Motorcycle Crash Detection and Alert System using IoT

Dr. G. *Karuna*[1], *R. P.* Ram Kumar[1*], *Vadlapatla* Tarun Sri Sai[2], *Jula* Abhishek[2], *Mood* Shashikanth[2], *Burra* Kashyap[2]

[1]Department of AIMLE, GRIET, Hyderabad, Telangana, India
[2]UG Student, Department of CSBS, GRIET, Hyderabad, Telangana, India

Abstract. Motorcycle travel is considered the most dangerous mode of transport in the world. Reports suggest that the fatality rate of motorcycles is 212.7 deaths for every million miles travelled on motorcycles. Unlike other forms of travel like cars, buses, etc, motorcycles expose the rider to their surroundings. In cars, the frame protects the driver from hitting the road or falling out of the car. But motorcycles do not have such a possibility. Therefore, the best way to minimize fatalities in accidents is to have an alert system that can alert the emergency services when it detects an imminent crash. This is where the motorcycle crash detection and alert system comes into the picture. It uses the MPU6050 Multi-axes accelerometer to detect when the motorcycle falls to its side. It sends the impact parameters to Firebase cloud and if the values meet the crash criteria, it sends an alert to the emergency contacts as well as to the emergency response services, who can then act according to it.

1 Introduction

IoT, which stands for 'Internet of Things' is defined as a system of interrelated devices or entities consisting of sensors and intelligence connected by a network that is responsible for the implementation of an automated software. The term Internet of Things was coined in 1999, by a computer scientist by the name of Kevin Ashton. While working in Procter & Gamble, he came up with an idea of using RFID (Radio Frequency Identification) chips on products to track them through the supply chain. To attract the attention of executives, he decided to go with the word 'Internet', which was a popular subject of discussion at the time. His idea of using the buzzword seemed to have worked, since now IoT has become one of the leading technologies in the entire world. There are a myriad of applications ranging from home security to the entire agriculture industry being dependent on IoT devices to save time and money.

An IoT system consists of smart devices that use embedded systems with the help of processors, sensors and other communication devices, to collect and operate on data that they acquire from their environments. IoT devices transfer the sensor data that they collect with the help of an IoT gateway, where data is either sent to the cloud to be analysed or analysed locally. There are also instances when these devices communicate with other related devices

* Corresponding author: ramkumar1695@grietcollege.com

and mutual sharing of data to act on immediately. The devices do most of the work without human intervention, although people can interact with the devices; for instance, to set them up, give them instructions or access the data. IoT can also make use of machine learning technologies and algorithms to make it easier to collect and process data while doing so with precision and accuracy. Modern IoT devices almost always consist of AI chips to facilitate faster Artificial Intelligence applications. Figure 1.1 shows an example of an IoT System. Any IoT system primarily consists of three phases: Data collection, Collate and transfer, and Data analysis. Examples of IoT devices that collect data are sensors, antennae, microcontrollers etc. IoT hub or gateway acts as a medium of communication between sensors and the user interface.

2 Literature review

The Smart Helmet System aims to be the solution for all motorcyclists who want it [1]. Increases awareness and safety while driving. Concept of closeness Sensing in automobiles is not a new concept and has been implemented in various papers and commercial products, before this. However, such insights tend to have merit with its own set of pros and cons that come with its design and Implementation of smart helmet systems aimed to maximize benefits. Similar products while avoiding or minimizing disadvantages. Its advantages are as follows;

- All the components required are easily available.
- No manual attention is needed.
- Automatically controlled and easy to us

Its disadvantages are as follows;

- In this paper usage of relays leads to consuming more power.
- One time investment cost.

Many accidents can be prevented and lives saved if intelligent braking systems are implemented [2]. The installation of such complex systems is often as mandatory as the use of seat belts. This means that injuries are often avoided to some extent. Our intelligent brake technology offers a snapshot of Long-term protection for your vehicle and how these particular devices are in more ways refinements are also used to prevent collisions and protect car occupants after being combined into a single car system. Its advantages are as follows;

- Smart braking system helps to prevent wheel lock up and therefore can keep the rider upright
- Smart braking systems can help to reduce braking distance and in the event of a crash.
- Smart braking system reduces the impact of speed.

Its disadvantages are as follows;

- Smart braking system is the increased cost it adds to the overall cost of a vehicle.
- Maintenance costs go up as the sensors on each wheel are expensive and get heavy on the pocket if they run out of calibration.

The side stand is used to support parked motorcycles [3]. If the driver forgets to fold the side stand beforehand, then hit and touch the ground in a non-distracting stance. The driver controls during the curve. A day sensor is now used to ensure the stand is in a released state. The motorcycle side stands offset metal rods and coil springs from the centre. Some side stands automatically retract when parked. Some have raised supports, while others are equipped with electric locks. Warning device or special retraction mechanism. Its advantages are as follows;

- The automatic side stand works on the simple mechanism and no need to take extra power while operating

- The design of the vehicle is not affected, only a simple mechanism is added to the vehicle.
- No electronic control required.

Its disadvantages are as follows;

- Maximum Fatigue
- Only indicate the position.
- Expensive

A device that transmits information about the crash site to local rescue workers is essential for a rapid response [4]. Many scholars have proposed various automatic collision warning systems in the research literature. These include smartphone-based incident detection, Global System for Mobile Communications (GSM), Global Positioning System (GPS). Technology, vehicle ad-hoc networks, various machine learning algorithms, mobile apps. All vehicles must be equipped with automatic detection of traffic accidents and an information communication system. Its advantages are as follows;

- We reduce the death rate due to accidents.
- Find out the stolen vehicle's exact location by the user.

Its disadvantages are as follows;

- Costlier controller
- Camera adds on to the cost of the overall system.

3 Proposed method

3.1 Problem statement

Motorcycle transport is considered as the most dangerous form of transport in the entire world. A survey conducted by the National Highway Traffic Safety Administration states that around 212.7 deaths are reported for every 100 million Vehicle Miles Travelled (VMT) on a motorcycle. Though motorcycles have become relatively much safer in the past decades, they still remain the cause for the most deaths in the transportation field. Motorcycle accidents are very common nowadays.

In 2019 alone, motorcycles accounted for 3% of all vehicles registered in the United States, but motorcyclists accounted for 14% of all traffic fatalities, according to the National Highway Traffic Safety Administration. An estimated 84,000 motorcyclists were also injured in accidents that year. Inherently, motorcycles are very dangerous. However, safe driving practices and helmets can reduce injuries and deaths. Collisions between left-turning vehicles and motorcycles are common and often fatal. In 2019, NHTSA reported that nearly half of all fatal accidents involving both cars and motorcycles involved a left turn at the time of the accident.

This paper aims at detecting a motorcycle crash, confirming the crash is not a false positive, and subsequently alerting the provided emergency contacts with details of the location and time of crash. This will provide enough information for the receiver to immediately call the necessary authorities to the crash site and provide the necessary help. This will also make it easier for the receiver to locate the crash in case the crash happens in their vicinity so they can go there themselves and assess the situation.

This is done by using a sensor called MPU 6050 which is a multi-axes accelerometer and gyroscope sensor to detect when a motorcycle has toppled over and a crash occurs. When fall happens, the position status is updated to the Firebase database which is then read by the rider’s smartphone. The smartphone then gets the location latitude and longitude, and sends the details of the crash to the emergency contact provided in the smartphone app itself. When there is a false alert, the driver can just press a button and disable this message.

3.2 Objectives

- Utilize the multi-axes accelerometer readings of the MPU6050 accelerometer sensor to detect an imminent crash of a motorcycle on the road by recording the event of the motorcycle toppling over.
- Develop a false alert service that can be used by the rider in case they feel safe or there was a mistake in the crash detection.
- Notify the emergency contact about the rider's motorcycle crash to let them alert the emergency services.
- To reduce the effective time taken by the emergency services to reach the crash site by providing necessary location coordinates and time of crash.

3.3 Architecture diagram

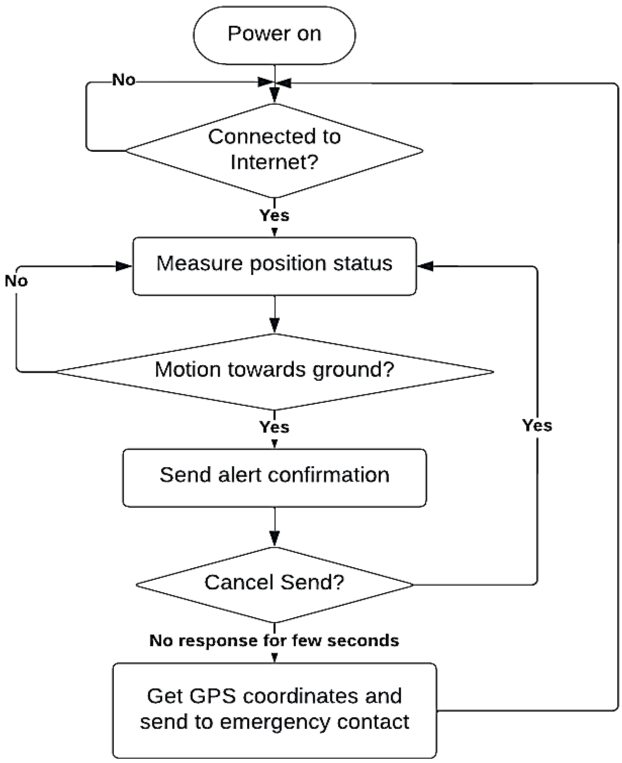

Fig. 1. Architecture diagram.

There are several phases that the program runs through every time it is powered on

1. Power on: The system is powered on, by linking it to the engine start/stop button. When the system starts, the GISMO-VI module blinks the blue LED indicating trying to connect to the internet. Internet connection is required to feed the database with position status and also read the custom motion threshold value set in the app.
2. Internet connection: Internet connection is given by using the mobile hotspot of the rider's mobile phone. The mobile hotspot credentials are initialized during the installation of the GISMO module to the motorcycle. This part needs to be done manually, everything else can be customized to the rider's liking.

3. Connection Retry: Wi-Fi connection status is read once every 30 seconds until the module is connected to the internet. In the case, the module doesn't get internet access in the latest Wi-Fi status reading, it resets the counter back to zero and resets the entire ESP32 program and tries to connect again.
4. Measure position status: Once the module is connected to the mobile hotspot of the rider's smartphone, it starts reading the accelerometer values of the MPU6050 sensor. At the same time, the location sensor of the phone is also used to constantly update the location coordinates of the motorcycle. This takes place in the smartphone at the pace of once every 60000ms or 60s. The accelerometer sensor reads the values in forward-backward and left-right tilt directions given by x and y axes respectively. The combination of both x and y axes is read as the z axis input, which is used to detect the motorcycle crash. The following reading takes place once every 1000ms.
5. Crash detection: A crash is determined by the absolute difference in magnitude of the motorcycle's relative velocity from 1 second ago. In normal conditions, the firebase is updated with the status "Upright" once every second. If the absolute difference in magnitude is greater than the specified motion threshold of the motorcycle set using the smartphone, a crash is detected. In the crash scenario, the Firebase receives the position status as "Fell". After this happens the rest of the phases are handled by the smartphone app.
6. False Alert notification: After receiving the position status as "Fell", the smartphone immediately begins a 10-second counter. A large red button that says "False Alert" appears for the user to press in case there isn't a real emergency or if the crash detection ends up being a false positive. During this time, the accelerometer stops reading values to ensure there are no multiple alert scenarios from the same crash. A warning is also announced as "Crash detected. Press False Alert to turn off alert".
7. Button pressed: On pressing the button within the 10-second period, the position status is reset to "Upright" and the alert notification won't be sent to the emergency contact. This feature exists in case there isn't a real emergency or if the crash detection ends up being a false positive.
8. No response: If the false alert button is not pressed within the 10-second period, on the 10th second the location coordinates are concatenated into a message. This message is then sent to the emergency contact using the phone's module itself. Once the message has been sent, the position is reset to "Upright" by the smartphone itself, the accelerometer again starts reading the values and updating to the Firebase, and the False Alert button appears. Even if the false alert button is pressed after the 10-second period, the alert will still be sent to the concerned emergency contact.

3.4 Modules and description

This paper is implemented in a total of five modules. The five modules are

1. Preparing the sensor module.
2. Connecting to Firebase.
3. Preparing thresholds for readings
4. Incorporating into the mobile app.
5. Testing the threshold values.

3.4.1 Module 1: Preparing the sensor module

In this module, the GISMO-VI code will be coded using Arduino IDE for the paper. This module implements the following features into the paper

- Read accelerometer values in x, y and z axes

- Display acceleration values in x, y and z axes in Serial Monitor
- Plot acceleration values in x, y and z axes in Serial Graph
- Read Firebase Threshold Limit for motion threshold set through mobile app.
- Calculate position status using the accelerometer readings from the MPU6050 Accelerometer.
- Send position status to Firebase database.
- The following libraries are used to implement the above features
- #include "I2Cdev.h": The I2C Device Library (i2cdev) is a most unified and well documented set of classes that provide simple and intuitive interfaces to I2C devices. Each device is designed to use a generic "I2Cdev" layer, which abstracts away the I2C communication at the bit and byte level of each specific device class, making it easy to keep the device layer clean while providing provides a simple way to modify just one layer to transcode I2C communication to different platforms (Arduino, PIC, MSP430, Jennic, simple bit-banging, etc.).
- The device classes are designed to fully provide all the features described in the documentation of each device, as well as all the useful general convenience functions. There are many examples in many classes that demonstrate basic usage patterns. The I2Cdev class is designed for static use, which reduces memory requirements if you have a lot of I2C devices in your paper. Just one instance of the I2Cdev class. Recent additions in late 2021 also allow for the transmission of non-default Wire objects (in the Arduino environment) to allow multiple I2C transceivers to be used at the same time, especially due to the number of people wanting to use them up to four people. The MPU-6050 IMU has no associated I2C mux IC.
- #include "MPU6050.h": This library is used to work on the MPU6050 multi-axes accelerometer and gyroscope.
- #include "Wire.h": This library allows communication with I2C/TWI devices. On Arduino boards with R3 layout (1.0 pinout), SDA (data line) and SCL (clock line) are on the pin header near the AREF pin. The Arduino Due has two I2C/TWI interfaces SDA1 and SCL1 near the AREF pin and another one at pins 20 and 21.

3.4.2 Module 2: Connecting to Firebase

In this module, the arduino code is written to connect the MPU6050 to the Firebase. To connect to Firebase and send the position status as well as read the motion_threshold values, internet is required. Following is the code implemented for Module 2.

- float motion_threshold = 0.9; // This is the current motion_threshold
- const float DEFAULT_MOTION_THRESHOLD = 0.7; // Default threshold
- WiFiInit is used to connect to the mobile hotspot of the user. It uses the Wi-Fi hotspot credentials from "Credentials.h" to connect to mobile hotspot. In addition to that, it also includes code to restart esp when wifi is not connected for 10 seconds.

3.4.3 Module 3: Preparing threshold value for acceleration

In this module, the default threshold value has been prepared. After some amount of testing, it is found that 0.7 works best for detecting a crash with 96% accuracy.

3.4.4 Module 4: Incorporating into the mobile App

In this module, the mobile app companion for GISMO is developed. Kodular creator software has been used to develop the Motorcycle Crash Detection Companion. The features that have been implemented in this module are

- GPS Location: Use Kodular's Location Sensor feature to read the latest location coordinates of the smartphone to keep track of the location when the crash occurs. This is updated in a 60-second interval.
- Position Status display: Position status "Upright" or "Fell" is updated once every 1000ms to the smartphone app. Figure 3.12 shows the implementation of this block of code. To update once every 1000ms there is a clock component being used that will fire once every 1000ms. When the clock is fired, the value is retrieved from the Firebase database using the token and database secret provided.
- Crash Threshold update: Crash threshold value can be updated through the app as well.
- False Alert button: False alert button is used to prevent accidental alerts and false positives. Figure 3.14 shows the implementation of the False Alert button. False Alert button will stop the alert procedure, reset the position to upright and restart the 1000ms accelerometer timer.

3.4.5 Module 5: Testing the threshold values

This module is implemented with the deployment of the app. The threshold values will be tested for the type of motorcycle type and then edited through the mobile app's crash threshold update feature.

4 Results and discussions

4.1 Experimental results

Experimental Results have been collected using the Arduino IDE's Serial Plotter feature. The accelerometer readings in terms of g's from three axes have been plotted on the graph against time in seconds. Four different scenarios were plotted on the graph

1. Upright position (Traveling on road)
2. Forward Crash Maneuver.
3. Left Side Fall Maneuver.
4. Right Side Fall Maneuver.

4.1.1 Graph for upright position

We observe here that the accelerometer readings barely change at all when the motorcycle is not crashed. So, when the motorcycle is in upright position, the ax, ay and az values only have a peak value of 0.3 during hard tilt on a curvy road. Otherwise, they're in a range of 0-0.1 for most of the ride. The legends are as follows;

- Blue curve - Y axis
- Red curve - X axis
- Green curve - Z axis

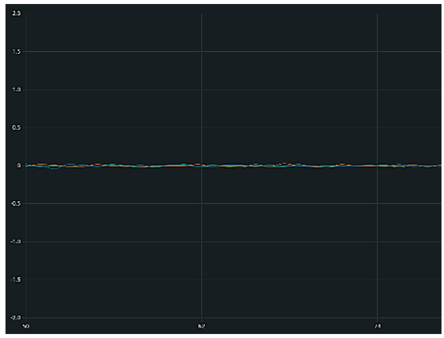

Fig. 2. Graph for 3 axes when the motorcycle is in upright position.

4.1.2 Graph for forward crash maneuver

Although the accelerometer reading in the y axis doesn't seem to cross a value of 0.5. This is because of the fact that the motorcycle never moved sideways during the entire event of the forward crash.

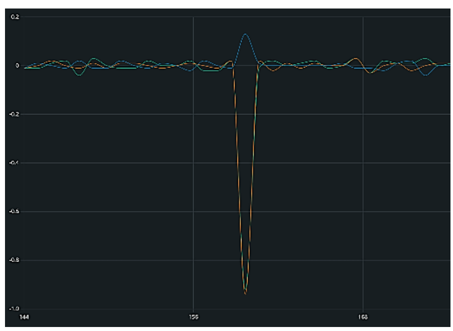

Fig. 3. Graph for 3 axes when the motorcycle is in forward crash maneuver.

Although the accelerometer reading in the y axis doesn't seem to cross a value of 0.5. This is because of the fact that the motorcycle never moved sideways during the entire event of the forward crash.

4.1.3 Graph for left side fall maneuver

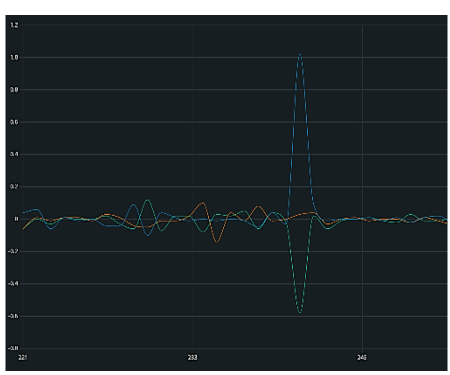

Fig. 4. Graph for 3 axes when the Motorcycle is in the left side fall maneuver.

This graph shows how during a left side fall maneuver, the blue curve or the Y-axis curve as well as the green curve or the Z axis curve is showing a steep incline while the red curve or the X axis curve is in neutral position. This indicates that the motorcycle is moving in a positive y-axis or left direction in this demonstration. The green curve indicates the fall of the motorcycle in the maneuver.

4.1.4 Graph for right side fall maneuver

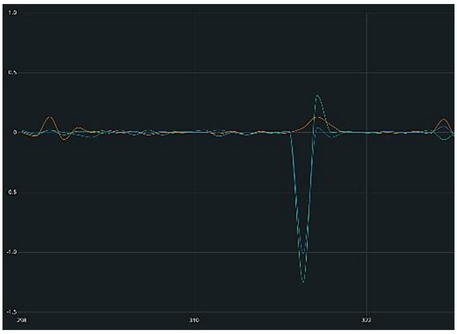

Fig. 5. Graph for 3 axes when the Motorcycle is in the right-side fall maneuver.

This graph shows how during a right side fall maneuver, the Y-axis curve as well as the Z axis curve is showing a steep incline while the red curve or the X axis curve is in neutral position.

5 Conclusion

IoT systems consist of intelligent devices that use embedded systems with the help of processors, sensors, and other communication devices to collect and process data received from the environment. IoT devices send collected sensor data using an IoT gateway. IoT gateways send data to the cloud for analysis or analyse it locally. These devices may also communicate with other related devices, exchanging data with each other and taking immediate action. Humans can interact with devices, but devices do most of the work without human intervention like access to Settings, Instructions, or Data. Most motorcycle accident fatalities are the result of delayed response, not deaths at the scene. If a driver is involved in an accident, they are more likely to be shocked. Even with an impact, they may or may not be able to call for help on their own, as either their hands, head, or chest can be damaged during the fall. Being exposed to the elements makes this type of injury very likely. Assuming the clash occurred in a crowded, or at least deserted, area, there is still no guarantee that an outsider would take the responsibility into their own hands and seek help or call emergency services. This kind of thinking makes things difficult. No one calls for help, delaying necessary emergency response. Others are too busy to help strangers when they have more important things to do. For this reason, it is necessary to notify important persons for the driver in such situations.

The paper aims to detect motorcycle accidents, verify that it is not a false alarm, and alert emergency contacts providing details of the location and time of the accident. This ensures that the recipient has enough information to contact the necessary authorities at the crash site immediately and provide the necessary assistance. It also makes it easier for the recipient to go to the scene and grasp the situation, even if an accident occurs nearby. This paper can further be extended by implementing some additional features. The amount of contacts can also be extended to have a list of emergency contacts who will receive the details of the crash.

It can also be implemented in a way to add a location address to each contact and then send only to the three nearest contacts to the crash site. This will increase the likelihood of getting proper help in times of a motorcycle crash.

References

1. J. Bonnells, J. De Gouveia, J. Reimers, B. Scherschel, Senior Design Group 22, University of Central Florida, Fall 2016 - Spring 2017, (2017)
2. M. Kumari, S. Kumar, A. Kumar, N. Kumari, Int. J. of Sci. and Res. 9, 11, (2020)
3. Automatic Motorbike Side Stand Slider, Available: Online: https://nevonprojects.com/automatic-motorbike-stand-slider/
4. S. Parameswaran, P. Anusuya, M. Dhivya, A. Harshiya Banu, D. Naveen Kumar, Inter. J. of Engg. Res. & Tech. 4, 11, (2016)