of road defects, including Cracks (CR), Potholes (PH), and Upheaval (UH). The abnormal road conditions can be sensed through the acceleration data during a drive. Comparing hand-labeled signal patterns, the team identifies three typical road defects.
- We pubilsh all sensor values onto OpenChirp to visualize captured data on the website.
- In the future, we will apply a machine-learning based approach for automated detection of diverse road conditions.

References

[1] Swindonian (2018). More than half a million potholes were reported last year throughout the UK.

[2] Ho, C. H., Snyder, M., & Zhang, D. (2020). Application of Vehicle-Based Sensing Technology in Monitoring Vibration Response of Pavement Conditions. Journal of Transportation Engineering, Part B: Pavements, 146(3), 04020053.

[3] Japan International Cooperation Agency (2016). Pavement Inspection Guideline. https://openjicareport.jica.go.jp/pdf/12286001_01.pdf.

[4] Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., & Balakrishnan, H. (2008). The Pothole Patrol: Using a mobile sensor network for road surface monitoring. In MobiSys'08 - Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (pp. 29–39).

[5] L76X GPS Module User Manual. https://www.waveshare.com/w/upload/5/5b/L76X_GPS_Module_user_manual_en.pdf.

[6] Brett Neal, 13 Pavement Defects and Failures You Should Know. https://www.pavemanpro.com/article/identifying_asphalt_pavement_defects/.

[7] Simple Device Tutorial, https://github.com/OpenChirp/docs/wiki/simple-device-tutorial.

[8] MPU6050 (Gyroscope + Accelerometer + Temperature) Sensor Module Wiki, https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module.

[9]Short-time Fourier transform, https://en.wikipedia.org/wiki/Short-time_Fourier_transform.
