Developing Intelligent Software Interface for Wireless Monitoring of Vehicle Speed and Management of Associated Data

Mominul Ahsan[1], Julfikar Haider[2], Jennifer McManis[3] and M. S. J. Hashmi[1]

[1]School of Mechanical and Manufacturing Engineering, Dublin City University, Dublin, Ireland

[2]School of Engineering, Manchester Metropolitan University, Manchester, United Kingdom

[3]School of Electronic Engineering, Dublin City University, Dublin, Ireland

Corresponding author

Mominul Ahsan

School of Mechanical and Manufacturing Engineering

Faculty of Engineering and Computing

Dublin City University

Dublin-9, Dublin, Ireland

Email: md.ahsan2@mail.dcu.ie

Tel.: 0035317007947

ABSTRACT

The aim of this work is to develop an intelligent wireless system for monitoring vehicle speed, identify speeding vehicles and imposing penalty for the speeding offenders. A prototype system has been developed in a laboratory environment to generate random speed data using a mechanical wheel (acts as a vehicle), measure the speed data with a Shimmer wireless sensor and transfer the data wirelessly to a server computer for further analysis. Software interface has been developed using Java based socket-programming to monitor the vehicle speed in a server computer and to send the data associated with a speeding vehicle to a remotely placed client computer. The functionality of the software has been tested by experimenting different traffic scenarios. If the vehicle speed is higher than the set speed limit for the road, the system automatically detects it and generates a report with the time of speeding, vehicle number, vehicle speed etc. The report is saved in a central database (client computer) in order to take further necessary actions for the speeding offender. The experimental evaluation results show that the system can measure and monitor the vehicle speeds wirelessly and manage the speeding data automatically.

Keywords

Intelligent Transportation System; Vehicle Speed Data Management; Software Development; Java Socket Programming; Wireless sensor

1. INTRODUCTION

Human errors, in particular, driver errors are the causes for the most of the road accidents. It is reported that over 80% of all major crashes on Irish roads are caused from inconsiderate driving [1] e.g., vehicle over-speeding, driving after consuming alcohol, distraction during driving, non-adherence to traffic signals, tailgating, poor lane discipline etc. Managing vehicle speed data in an efficient and intelligent way is an on-going issue in modern transportation systems. Furthermore, the existing traffic systems are power hungry and connected through wires in terms of traffic data collection and management. Most of them are very expensive due to complex hardware as well as installation. In addition, some roads have no appropriate monitoring system for identifying the speeding vehicles. Therefore, vehicles passing through these roads and exceeding the maximum speed limits can cause fatal accidents. Although many systems have been developed in order to minimize the accidents, still further work is necessary to improve the situation.

The concept of Intelligent Transportation System (ITS) has emerged to improve transportation service, road safety and traffic efficiency by applying advanced technologies [2]. Road design, traffic design, traffic management, control and implementation, vehicle speed monitoring, Dynamic Route Guidance System (DRGS), traffic information collection are integral parts of ITS. U.S transportation department has classified the ITS applications in two groups: intelligent infrastructure systems and intelligent vehicle systems. Intelligent traffic control is an important part of ITS application to reduce traffic congestion and road accidents. Furthermore, increasing driver and pedestrian safety, enhancing convenience and mobility, and promoting productivity by expanding economic and social improvement are the blessings of ITS technologies. More recently, the application of sensor networks, sensor technologies and networking protocols in ITS has significantly improved traffic monitoring and management. Improving the efficiency of intelligent transportation system bears tremendous economical and environmental impacts.

In the view of social context, the U.S department of Transportation arranged budget around $200 billion a year to control the traffic congestion in May 2006. However, the traffic management systems in rapidly developing countries such as China and India are very poor [2].

The application of a wireless sensor network (WSN) in vehicle speed monitoring can reduce accidents through wireless sensing and communication [3-6]. Low cost, efficient and high performance of wireless sensors with integrated computer software can change the landscape of automatic traffic data collection and management. Fuel consumption, air pollution, traffic congestion and travel time can be significantly reduced through the integration of traffic control system with the wireless sensor technologies. Although there are some systems available for wireless communication, the operations of these systems are not in complete wireless mode. Furthermore, the existing systems only focus on detecting vehicle and calculating vehicle speed rather than developing a fully automated one in terms of vehicle detection, speed measurement, monitoring and data communication. Systems are also not available to manage and store speeding data automatically in a central database, which can eliminate any need for human intervention.

This research proposes a wireless sensor based framework for vehicle speed monitoring and managing the associated data. In this framework, a sensor (accelerometer-based) and other auxiliary devices are used to detect a moving vehicle. The vehicle speed is calculated by the sensor and the processing unit and cross checked with a speed limit set for a particular road. If the detected speed is greater than the set speed limit, the device sends the speed data to a central control center wirelessly and generates a report containing the details time, road and vehicle to implement the speeding penalty. In this work, a commercial wireless sensor (Shimmer^{TM}) has been used to develop the monitoring system.

The aim of the research is to develop a prototype system based on wireless sensor for detecting a vehicle, measuring and monitoring speed and managing the relevant speed data. In order to reach the aim, the objectives of this work are: (1) to develop an experimental set-up modelling the generation of vehicle speed in a laboratory environment (2) to design and develop a wireless vehicle speed monitoring software system and (3) to simulate real-life traffic monitoring scenarios using the experimental set-up and monitoring system.

2. LITERATURE STUDY

2.1. Sensor-based Traffic Monitoring Systems

Although speed camera is commonly used for road traffic monitoring, more recently a variety of sensors have been employed for vehicle detection, counting, classification and speed measurement. Generally, sensor-based traffic monitoring systems can be classified in terms of the placement of the sensors: roadway-based (on the road) and vehicle-based (inside the vehicles). Again, roadway-based sensors are categorized into two groups [2]. The first group is intrusive sensors, which include inductive loops, piezoelectric cables, pneumatic road tubes etc. The second group including video cameras, passive infrared, ultrasonic, microwave, passive acoustic array, laser, radar etc. is non-intrusive.

Mohan et. al. have developed a traffic monitoring system called TrafficSense, which uses a smart phone, a microphone, camera, GPS and accelerometer to estimate the speeds of moving vehicles [7]. An alternative traffic monitoring arrangement, known as Traffic-Dot has been reported by Coleri et al., [8]. The system consisted of a built-in processor, a radio and a magnetic sensor estimates the speeds of the moving vehicles using wireless sensor network. Sensys™ is a wireless vehicle detection system [9] that uses pavement-mounted magneto-resistive sensors to detect the movement of the moving vehicles, particularly the speeds of the vehicles. Magnetic sensor installed on the road has also been used for detecting vehicles [10]. Ma et al. [11] proposed an Automatic Vehicle Classification (AVC) system using wireless sensors (accelerometers and magnetometers), which reports vehicle arrivals and departures and only determines speeds of the moving vehicles. A vehicle speed monitoring system, proposed by Mastuo [12], measures the speed of moving vehicles on the road using an ultrasonic sensor and counts the number of vehicles. However, the system has not been expanded for detecting speeding vehicle as well as report generation.

Effective speed estimation and monitoring technique [13] has been used for collecting vehicle data wirelessly using loop detector and multi-sensor data fusion technique. The system does not focus on storing data automatically. Chen et al. [14] have used microphone array for detecting the sound waves generated by moving vehicles on the road. The sound waves are digitised and translated by computer programme using a correlation-based algorithm to measure the speed of the vehicle without considering the details of speeding vehicle. Luciani [15] has proposed a traffic monitoring system using transceiver and Bluetooth® wireless technology. One fixed transceiver located near the road communicates with another transceiver located inside a vehicle. The speeds are calculated by comparing with speed of the moving transceiver and reference speed on the road. However, the system is unable to generate evidence for speeding vehicles. CitySense [16] is consisted of 100 Linux-based embedded PCs outfitted with radios and various sensors and nodes. Although, it provides monitoring and storing data service to the end user by web-based interface, it does not provide facility for appropriate user data management.

GPS based vehicle speed monitoring systems have also been proposed by several researchers to communicate wirelessly between a moving vehicle and a control centre [17], to compare between vehicles' speeds and the pre-set speed limit on the road [18] and to obtain position data for speed measurement [19]. A speed control technique for heavy-duty vehicles under constant vibration acceleration has been reported by Endo et al. [20]. Agerholm [21] has reported an Intelligent Speed Adaptation (ISA) system, which calculates the right speed of the vehicle and its position and compares with the corresponding speed limit of the road to give warning to the driver in the event of speeding. The set-up of ISA system is complicated and the installation and maintenance costs are comparatively high. There are also some existing systems available for measuring vehicle speed. Boyraz et al [22] has proposed drowsiness architecture to predict drowsiness level as uncertainty in the steering wheel input. The system does not consider any action for speeding vehicle. An automatic speed violation system is used to calculate the speed between the distance of the two road side unit. The system can generate GUI to display the speed details on a front end user. However, it does not generate any automatic report for speeding vehicle [23]. Furthermore, an Average Speed Monitoring System (ASMoS) is used RFID technology for road vehicles in Brunei Darussalam. The system can generate GUI to display the vehicle speed. However, the system is unable to provide maximum speed and speed limit for a particular road on the GUI [24]. Furthermore, A GSM/GPRS module is used integrating Raspberry pi inside the vehicle using USB interface to transmit the updated vehicle information from database to the server. The system only track if a vehicle goes in the wrong direction or any accident happened rather than monitoring vehicle speed and taking action on speeding vehicle [25]. Another system is developed to receive, display and print the data sent from remote monitoring stations. However, the system does not compare the vehicle speed with speed limit on the particular road and associated speed data [26].

2.2. Traffic Data Management

Data management is one of the vital parts of road traffic monitoring. Conventional systems include a number of techniques for data acquisition from speeding vehicles. Roberts [27] has collated and sent traffic count data and other related data (e.g., location) using a short messaging by Publicly Switched Telephone Network (PSTN). Data collection device is embedded in a remote traffic-monitoring unit. Monitored traffic count data are routed from the traffic-monitoring unit to a central computer. The control information is routed from the central computer to the traffic-monitoring unit. The traffic count data is stored and further processed at a remote location. However, the system does not provide the real time report for speeding vehicles. Zhou et al. [28] used principal component analysis in a programming interface for monitoring traffic. In CarTel network [29], collected traffic data is transmitted to a portal via software to be monitored by the end users. TrafficView [30] is a wireless traffic monitoring system with programming interface that provides the driver with a dynamic view of the road traffic. The 'TrafficView' software on the node of the network periodically updates the speed status of the vehicle. Inductive loop detector can acquire data related to traffic flow, usage of the roadways and speed of the vehicles for continuous monitoring [31].

2.3. Critical Analysis of Literature

A critical analysis of the available literature reveals that there are several shortcomings of the existing road traffic monitoring systems. Some of the existing systems have high equipment cost, complex installation processes, high maintenance cost and dependency on weather condition. Other systems have difficulties in managing the collected data. The difficulties at the user end of the client computers for accessing and further processing is cumbersome in some of the prevailing systems. Literature also shows that no integrated system is available for facilitating vehicle detection, speed measurement and monitoring, and collating and managing vehicle speed data. This sensor device enables the user to transfer a large volume of data to the server. Shimmer sensor device generates vibration acceleration data. This small sensor device can capture and communicate data in real time. The applications of this sensor device include wireless magnetometer, accelerometer, gyroscope, GPS etc. It provides a range of functions including basic motion sensing, data capture, processing and wireless transmission of sensed data. Shimmer is designed for wearable and remote sensing applications and is highly flexible and adaptable. The benefit of this device is that it can be easily integrated and can interact with existing systems and technologies [32]. No evidence of employing the shimmer wireless sensor in traffic monitoring application was found in the literature. Furthermore, no study has considered the use of Java socket programming to develop data management software related to vehicle speed.

3. EXPERIMENTAL SET-UP DEVELOPMENT

First, automatic vehicle speed monitoring using wireless sensor and associated traffic data management system were conceptualised. Subsequently, the concept was implemented by developing a prototype experimental set-up using a mechanical wheel with the sensor in a laboratory environment. This section also presents configuration and calibration of the sensor, speed measurement and speed data collection.

3.1 System Architecture

In order to test the functionality of the general vehicle speed monitoring framework a prototype system was developed in a laboratory environment (Figure 1). A mechanical wheel was considered as a vehicle. A wireless sensor was attached to the wheel to measure the vibration of the wheel. The vibration data was sent to a server wirelessly and converted them to rotational wheel speed (rpm), which was again converted to linear speed in order to make it a realistic vehicle speed. The speed plot with respect to time was also transferred and displayed in a remote client. If the speed was greater than a set speed limit, a report was generated based on the speeding data to impose penalty for the offenders.

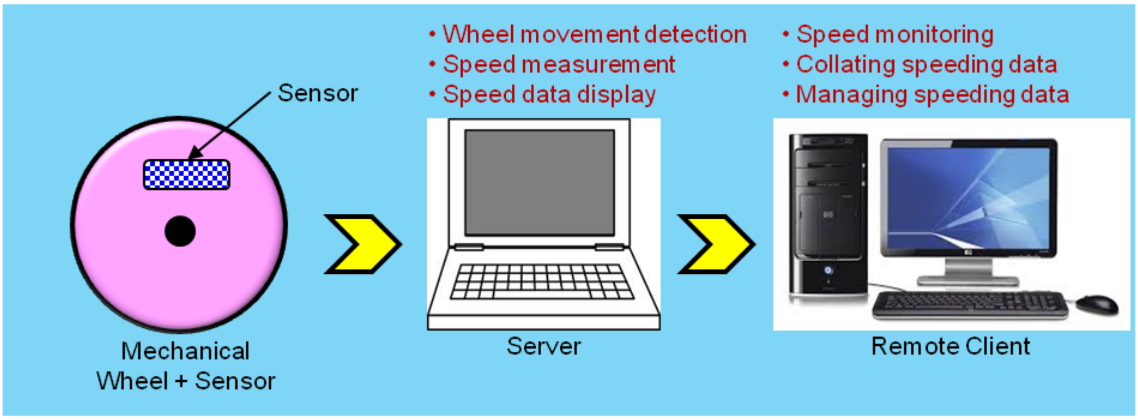


Figure 1: Prototype system architecture for monitoring vehicle speed

3.2 Sensor Configuration and Calibration

The Shimmer sensor measures the vibration acceleration. The vibration acceleration data was collected from the sensor by installing ShimmerConnect software in the server computer. The sensor was placed in a Docking station connected with the server via USB and configured in accelerometer mode. The connection between the server and sensor was established by Bluetooth pairing. Furthermore, the sensor device was calibrated with respect to the gravitational vibration acceleration ('g') before conducting the vibration measurement. The detail procedure for calibration is not in the scope of this paper.

3.3 Shimmer Sensor and its Communication

The Shimmer sensor device has an on-board microcontroller that has the capability of wireless communication via Bluetooth or low power radio. The device performs data storing through a micro SD card. Figure 2 illustrates the internal infrastructure of the shimmer sensor device. The device can sense motion using an integrated accelerometer. The device works as a baseboard for a full range of Shimmer wireless sensor modules e.g., Gyro, Magnetometer, GPS/temperature, or strain gauge modules [33].

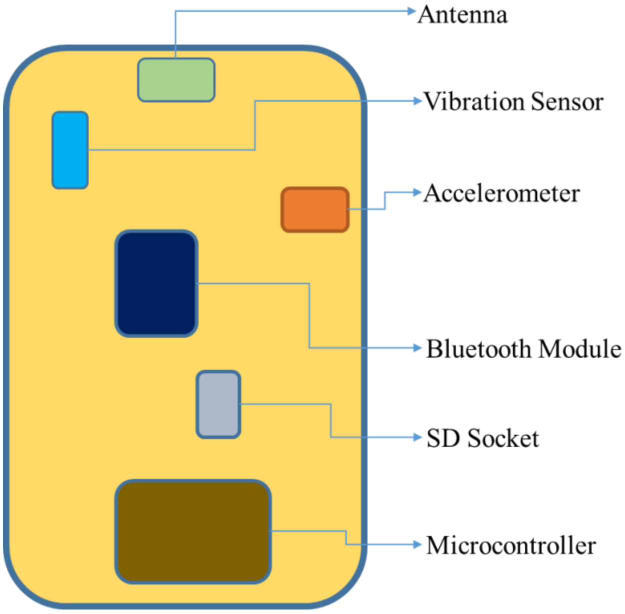


Figure 2: Internal infrastructure of shimmer sensor device

The connection between the computer and Shimmer sensor is conducted by Bluetooth pairing. The wireless connection between Shimmer sensor and the computer has been illustrated in Figure 3. The green light signal indicates the activation of connection. The Bluetooth connection between the server and the Shimmer device is still activated once the sensor device is removed from the dock. An appropriate port number, sampling rate, sample type and sample range are selected from the ShimmerConncet interface GUI. The front panel of the ShimmerConnect performs to stream vibration acceleration data.

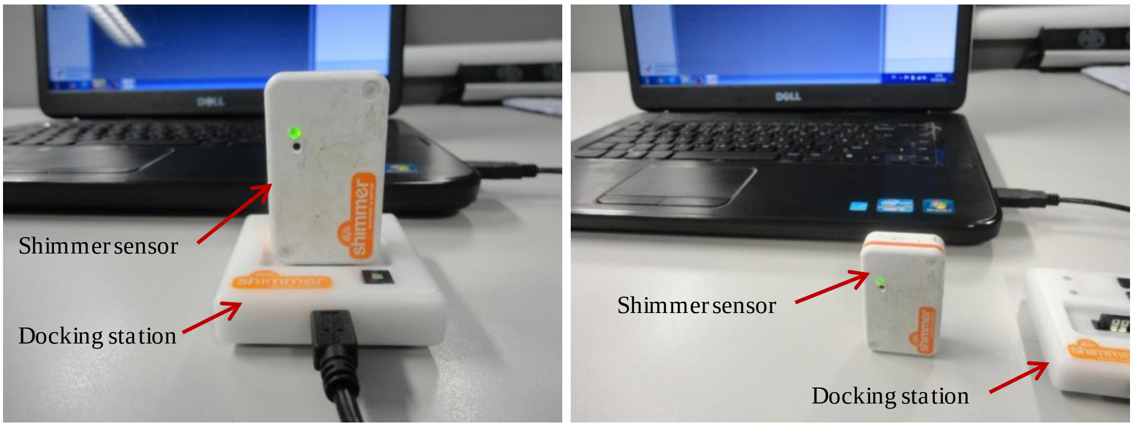


Figure 3: Shimmer sensor connected via Bluetooth with or without the docking station

3.4 Set-Up Development with a Mechanical Wheel

An experimental set-up was developed to generate continuously varying speed, which could be measured and transferred to a computer wirelessly. The set-up includes the Shimmer sensor [32], a mechanical wheel driven by a variable speed motor, an integrated speed controller and a server computer. The motor drives the wheel mounted on a shaft. The speed controller controls the wheel speed and the tachometer measures the wheel speed in rpm. The sensor was attached to the wheel as shown in Figure 4. The Shimmer sensor was connected with the server through Bluetooth connection to transfer the wheel vibration acceleration data wirelessly to the server where all the vibration data are stored for further processing.

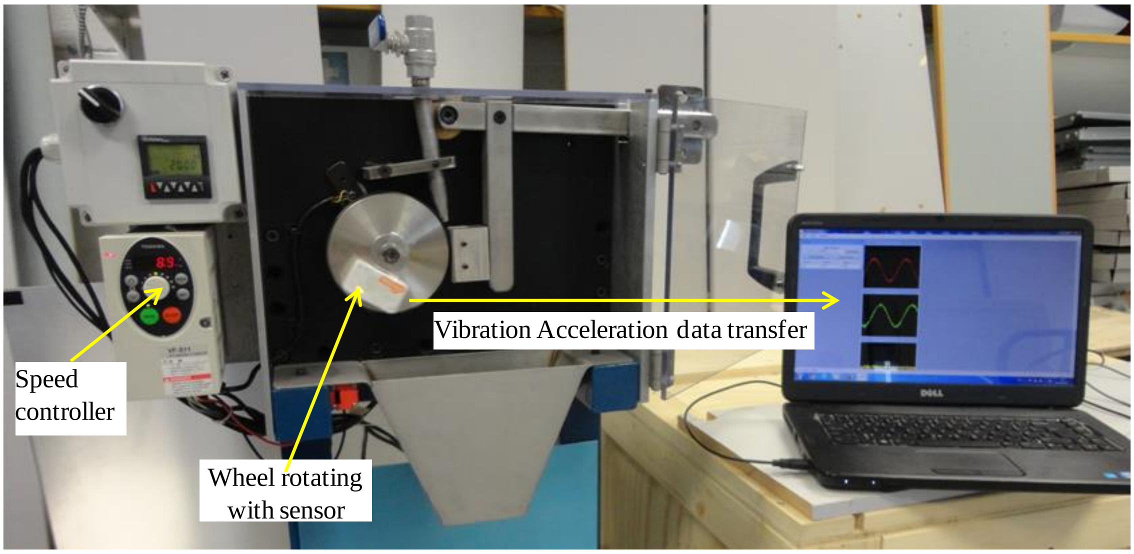

Vibration Acceleration data transfer
Speed controller
Wheel rotating with sensor

Figure 4: Connection establishment between the sensor and the server computer.

3.5 Wheel Speed Measurement

The wheel speed was measured with the sensor using a calibration technique. The purpose of the calibration was to obtain a feasible mathematical relationship between vibration acceleration of the wheel obtained from the Shimmer device and the rotational speed of the wheel. By setting the wheel at a known speed, the sensor indirectly measures the wheel speed through sensing vibration acceleration of the wheel.

The wheel can be rotated at a wide range of rotational speeds by the speed controller connected to the motor. In order to measure the wheel speed by the sensor, a set of wheel speeds was considered (10, 15, 20, 25, 30, 35, 40, 45 and 50 rpm). Vibration acceleration data in X, Y and Z directions were generated for different known wheel speeds. Initially, the sensor device was set at 100 Hz sampling rate and the wheel was run for 30 seconds at each rpm to collect substantial amount of data. The data were collected as CSV file through the ShimmerConnect interface page. Average vibration acceleration was calculated for each direction and for all speeds. From the average vibration accelerations in each direction, resultant vibration acceleration was calculated for all speeds using Eq. 1 [34].

$$R_{Accel} = \sqrt{X_{Accel}^2 + Y_{Accel}^2 + Z_{Accel}^2} \text{--(1)}$$

where, X_{Accel} = vibration acceleration in X direction, Y_{Accel} = vibration acceleration in Y direction and Z_{Accel} = vibration acceleration in Z direction

The resultant vibration acceleration data were plotted against different wheel speeds (Figure 5). It was noticed from the graph that the resultant vibration acceleration increased proportionally with the rotational speed of the wheel. Therefore, the relationship between the resultant vibration acceleration of the wheel while rotating at a certain rpm can be represented by a linear equation (Eq. 2)

$$R_{Accel} = 31.655\text{N} + 3024.8 \text{---(2)}$$

where R_{Accel} and N represent the resultant vibration acceleration and wheel speed respectively. For a known value of resultant vibration acceleration, the wheel speed N can be determined.

The relationship was validated by calculating the speeds using the equation and comparing that against the actual wheel speeds.

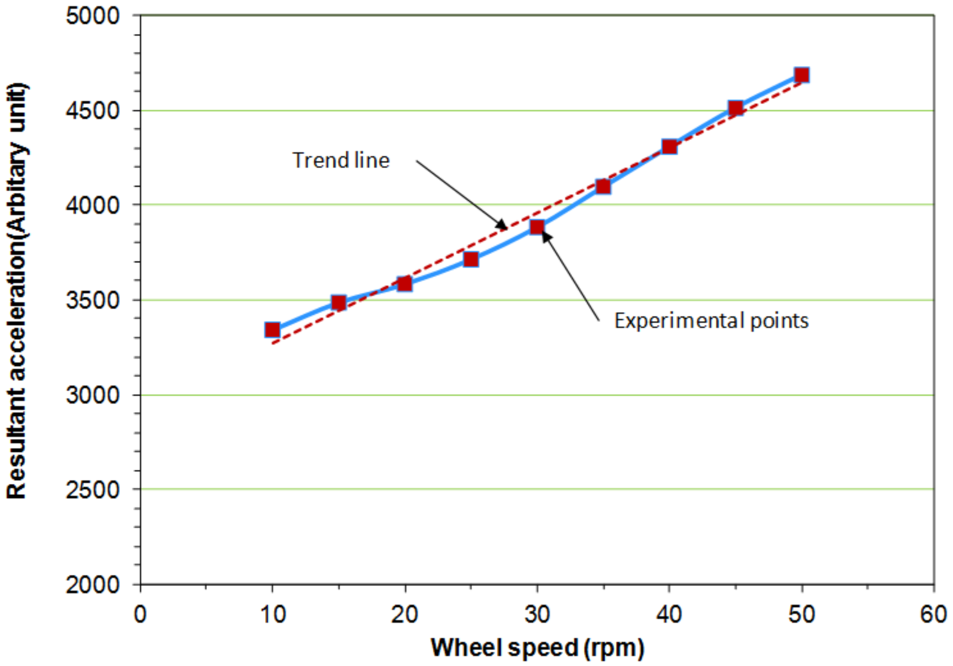


Figure 5: The resultant vibration acceleration plot against different rotational speeds (rpm) of the wheel

The calculated wheel speeds were in good agreement with the wheel speeds set during the experiment (Figure 6). This further demonstrated that wheel vibration measurement could lead to accurate calculate the wheel speeds.

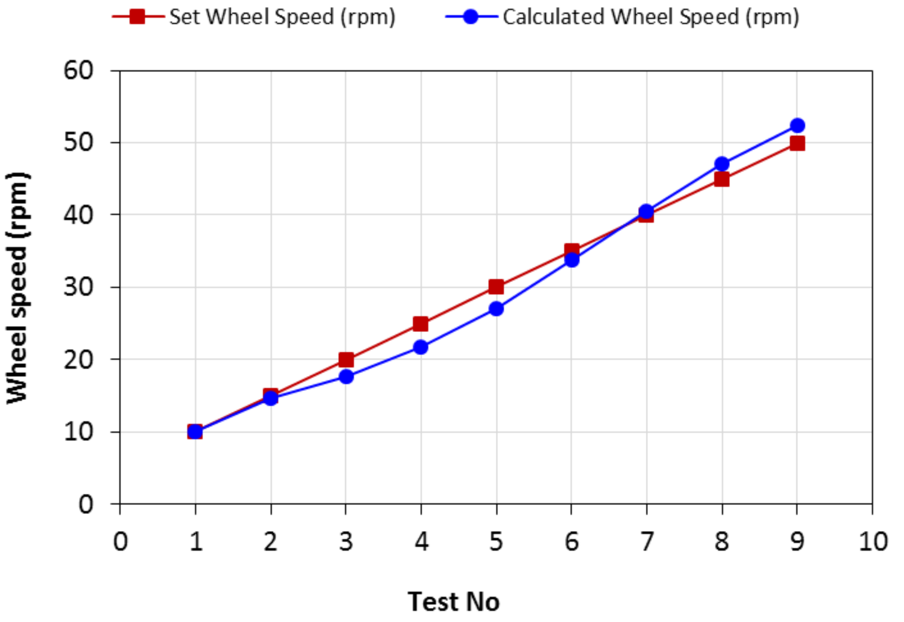


Figure 6: Comparison of set wheel speeds and calculated wheel speeds

The rotational speed of the wheel does not represent a real life vehicle speed. Therefore, first the rotational speed was converted in to linear speed using the following formula (Eq. 3).

$$Simulated\ vehicle\ speed\ (km/hr) = \frac{2\pi N}{60} \times r \times \frac{1}{1000} \times 3600\text{---------------------------------}(3)$$

where, r is the radius of the wheel, which is 0.055 m. For example, when the rotational speed is 10 rpm the simulated vehicle speed is 0.207339 km/h, which is a very small value. Therefore, it was assumed that 10 rpm (0.207339 km/h) was equivalent to 20 km/h and a multiplying factor could be calculated as 96.46. This factor converted all the rotational speeds to an equivalent real life vehicle speed. Therefore, the final simulated vehicle speed was represented by the Eq. 4.

$$Final\ simulated\ vehicle\ speed\ (km/hr) = \frac{2\pi N}{60} \times r \times \frac{1}{1000} \times 3600 \times 96.46\text{---------------}(4)$$

3.6. Experimentation for Speed Data Collection

In the previous section, it was demonstrated that wheel speed could be measured by the Shimmer sensor using a calibration technique. Therefore, the wheel speed was varied randomly within a certain range, and the speed was measured by the sensor after transferring the vibration data to the server. A connection was established between the sensor and a computer through Bluetooth. The following procedure was adopted for data acquisition from the experimental set-up and to transfer the data to the server:

(i) the wheel was rotated at a random speed

(ii) average vibration acceleration was calculated for each direction in the Cartesian coordinate

(iii) the resultant vibration acceleration was calculated from the average vibration accelerations in each direction and

(iv) the wheel speed was calculated using Eq. 4.

4. SOFTWARE INTERFACE DEVELOPMENT

In this section, first general software architecture has been presented to explain the functioning of the software interface. Finally, details of the algorithm design have been explained.

4.1. Software Architecture

The general concept of the proposed software development is that the speed of a vehicle can be measured using an accelerometer-based wireless sensor (Figure 7). The vibration acceleration data from the sensor are transferred wirelessly to the server computer. The data are then processed in the server to convert them to relevant vehicle speed format and transferred wirelessly to a client computer. The server and the client are two different systems accomplishing different functions. The server provides services to other systems within a network while the client uses remote services provided by the server. The socket is known as one end-point of a two-way communication link between two programs running on a network using server and client [35]. Socket programming has been used for software design based on client/server communication [36]. Cai [37] has developed a network point-to-point chat system based on Java socket and cloud computing technologies in order to improve the communication network chat for safety and efficiency. The server creates a socket object on its end of the communication once the connection is made. The client and server can communicate by writing to and reading from the socket. In socket programming, the socket classes are used for building the connection between a client program and a server program. Two classes are provided by the java.net package, e.g., Socket and ServerSocket. Socket class implements the client side of the connection and while the ServerSocket class provides a mechanism for the server program in order to listen for the client and establish connections with it.

The system has been designed to continuously monitor the speed data through a graphical User Interface (GUI). When the vehicle speed details are displayed in the server computer, at the same time the client computer can read and display the same speed information. A DOS window is created in the client computer where all the speed data are updated and displayed as a chart for monitoring purpose. A graphical user interface (GUI) is also generated in the client computer to display a time-speed graph. If a speed value is greater than a pre-set speed limit, a report is generated to be stored in the client with the details of the vehicle and its speed at that instantaneous time. Therefore, the software system has the capability of managing vehicle speed data automatically and wirelessly. The design of the software architecture between server and client computer has been previously presented in a conference [38].

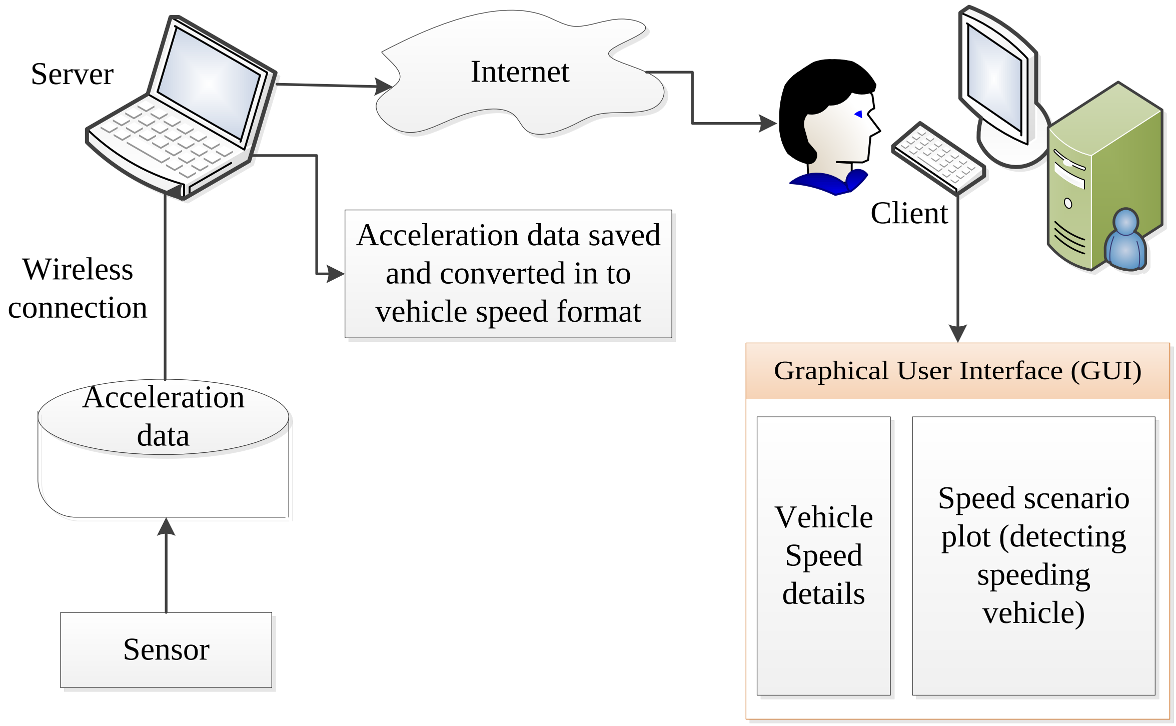

Server
Internet
Client
Acceleration data saved and converted in to vehicle speed format
Wireless connection
Acceleration data
Sensor
Graphical User Interface (GUI)
Vehicle Speed details
Speed scenario plot (detecting speeding vehicle)

Figure 7: Software architecture [33]

4.2. Algorithm Design

The SpeedManage software has been developed using Java socket programming based on two algorithms namely socket server and socket client algorithms. The server computer collects the acceleration data from the wireless sensor and converts the data into vehicle speed format according to the server algorithm as shown in Figure 8. On the other hand, the client computer reads the speed data and displays the status of vehicle speed on a Graphical User Interface (GUI). Furthermore, automated reports are generated and stored for speeding vehicles in the client computer following the socket client algorithm as shown in Figure 8.

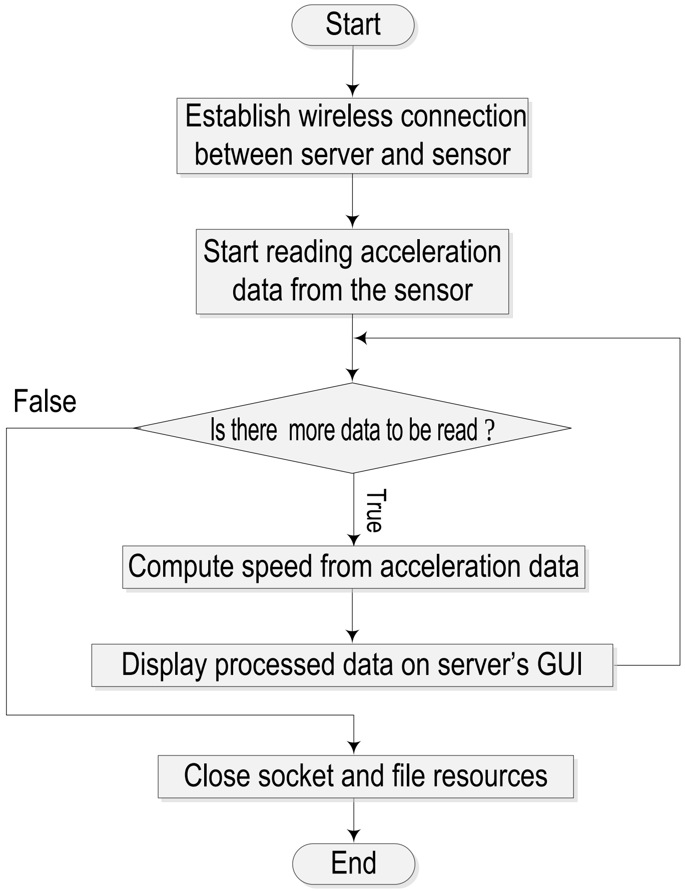

Start
Establish wireless connection between server and sensor
Start reading acceleration data from the sensor
False
Is there more data to be read ?
True
Compute speed from acceleration data
Display processed data on server's GUI
Close socket and file resources
End

Figure 8: Socket Server algorithm for processing sensor data into vehicle speed format

First, a connection between the server and the wireless sensor has been established using the codes of socket programming. As soon as the mechanical wheel starts moving, the vibration acceleration data is transferred to the server computer from the sensor as raw data and is saved as a csv file. Data are transferred to the server computer continuously until the wheel stops. A check is incorporated to determine if there are more data to be read from the sensor. Subsequently, if the data are read continuously, the server algorithm converted the raw data into resultant acceleration, then rotational speed (rpm) and finally into vehicle speed (linear speed) format. A Graphical User Interface (GUI) on the server computer continuously displays the speed data. The individual speed data represents the status of a vehicle speed. As soon as the wheel stops, there are no data to be read in the server computer and the socket and file resources are closed.

The client computer algorithm starts with a connection establishment between the server and the client computers. The server computer communicates with the client computer wirelessly using port selection and local IP address as instructs by a set of Java socket programming codes.

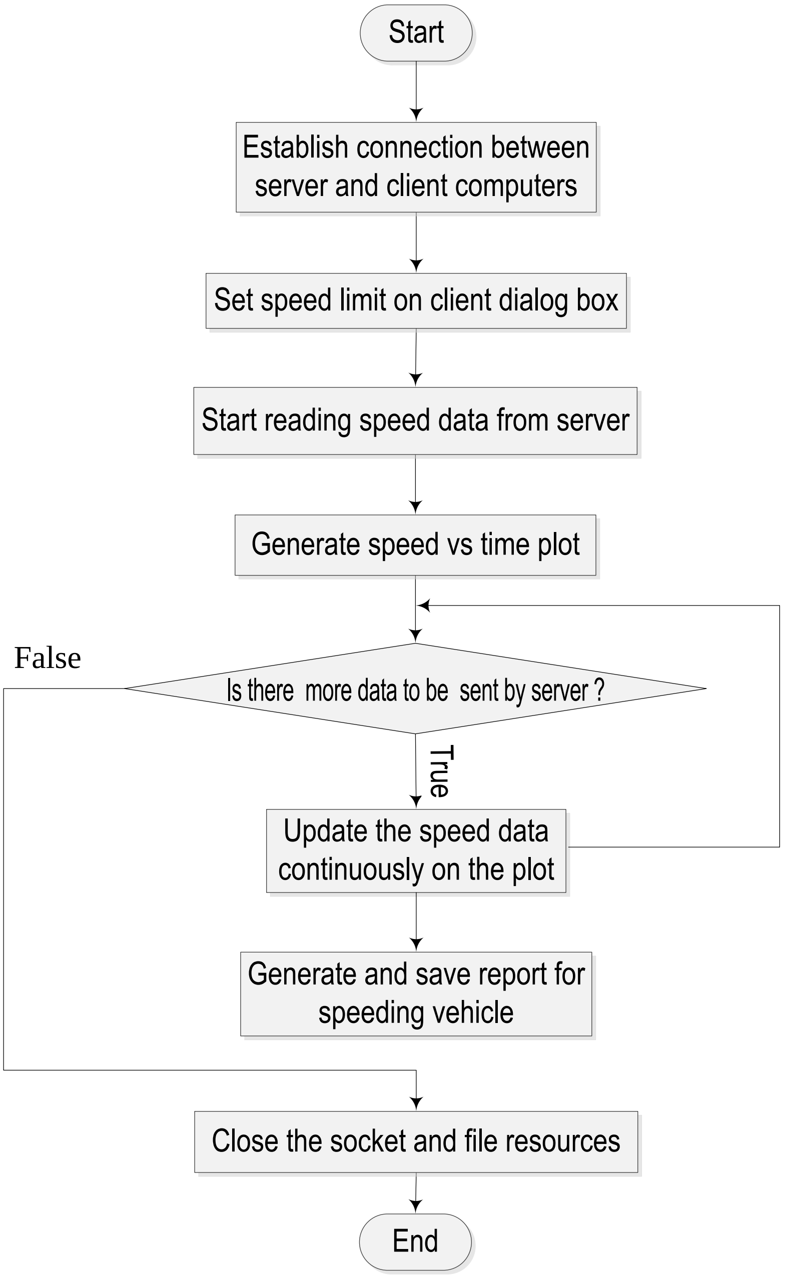

Start
Establish connection between server and client computers
Set speed limit on client dialog box
Start reading speed data from server
Generate speed vs time plot
False
Is there more data to be sent by server ?
True
Update the speed data continuously on the plot
Generate and save report for speeding vehicle
Close the socket and file resources
End

Figure 9: Socket client algorithm for detecting a speeding vehicle and generating speeding report

Once the connection has been established between the server and client computers, a speed limit settings dialog box is appeared in the client computer. The user can set a speed limit in the dialog box based the location of the road (within a city or highway).

After selecting the speed limit, a graphical user interface (SpeedManage) is generated in the client computer to display the vehicle speed status read from the server computer in the form of a time-speed plot. A conditional check is devised in the algorithm to indicate whether any speed data left in the server computer to be read by the client. If a vehicle exceeds a set speed limit, it would be considered as a speeding vehicle on that road. A horizontal red line indicates the maximum speed limit on the plot for a particular road. The speeds below and above the maximum speed limit are represents by different colours for the case of monitoring. The client computer stores the information of a speeding vehicle as csv and pdf file formats with information such as car number, road speed limit, speed of the car, time and date as speeding evidence of a speed offender. After complete reading all data from the server, the socket connection is closed.

4.3. Description of Code Structure

This section describes the programming code structures according to the algorithm designed for the server and client sides.

In the server side, (*CarServer*) class has been assigned. The first step is to declare 'Seversocket' and 'Socket' for the server and the client respectively. Declaration of the 'Seversocket' and 'Socket' follows declaration of a port in the server. The main method *(public static void main(String[] args))* of the server attempts a connection with the client. IP address is used to communicate between the server and the client. A new object (*BufferedOutputStream*) has been created to read the acceleration data for speed calculation in the server side. A class (*JFrame*) of the *javax.swing* library has been used for accepting user input and displaying car speeds by creating a message box. The speed and acceleration are displayed one by one in the message box named as ‘Server’ in the server side.

A main class (*SingleCarClient)* as has been assigned in the client side. An object, (*InputStreamReader*) reads the processed speed data of the server in the client computer. Another object has been created as (*PrintWriter*) which can write the output speed data in a file as ‘output.csv’. A class (*PlotWithArrowLabelsAndNoLines*) has been defined for GUI (SpeedManage) to create an application framework. ‘SpeedManage’ is a GUI that shows speed details of car while it is moving on the road. Different speed limits are set for different road types. The processed speed data are displayed in ‘SpeedManage’ using a method *(public Paint getItemPaint)*. A red line in the GUI represents the threshold value to track the speeding vehicle. According to the conditional relationships, ‘SpeedManage’ displays different colours for the speeds below, equal and above the set speed limit by using another method (*public Color getItemColor*). The speed details with date and time are constantly updated on the left hand side of the GUI with speed status of that vehicle in the right hand side through graphical representation. The following library functions have been used for the GUI. The client computer stores the information of the speeding car such as car number, road speed limit, speed of the car, time and date. An automated report is generated as pdf file format with speeding details and saved in the client computer. The method *(public static void CreatePdf)* has been used to generate the report. The following library functions have been used for the GUI interface.

```
import java.awt.*;
import java.io.*;
import javax.swing.*;
import org.jfree.chart.*;
import org.jfree.chart.axis.*;
import org.jfree.chart.plot.*;
import org.jfree.chart.renderer.xy.XYLineAndShapeRenderer;
import org.jfree.data.*;
import org.jfree.data.xy.*;
import org.jfree.ui.*;
import org.jfree.chart.annotations.XYPointerAnnotation;
```

4.4. Main Features of the Software

The flexibility of the proposed vehicle speed monitoring system shows great advantages in terms of detecting speeding vehicle, speed monitoring, sending and saving the detailed records of the offending vehicle to a central database automatically and instantly without any involvement of human effort. The cost for monitoring vehicle speed and managing the associated data can be significantly reduced using this automated system. The GUI can display the speed information in a user friendly way to detect the speeding vehicle. Therefore, the system can play a vital role in integrating all the activities related to vehicle speed monitoring in an intelligent and efficient way.

5. EXPERIMENTAL EVALUATION OF THE VEHICLE SPEED MONITORING

The concept for vehicle speed monitoring system was simulated in a laboratory environment to test its effectiveness. In this section, first experimental evaluation scenarios was designed and then experimental output was presented in graphical format.

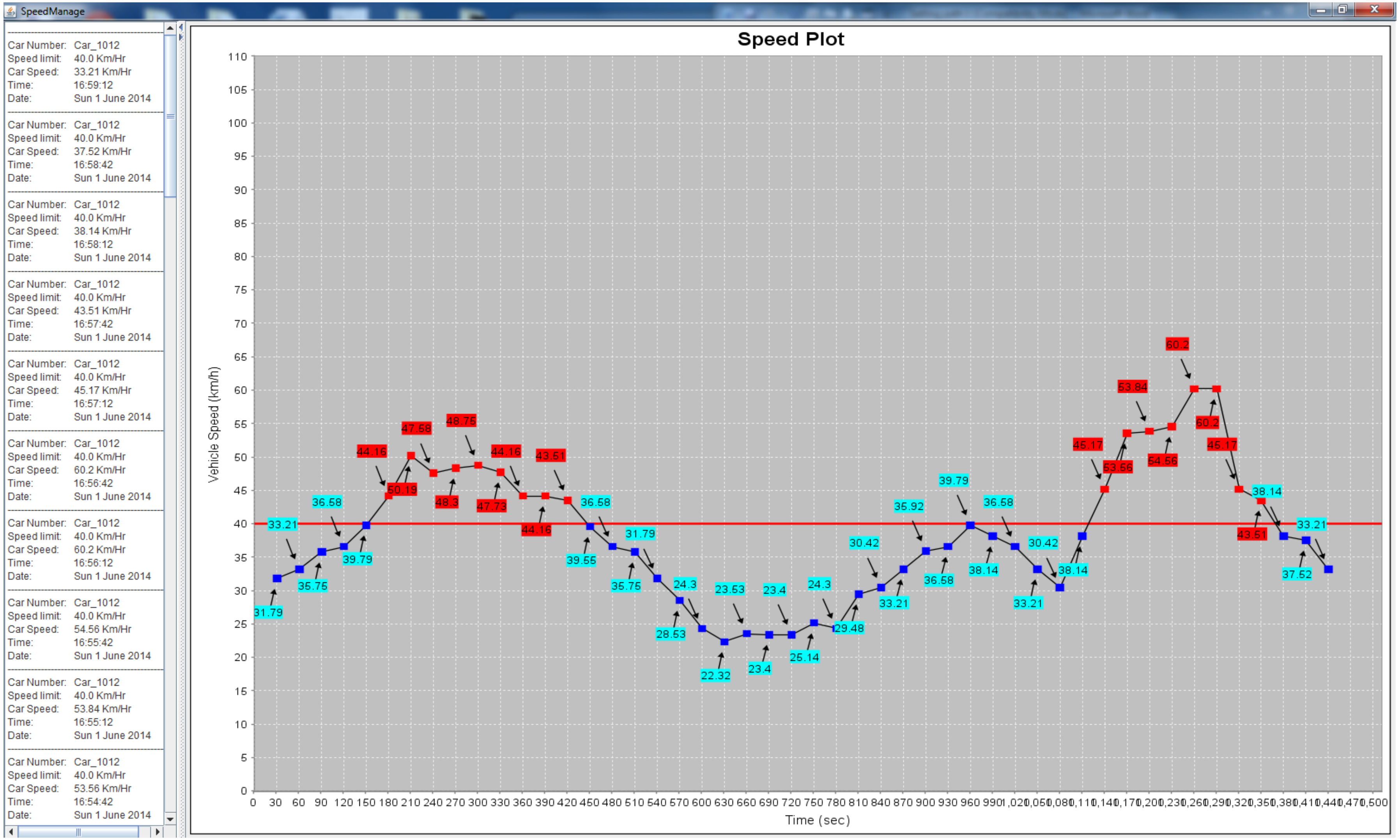
5.1. Experimental Design

Two scenarios were considered for the experimental evaluation: vehicles running on roads inside and outside a city. The first scenario was designed to measure and monitor the vehicle speed inside a city with a speed limit of 40 km/hr, whereas the second scenario was considered for vehicle running on highway with a speed limit 60 km/hr.

5.2 Experimental Results

First, a connection was established between the server and client computers in order to display the vehicle speed information in a graphical user interface (GUI) named as SpeedManage. For experimental evaluation, the speed limit was set to 40 km/hr using the drop down menu in the GUI of client computer and the server and client started communicating between them. Subsequently, the vehicle speed details were displayed in the server computer. At the same time, the GUI in the client computer displayed the same information. Figure 10 displays a snapshot of vehicle speed plot at any particular moment. A red line across the plot indicated the speed limit set for the particular road. The speed plot displayed the speeds after every 30 seconds. Each point in the plot also displayed the actual value of the speed. The plot automatically highlighted the speed situations with different colours. Red colour indicated that the vehicle was running at a higher speed than the set speed limit. On the other hand, a cyan colour indicated that the vehicle was running below the speed limit. Vehicle speed equal to the speed limit was indicated by the amber colour. The speed details were also continuously updated on the left hand side of the GUI. The software would then automatically detect the speeding situation, generate speeding report and save the report in the client computer in pdf or csv file formats. Therefore, the report can be sent to the offender for imposing penalty. A pdf report for a speeding vehicle has been presented in Figure 10. In case of vehicle running outside a city/town, the speed limit can be set to 60 km/hr and the software would follow the same procedure in identifying the speeding situation.

The experimental evaluation results showed that the proposed system can identify each vehicle speed accurately both within and outside the speed limit. Therefore, the software is capable of automatically detecting a speeding vehicle and generating detailed report (Fig. 11) for taking further necessary actions.



Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 33.21 Km/Hr
Time: 16:59:12
Date: Sun 1 June 2014

Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 37.52 Km/Hr
Time: 16:58:42
Date: Sun 1 June 2014

Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 38.14 Km/Hr
Time: 16:58:12
Date: Sun 1 June 2014

Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 43.51 Km/Hr
Time: 16:57:42
Date: Sun 1 June 2014

Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 45.17 Km/Hr
Time: 16:57:12
Date: Sun 1 June 2014

Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 60.2 Km/Hr
Time: 16:56:42
Date: Sun 1 June 2014

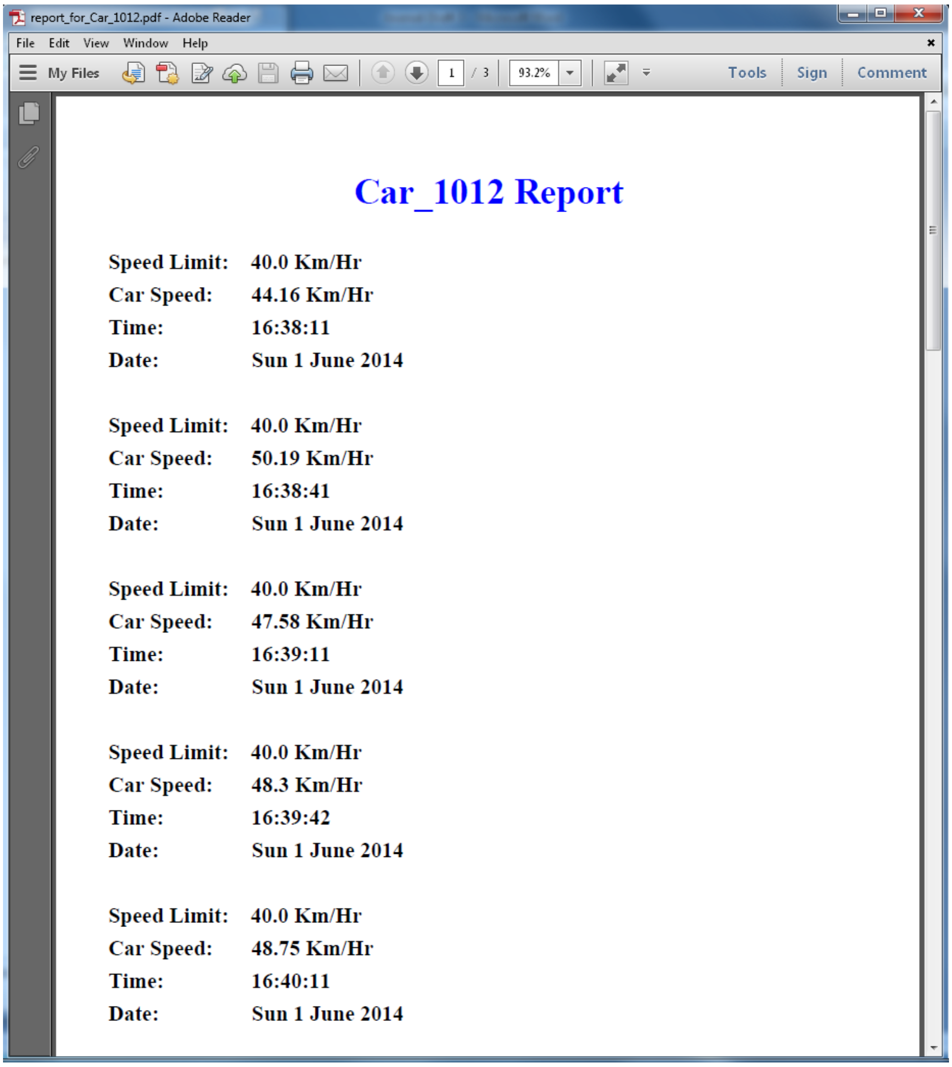
Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 60.2 Km/Hr
Time: 16:56:12
Date: Sun 1 June 2014

Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 54.56 Km/Hr
Time: 16:55:42
Date: Sun 1 June 2014

Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 53.84 Km/Hr
Time: 16:55:12
Date: Sun 1 June 2014

Car Number: Car_1012
Speed limit: 40.0 Km/Hr
Car Speed: 53.56 Km/Hr
Time: 16:54:42
Date: Sun 1 June 2014

Figure 10: Display of vehicle speed plot on a GUI in the client computer for a vehicle running at slower speed limit



Car_1012 Report

Speed Limit: 40.0 Km/Hr
Car Speed: 44.16 Km/Hr
Time: 16:38:11
Date: Sun 1 June 2014

Speed Limit: 40.0 Km/Hr
Car Speed: 50.19 Km/Hr
Time: 16:38:41
Date: Sun 1 June 2014

Speed Limit: 40.0 Km/Hr
Car Speed: 47.58 Km/Hr
Time: 16:39:11
Date: Sun 1 June 2014

Speed Limit: 40.0 Km/Hr
Car Speed: 48.3 Km/Hr
Time: 16:39:42
Date: Sun 1 June 2014

Speed Limit: 40.0 Km/Hr
Car Speed: 48.75 Km/Hr
Time: 16:40:11
Date: Sun 1 June 2014

Figure 11: Report in pdf format for speeding vehicle inside a city/town

6. DISCUSSIONS

The proposed system can be used for measuring vehicle speed and identifying speeding vehicle. A Java based software interface has been developed for communicating with a server and a client and tested for different vehicle speed scenarios. The system can successfully function in collecting and managing speeding data wirelessly through wireless sensor. The system can measure the vehicle speed and display on the GUI in the client side comparing with the speed limit for a particular road. Finally, automated reports are generated for speeding vehicle to identify the speed offenders for taking further necessary action. Therefore, the system meets the requirements of a complete wireless speed monitoring system through speed measurement, wireless data transfer, identifying speeding vehicles and report generation. It should be noted that the system is not an additional feature that would be added to vehicles in the future as part of an Intelligent Transport System. It is envisaged that the prototype system can be implemented for real life traffic management situation as a replacement for the traditional speed monitoring system. Using the wireless system, it is possible to reduce the high cost of traditional systems, to overcome the traditional data collection methods and to collect large quantity of data over time.

The focus in this work was to build a software system that can effectively collect speed data wirelessly and identifying the speeding vehicle to take necessary action. Through experimental evaluation, it has been demonstrated that the developed algorithm and software successfully can integrate these functions. The speed data used in this system were not taken from an actual vehicle, rather they were generated by calibrating speeds of a mechanical wheel with the corresponding vibration. The work can be extended to measure actual vehicle speed by attaching the sensor to the vehicle's wheel. It must be emphasised that the accuracy of the speed measurement must be very accurate possibly within few km/hr in order to impose the speeding penalty accurately.

7. CONCLUSIONS

In this work, a concept for wireless detection, measurement and monitoring of vehicle speed has been developed as a part of Intelligent Transportation System (ITS). An experimental prototype set-up has been developed using a variable speed mechanical wheel, a wireless sensor and computers to simulate the detection and measurement of vehicle speed. Intelligent data management software (SpeedManage) has been developed using Java socket programming to monitor the vehicle speed automatically. The system is capable of detecting a speeding vehicle by comparing with a set speed limit, generating a report in the event of speeding and sending the report to a central database for storing. The report can be used to take necessary action against the speed offenders. The software has been employed for simulating different vehicle speed scenarios and it can successfully function in speeding data collection and management of the same. Based on the performance of the prototype system, it can be concluded that the concept would be suitable for implementing in real-life traffic management situation for effective and efficient management of vehicle speed data.

ACKNOWLEDGMENT

The authors would like to acknowledge the support from the School of Mechanical and Manufacturing Engineering, Dublin City University, Ireland.

REFERENCES

[1] http://www.roadsafetymayo.ie/CausesofAccidents/ (accessed November 2014)

[2] Tubaishat, M., Zhuang, P., Qi, Q. and Shang, Yi., ‘Wireless sensor networks in intelligent transportation systems’, Wireless Communications and Mobile Computing, 2006, vol. 9, no. 3, pp. 287-302

[3] Selvarajah, K., Arief, B., Tully, A., and Blythe, P., ‘Deploying Wireless Sensor Devices in Intelligent Transportation System Applications’, Intelligent Transportation Systems, Dr. Ahmed Abdel-Rahim (Ed.), 2012, InTech

[4] Losilla, F., Garcia-Sanchez, A. J., Garcia-Sanchez, F., Garcia-Haro, J., & Haas, Z. J. ‘A comprehensive approach to WSN-based ITS applications: A survey,’ 2011, vol. 11, no. 11, pp. 10220-10265

[5] Pascale, A; Nicoli, M.; Deflorio, F.; Dalla Chiara, B.; Spagnolini, U., ‘Wireless sensor networks for traffic management and road safety,’ IET Intelligent Transport Systems, 2012, vol. 6, no. 1, pp. 67-77

[6] Khanafer, M., Guennoun, M., & Mouftah, H. T., WSN architectures for intelligent transportation systems. In New Technologies, Mobility and Security (NTMS), IEEE 3rd International Conference, Cairo, Egypt, 2009, pp. 1-8

[7] Mohan, P., Padmanabhan, V. N., and Ramjee, R., ‘TrafficSense: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones, Microsoft Research’, Technical Report No. MSR-TR-2008-59, 2008, pages 29

[8] Coleri, S., S.Y. Cheung, P. varaiya, ‘Sensor Networks for Monitoring Traffic, 42nd Annual ALLERTON Conference on Communication, Control, and Computing’, Invited paper, September 2004.

[9] Sensys Networks, Inc. ‘Sensys™ Wireless Vehicle Detection System’. Reference Guide P/N 152-240-001-001 Rev D., 2010

[10] Hisao, O. and Takashi, A., ‘Vehicle detecting device and traffic quantity measuring equipment’. 1998, Japanese Pat. No 2729977(B2)

[11] Ma, W., D. Xing, A. McKee, R. Bajwa, C. Flores, B. Fuller, and P. Varaiya, ‘A wireless accelerometer-based automatic vehicle classification prototype system’, IEEE Transaction on Intelligent Transportation Systems, 2014, vol. 15, no.1, pp.104-111,

[12] Matsuo, T., Kaneko, Y. and Matano, M., ‘Introduction of intelligent vehicle detection sensors’, Proceedings IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, Japan, 1999, pp. 709-713

[13] Bachmann, C., Abdulhai, B., Roorda, M.J. and Moshiri., B. ‘A comparative assessment of multi-sensor data fusion techniques for freeway traffic speed estimation using microsimulation modeling’. Transportation Research Part C: Emerging Technologies, 2013, vol. 26, pp. 33–48

[14] Chen, S., Sun, Z.P. and Bridge, B. ‘Automatic traffic monitoring by intelligent sound detection’. IEEE Conference on Intelligent Transportation System, Boston: MA, 9-12 Nov 1997, pp. 171 - 176

[15] Luciani, S. ‘Traffic monitoring system and method’. 2003, US Pat. No. 6,505,114B2

[16] Murty, R.N., Mainland, G., Rose, I., Chowdhury, A.R., Gosain, A., Bers, J., and Welsh, M. CitySense: ‘An urban-scale wireless sensor network and testbed’. In: Proceedings of the IEEE Conference on Technologies for Homeland Security, Waltham: MA 12-13 May 2008, pp. 583–588

[17] Zhang, L., Wang, R., and Cui, L. ‘Real-time traffic monitoring with magnetic sensor networks’. Journal of Information Science & Engineering, 2011, Vol. 27, No. 4, pp. 1473-1486

[18] Karthikeyan, B., and Tamileniyan, M. ‘Dynamic Data update for Intelligent Speed Adaptation (ISA) System’, International Journal of Computer Applications, 2010, Vol. 11, No.1, pp. 8-13

[19] Herrera, J. C., Work, D. B., Herring, R., Ban, X., Jacobson, Q. and Bayen, A. M. ‘Evaluation of traffic data obtained via GPS-enabled mobile phones,’ The Mobile Century field experiment, 2010, vol. 18, no. 4, pp. 568-583

[20] Endo, S., Ukawa, H., Sanda, K. and Kitagawa, A. ‘Simulation of speed control in acceleration mode of a heavy-duty vehicle’, JSAE review , 1999, Vol. 20, No. 1, pp. 81-86

[21] Agerholm, N., Waagepetersen, R., Tradisauskas, N. and Lahrmann, H. Intelligent Speed Adaptation in Company Vehicles, Intelligent Vehicles Symposium, 4-6 June, Aalborg, 2008, pp. 936-943

[22] Boyraz, P., and Hansen, J.H.L. ‘Active accident avoidance case study: Integrating drowsiness monitoring system with lateral control and speed regulation in

passenger vehicles', IEEE International Conference on Vehicular Electronics and Safety, ICVES, Columbus, OH, 2008, pp. 293 – 298

[23] Oche, M., Noo, R.M., AI-jawfi, A. S., Bimba, A. T. and Nasir, M.K. 'An Automatic Speed Violation Detection Framework For VANETs', IEEE International Conference on RFID Technologies and Applications, Johor Bahru, 2013, pp. 1 – 6

[24] Sarbini, M.A.M., Hassan, S.B., Jiann, T.S. and Ahmad, P.M.N.P.H. 'Design of a RFID-based speed monitoring system for road vehicles in Brunei Darussalam', International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, 2014, pp. 219 - 223

[25] Shinde, P.A., Mane, Y.B. and Tarange, P.H. 'Real Time Vehicle Monitoring and Tracking System based on Embedded Linux Board and Android Application', International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2015, pp.1-7

[26] Zengqiang, M., Guosheng, G., Wanmin, S. and Yan, Y. 'Wireless monitoring system of vehicle overspeed on freeway based on GPRS', 27th Chinese Control Conference, Kunming,Yunnan, China, 2008, pp. 550 - 553

[27] Roberts, Jr., E.V. 'Traffic monitoring system and method'. 2002, US Pat. No. 6384739B1

[28] Zhou, J., Gao, D. and Zhang, D. 'Moving vehicle detection for automatic traffic monitoring', IEEE Transactions on Vehicular Technology, 2007, Vol. 56, No. 1, pp. 51-59

[29] Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., and Madden, S. 'CarTel: A distributed mobile sensor computing system', Proceedings of the 4th international conference on Embedded networked sensor systems, 1–3 November, 2006, Boulder, Colorado: USA, pp. 125-138

[30] Nadeem, T., Dashtinezhad, S., Liao, C. and Iftode, L. 'TrafficView: a scalable traffic monitoring system', Proceedings of the IEEE International Conference on Mobile Data Management, 2004, pp. 13–26

[31] Chen, X. Zhang, J. Qian, S. and Xu, P. 'Applied Research on Traffic Information Collection Based on Wireless Sensor Networks'. Energy Procedia, 2012, Vol. 17, Part A, pp. 602-606

[32] Shimmer Research, www.shimmerresearch.com (accessed November 2014)

[33] Shimmer Research, www.shimmerresearch.com (accessed December 2013)

[34] Török, J. 'Analytical Mechanics: With Introduction to Dynamical Systems', John Wiley & Sons, Inc.: USA, 2000.

[35] Harold, E. R. Java Network Programming, 3rd Edition, O'Reilly Media, Inc. 2005.

[36] Xue, M. and Zhu, C. 'The Socket Programming and Software Design for Communication Based on Client/Server', Pacific-Asia Conference on Circuits, Communications and Systems, Chengdu, 2009, pp. 775 - 777

[37] Cai, M. 'The Design Method of Network Chat System Based on Socket and Cloud Computing', International Conference on Computer Science and Service System, Nanjing, China, 2012, pp. 610 – 613

[38] Ahsan, M., McManis, J. and Hashmi, M.S.J. 'Prototype System Development for Wireless Vehicle Speed Monitoring', 9th IEEE International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP), Manchester, UK, 2014, pp. 287 - 292