JOURNAL OF ENGINEERING AND EARTH SCIENCES
JEES ISSN: 1897-8680
JEES 2021
VOLUME 14 ISSUE NO 1
PAGE 98-112

DESIGN AND IMPLEMENTATION OF VEHICLE SPEED DETECTION USING GAM OPTIMISATION TECHNIQUES

[1]Omotoso. G. S., [2]Oladimeji, .T. T., & [3]Idowu, O. D.
[1,2,3]Department of Electrical/Electronic Engineering, Federal Polytechnic, Ado-Ekiti, Nigeria.
[1]gbengaomotoso50@gmail.com, [2]tolulopett@yahoo.com and [3]opeyemi.idowu@gmail.com

ABSTRACT

This research presents a design method of a Model Predictive Control (MPC) with low computational cost for a practical Adaptive Cruise Control (ACC) running on an in-built microprocessor. The major drawback of the previous ACC is its low speed response in traffic jams, in which stop-and-go driving is required. ACC with a stop-and-go function has been widely commercialized. It enables drivers to be free from driving stress in traffic jams and has the potential to enhance safety driving. However, during the traffic jam, conventional ACCs have a slower response time than normal drivers' traffic flow. The response delay is caused by using general control methods designed based on a linear time invariant model, and not making full use of output characteristics of an engine and brake actuators. Using such ACC, other cars cutting in front of own car from adjacent lane tend to be increased due to the response delay. As a result, traffic flow will be disturbed, and the risk of traffic accident will increase. To improve the control performance, it is important to design a controller considering vehicle characteristics which significantly changes depending on driving conditions. In this research, we attempt to solve the problem by using MPC that can explicitly handle constraints imposed on, e.g., actuator or acceleration response. Furthermore, we focus on decreasing the computational load for the practical use of MPC by using low-order prediction model. From these results, we developed ACC with high responsiveness and less discomfort even for traffic jam scene.

KEYWORDS: Adaptive cruise control; Generalized additive model; Model predictive control; Optimization

INTRODUCTION

Road accidents have been very prevalent in today's globe, the main cause of which being careless driving. It was very important to monitor this, and various techniques have been applied up until now. However, with the development of technology, various governing bodies are demanding some kind of computerized technology to regulate the problem of speed driving. Nigeria and many nations around the globe have experienced a rise in road accidents, much of which can be ascribed to human error, such as over-speeding. The vehicle speed controller mentioned in this research is intended to automatically regulate the speed of the vehicle so that the speed of the vehicle does not exceed the speed limit set for a specific area by a regulatory body such as the Federal Road Safety Commission (FRSC) in Nigeria (Thomas et al., 2014). In order to detect vehicles speed, a radar detector was used.

Among the different techniques for speed detection, it seemed that radar detection is the most precise way for the detection of vehicle speed. It is used extensively in the commercial speed guns used by the Traffic Police to check speed vehicles. The radar sensor has an oscillator and mixer integrated. The signal hits the moving car and reflects the sensor with the Doppler principle changing frequency. (Akihiko et al., 2016). The mixer produces a signal that is the distinction between the signal transmitted and the signal received, and the output signal is proportional to the speed of the car.

The vehicle speed controller works by taking inputs from a set of three switches that can set speed limits of 40, 80 and 120 kilometers per hour, respectively. Each switch environment is compared to the actual speed of the vehicle obtained from the speedometer of the vehicle. If the speed of the vehicle exceeds the setting, the actuator shall be called into service in such a way that the speed of the vehicle is not permitted to exceed the specified threshold for the area. The microcontroller was programmed to take inputs from the switches and the speedometer; depending on the settings, the microcontroller switches on the Light Emitting Diodes LEDs) and displays the appropriate messages on the LCD screen. In the simulation model, the stepper motor was governed by the microcontroller via the motor driver to represent the speed adjustment in the actual setting. The program for the microcontroller was written using the micro C development environment and the hardware simulation was carried out with the help of Proteus Design Suite Version 8.0. The speed settings for the area can be changed to suit the decision of the regulatory agency. Installation of the vehicle speed controller will not only offer drivers early warnings, but will prevent over-speeding leading to a decrease in the number of road accidents.

Problem Statement

Vehicle speeding is a significant issue for road safety and needs to be properly addressed in order to minimize accidents. Excessive Speed is a factor in one third of all deadly accidents. Vehicle speed detection is based on the use of Doppler radar to detect the speed of moving cars. The most trending techniques that prompt development of this project. Most old vehicle does not have vehicle speed device, cost of installation of latest technology by the government and no embedded dashboard in case any feature damage. Practical Adaptive Cruise Control (ACC) running on an embedded microprocessor was used to solve all these aforementioned problem but give a very low response. In order to reduce the computational cost, integrate working of dashboard to improve efficiency of the vehicle and reduce road accidents. Model Predictive Control (MPC) of generalized additive model was hereby used.

Aim and Objectives

The aim of this paper is to develop a cost-effective device for monitoring vehicle status and safety on the highway. The objective includes;

i. Model formulation using Linear programming
ii. To use generalized additive model to implement (i)
iii. To evaluate result based on the vehicle speed detection

Review of related works

Nigeria and the wider world are losing a big number of its people every year in the case of road accidents (Omidiji & Ibitoye 2010). The 2013 research by the Federal Road Safety Commission (FRSC) of Nigeria shows that as many as 32% of all accidents that occurred that year were attributed to speed-related issues (Federal Road Safety Corps 2013 Over the last three decades, distinct types of technology have been used to regulate the velocity of vehicles in order to reduce fatal accidents. Akihiko et al., (2014) patented an automatic car speed controller, the driving circuit of which operates a difference between the actual speed of the car and the speed of the memorized car; the difference signal is used to operate an electromagnetic clutch. A similar system has also been established in Akihiko et al., (2015). The systems referred to in Akihiko et al., (2014) and Akihiko et al., (2015) have been rendered obsolete by newer and improved car manufacturing methods. Santoshi et al., (2015), Eswaramoorthy & Araunkumar (2014) and Saivignesh et al., (2015) report more modern electro-hydraulic braking systems and electronic throttle control systems.

Byum, et al. (2016) used the Unified Modeling Language (UML) notation to theoretically model the car speed regulator. Byum (2016) discusses an electronic vehicle speed control system using two variable resistors for the production of velocity control signals, which are processed and converted into a digital signal for the monitoring of the driving speed of an electric car. Carsten and Tate (2017) reports on a system that uses Radio Frequency Identification (RFID) to detect highway traffic signals as well as vehicle communication infrastructure. This advanced scheme not only controls the speed of the vehicle and shows its position using the Global Positioning System (GPS) but is also capable of stopping road infrastructure accidents. There are many sensors and electronics used in this system, which makes the design costly. Other systems based on pairs of RFID and RF transmitters are registered in Madhu et al. (2014), Thomas et al. (2014) and Kameswari et al. (2011). These schemes require the installation of a big road infrastructure in the form of highway beacons.

The most modern autonomous system that uses GPS for car location and velocity control is the Intelligent Speed Adaptation (ISA) (Paine, 2017; Carsten & Tate, 2017; Paine et al., 2015). This system uses the GPS in the vehicle to determine its location. The memorized digital map included in the system enables the speed limit of the location to be determined so that the driver can be advised to stay within the appropriate speed limit. This complex system will certainly require the accessibility of the GPS satellite scheme, the thorough development of digital maps of all roads in the country where it is deployed, as well as the accessibility and maintenance of outstanding roads. These factors may not always be present in developing countries.

The system described in this work is strictly advisory in that it does not take any active action to brake or throttle the car; however, this component is efficiently simulated in the role of the engine. The system uses visual indicators, i.e. light emitting diodes (LEDs) and writes text on the liquid crystal display screen (LCD) to alert the driver to the need to remain within the prescribed speed limits. The audible alarm will also be triggered if the driver exceeds the speed limit. The system described in this work is simple to implement and is also much cheaper than the other systems listed in the literature review.

MATERIALS AND METHOD

In the development of generalized linear models, we use the link function g to relate the conditional mean $\mu(x)$ to the linear predictor $\eta(x)$. But really nothing in what we were doing required η to be linear in x. In particular, it all works perfectly well if η is an additive function of x. We form the effective responses z_i as before, and the weights w_i , but now instead of doing a linear regression on xi we do an additive regression, using back fitting . This gives us a Generalized Additive Model (GAM).

Model Formulation

Consider a vehicle and driver whose processes are described by Paine (2017). Driver in the moving vehicle generates request through the accelerator to increase speed.

VM on compute cloud gets data from storage cloud

VM on compute cloud aggregates information for driver.

VM on compute cloud responds with information to driver through Generalized Additive Model.

From the processes described, three parameters are identified. These are

i) Vehicle speed.
ii) Accelerometer information
iii) Transfer of speed status from the driver to the monitoring unit.

Based on that, the problem is modeled in terms of Linear Programming problem.

A Cloud provider has physical machine that may hold instances of virtual machines and storage.

The input sets for the formulation of the problem are as follows:

- I = A set of physical machines.
- J = A set of virtual machines in physical machine.
- N = A set of barriers in determination of vehicle speed.
- K = A set of web objects or content

The parameters that describe a virtual machine are as follows.

- c_j = cost of virtual machine j, for each j in J
- v_j = number of virtual machine instances
- t_j = number of hours used by virtual machine

The parameters that describe a web object are as follows.

- s_k = size of object k
- $\sum s_k$ = aggregate size of all objects k
- r_k = request rate for object k by client n
- δ_k = unit delay for transferring object k
- $\Delta_k = \delta_k \sum r_k$ Aggregate delay for all k
- $\lambda_k = s_k / \delta_k$ Throughput for transferring vehicle speed
- b_k = bandwidth for transferring vehicle speed
- $\beta = \sum b_k$
- c_k = cost of outbound transfer of speed detection k
- $c_t = s_k . r_k . c_k$ Cost of transferring object k from storage to client n.

Parameters that describe storage

- c_s = cost of storage/unit
- S_{cap} = storage capacity

QoS metrics considered in the SLA t = throughput as stated in the SLA

- b = bandwidth as stated in the SLA
- d = delay as stated in the SLA

X_{ij} = 0 or 1 defines the decision variable representing the virtual machine j in physical machine i responding to request from client n in N. The decision variable takes the value 1 if the vm erves the request otherwise the value is 0.

The cost of transferring a single vehicle speed k includes the cost of vm instance, cost of transferring the speed from driver to monitoring unit n, and the cost of storage. This can be described as: $c_j + s_k.r_k.c_k + s_c$. The objective function therefore represents the total cost of transferring multiple vehicle speed from driver to monitoring unit $_n$ N. It is defined as:

$$Z = \min \left(\sum \sum c_j X_{ij} + \sum \sum \sum s_k r_k c_k X_{ij} + s_c \sum s_k\right)$$

Subject to:

$$\sum b_k < b$$
$$\lambda_k \geq t$$
$$\Delta_k \geq d$$
$$\sum s_k \leq S_{cap}$$
$$X_{ij} = 0 \text{ or } 1$$

Generalized additive model

Variation in speed detection using Doppler Effect

If c be the velocity of light, the target velocity then the shifted frequency (f_r) can be expressed as a function of the original frequency (f_t) (Linscott, 2014)

$$f_r = f_t\left(\frac{1+v/c}{1-v/c}\right)$$

Assuming the direction of the source and the direction of motion of the target is perfectly aligned, change in the frequency between the transmitted and reflected wave, (Doppler frequency) (f_d), is thus:

$$f_d = f_r - f_t = 2v\frac{f_t}{(\mathrm{c}-\mathrm{v})}$$

Since for most practical applications of radar, $v \ll c$, so $(\mathrm{c}-\mathrm{v}) \to \mathrm{c}$ We can then write:

$$\mathrm{f}_d \approx 2v\frac{f_t}{c}$$

If the direction of the moving object makes a certain angle with the source direction, Doppler frequency is reduced by a value equal to the cosine of the angle between source and target's direction.

Algorithms for Frequency Variation of signals

Frequency variation for speed limit is divided into two stages

Algorithm for measuring frequency

1) Enable external interrupt and check for rising edge in the input signal
2) At rising edge, start counter
3) Wait for next rising edge
4) At next rising edge, stop counter
5) Divide clock frequency of counter by output of counter
6) Repeat steps 1-5 for a number of times and take average

The microcontroller is run with a 16 MHz crystal. So if the counter inside it counts 'n' between two successive rising edges of the input signal then the frequency of the input signal is 16/n MHz. This approach of measuring frequency is accurate when the frequency of the input signal is very small compared to the clock of microcontroller.

Optimization with generalized addictive model

The set-up for an arbitrary GLM is a generalization of that for logistic regression. We need *A* linear predictor , $\eta(\mathrm{x})=\beta_o + xc\beta$

- A link function g , so that $\eta(\mathrm{x})=\mathrm{g}(\mathrm{r}(\mathrm{x}))$ For logistic regression, we had $g(r) = \log r/(1-r)$
- A dispersion scale function V , so that $\mathrm{Var}[\mathrm{Y}/\mathrm{X}=\mathrm{x}] = \sigma^2 V(\mathrm{r}(\mathrm{x}))$ For logistic regression, we had $V(r) = r(1-r)$ and $\sigma^2 = 1$

With these, we know the conditional mean and conditional variance of the response for each value of the input variables x. As for estimation, basically everything in the IRWLS set up carries over unchanged. In fact, we can go through this algorithm:

1. get the data (x_1, y_1),,,,,,,,,,,(x_n, y_n), fix link function g® and dispersion scale function®, and make some initial guesses $\beta_o \beta$
2. until β_o, β converge

a. calculate $\eta(x_i) = \beta_o + x_i.\beta$ and the corresponding $r(x_i)$

b. find the effective transformed responses $z_i = \eta(x_i) + \frac{y_i - r(x_i)}{g}(r(\mathrm{x}_i))$

c. Calculate the weights $w_i = [(g^{`}(r(x_i))^2 V(r(x_i))]^{-1}$

d. Do a weighted linear regression of z_i on x_i with weights w_i and set $\beta_o \beta$ to the intercept and slopes of this regression

Pinout diagram

Description

- 5.5 Inch LCD display is portable and easy for drivers to read data
- Detect all important driving information via the OBD 2 system
- KM and MPH switch freely, Normal display mode, high-speed display mode, automatically display mode are available
- Display vehicle speed, engine speed, water temperature, fuel consumption, mileage, low fuel hint, engine failure
- Auto power on and off with vehicle started and shutdown

Features

- ❖ Plug and play, HUD can adapt to vehicle type automatically, which in line with OBDII or EUOBD
- ❖ Comprehensive display, HUD can display many parameters at the same time, as follows, vehicle speed, engine speed, water temperature(voltage, throttle valve position, ignition advance angle, time of 100 km acceleration), fuel consumption, mileage, low fuel hint, engine failure;
- ❖ Suitable size. The display area is 5.5 inches which is very clear for driver to read data.
- ❖ Automatic power on and off. Starting with the vehicle starting, stopping with the vehicle stopping to protect the vehicle’s attery, Moreover, we operate the hand switch function to control the HUD.
- ❖ Engine speed alarm reminder, facilitating timely shift and saving fuel consumption;
- ❖ Overspeed single-stage and four-stage alarm modes for choose, more conductive to safety driving;
- ❖ Automatic and manual brightness adjustment, not dazzling

RESULTS AND DISCUSSION

Result evaluation for model formulation

MATLAB®, Simulink® and Model Predictive Control Toolbox™ were used in the simulation. The optimization problem is solved by a QP solver, based on KWIK algorithm, in Model Predictive Control Toolbox. We used “Adaptive MPC Controller” block provided by the toolbox to model the controller and run the closed-loop simulation with the plant model in Simulink. C-code can be generated from the MPC block for the implementation to an embedded microprocessor. Figure 4.1 and Figure 4.2 illustrates the simulation result of MPC and LQR. Each figure shows, from above, (a) inter-vehicle distance following error $y1$, (b) relative velocity $y2$ and (c) preceding vehicle’s acceleration $\dot{v}$p host vehicle’s acceleration $y3$ and control input u. As can be seen in Figure 4.1, the MPC achieves that all the plant output yi (i = 1, 2, 3) converges to zero while those input constraints are satisfied. To follow the preceding vehicle fast, the MPC tries to make full use of the actuator capability. On the other hand, as shown in Figure 4.2, the LQR preforms slower response compared with the MPC. The weight of the LQR is tuned on a trial and error basis to meet the input constraints for this specific scenario. The response is conservative due to the small feedback gain to reduce overall magnitude of the control input by using such linear controller. The advantage of MPC for ACC is that we can realize high control performance since such constraints can be explicitly dealt with and the tuned weights can be intuitively and flexibly designed as the function of time or if-then rule, considering various driving situations.

Results of the experimentation are shown in Figure 4.2(MPC) and Figure 4.3 (normal driver). From comparing both results at time=21–26[s] which is the acceleration scene during traffic jam, we were able to obtain the equivalent response delay (approx. 1.5s) as a normal driver. Also, it can be seen that sudden acceleration/deceleration is suppressed in the high speed driving scene in Figure 4.4. This is because the weight for the change in the control input is functioned effectively. From the above, it is understood that the response is good even in the traffic jam scene, and the controller is functioning with less discomfort in the high speed.

Variation in speed detection using Doppler effect

The magnitude of the Doppler Shift is proportional to reflection of transmitted energy and is in the range of few Millivolts. A high gain low frequency amplifier is usually connected to the IF terminal in order to amplify the Doppler shift to a process able level. Frequency of Doppler shift is proportional to velocity of motion. The Received Signal Strength (RSS) is the voltage measured of the Doppler shift at the IF output. It can be used to predict the distance between the source and the target. Target closer to the source produce a high level of RSS. But the exact value may vary greatly depending on the type of the surface and reflection coefficient of the target. Noises are of major concern when designing an amplifier. Noise may be the result of the internal circuitry of the sensor and amplifier or due to fluctuation in the power supply. High frequency noise from power supply may affect the output significantly.

Frequency variation of signals

The accuracy of the system in frequency measurement was tested using a function generator and an oscilloscope. The input to the amplifier was connected to the output of the function generator and an oscilloscope was connected for reference. Small sinusoidal signal was applied to the circuit and the frequency varied. The table below summarizes the result obtained from the test.

The test showed very high degree of accuracy in measurement of frequency of input signal. However, in actual system with sensor as source of Frequency Noises in various forms are present. So a slightly higher degree of inaccuracy than the data shown in above table can be expected.

Measuring frequency

If 'f' be the value of frequency measured by the microcontroller, then the corresponding speed in km/hr is obtained by dividing it with a factor 19.49. Some of the values of frequency and their corresponding speed are presented in the table below.

Optimization with Generalized Addictive Model.

Testing accuracy of the sensor and the overall system in speed measurement required a standard calibrated device such as a speed gun for determining speed of moving vehicles. But due to lack of such resources available for testing, the accuracy of system could not be exactly determined. Attempts were made to test the system output against the speedometer of vehicle and despite the complexity in observation some data were produced to check the accuracy of system.

The accuracy of the system was observer to be very low. This may be due to a number of factors:

- ✓ The direction between the motorbike and the sensor was not perfectly aligned (i.e. angle A not equal to 0).
- ✓ The table is based on the observation made by the motor bike rider and the observer
- ✓ looking at the LCD display
- ✓ It was hard to synchronize the readings taken by two observers.
- ✓ Furthermore the readings from the analog spectrometer of the bike could not be observed precisely.
- ✓ The output was affected by number of objects moving in the surrounding.

CONCLUSIONS AND RECOMMENDATIONS

In this research, we proposed a method of designing a practical MPC suitable for embedded processors corresponding to various traffic situations. Computer simulation showed that the MPC (GAM optimization techniques) is superior to conventional controllers because it treats actuator constraints and has higher response as ACC. Experimental verification results showed that the proposed MPC controller can be implemented in embedded microprocessors and can achieve high responsiveness and less discomfort. Considering the usefulness of a system, people tend to use or not use an application to the extent they believe it will help them perform their job better and performance gains are often dependent upon the users' level of willingness to accept and use the system. The automatic car speed controller system has been designed and implemented using a microcontroller.

REFERENCES

Akihiko, T., Iwaoka, T., Yamaguchi, Y. & Danzaki, T. (2015). *Automatic Car Speed Controller.* United States Patent Number 5107947. [Online] Available:http://www.freepatentsonline.com/5107947.

Akihiko, T., Iwaoka, T., Yamaguchi, Y. & Danzaki, T. (2016). *Automatic Car Speed Controller with Detection of Abnormality of the Electromagnetic Clutch*. United States Patent Number 5014201.[Online]Available:http://www.patents.com/us-5014201.html.

Byum, L.S. (2016). *Speed Control Apparatus and Method for Electric Car.* United States Patent Number 5726544. [Online] Available: http://www.patents.com/us-5726544.html.

Carsten, O., & Tate, F. (2017). Intelligent speed adaptation: The best collision avoidance system. *Proceedings of the 17 Conference on the Enhanced Safety of Vehicles, Netherlands*. [Online]Available:http://tinyurl.com/26cfod .

Eswaramoorthy, P., & Arunkumar, M. (2014). Intelligent vehicle control based on identification of road signs by solar powered RFID transponders. *International Journal of Research in Engineering and Technology, 3*, 85-89.

.

Federal Road Safety Corps (FRSC). (2013). *Road crash statistics*. Retrieved from http://frsc.gov.ng/2013%FRSC%20Annual%20Report.pdf

Kameswari, U.J., Satwik, M., Lokesh, A. & Reddy, G.V. (2011). A design model for automatic vehicle speed controller. *International Journal of Computer Applications, 35*, 19-24.

Linscott, R. (2014). *Automotive speed indicator.* Retrieved from http://www.circuitdb.com/?p=398

Madhu, Adhaulya, R., Singh, S., Sambhavi, & Chauhan, A. (2014). An advanced robotic car for accidental preventions. *International Journal of Engineering Research and Management Technology,* 1, 263-267.

Omidiji, A. A., & Ibitoye, S. A. (2010). Crime and road crashes prevention in public transportation system in Nigeria: The case study of Kwara, Kogi and Ekiti States. *24th ARRB Conference- Building on 50 years of Road and Transport Research, Melbourne, Australia.*

Paine M. (2017). *Comments on draft protocol for speed-limitation devices*.Retrieved from http://www.mpainesyd.com/idisk/Public/ANCAP%20speed%20limiter%20comments%20mar08.pdf

Paine, M., Paine, D., Griffiths, M. & Germanos, G. (2017). In-vehicle intelligent speed advisory systems. Retrieved from http://www-nrd.nhsta.dot.gov/pdf/nrd-01/esv/esv20/07-0247-w-pdf

Saivignesh, H., Mohamed, S. M., Nagaraj, M., Sharmila, B., & Nagaraja, P. M. (2015). RF based automatic vehicle speed limiter by controlling throttle valve. *International Journal of Innovative Research in Science, Engineering and Technology, 4*, 18722-18728.

Santoshi, G.V.L., Hima, B.N. & Uma, R.K. (2015). Intelligent vehicle control based on identification of road and traffic signal operated rfid transponders. International Journal of Engineering Research, 40-45.

Thomas, L., James, S. A., Joseph, S., Arya, K. B., Narah, T. & Pangu, O. (2014). Automatic speed control of vehicles using RFID. *International Journal of Engineering and Innovative Technology, 3,*118-120

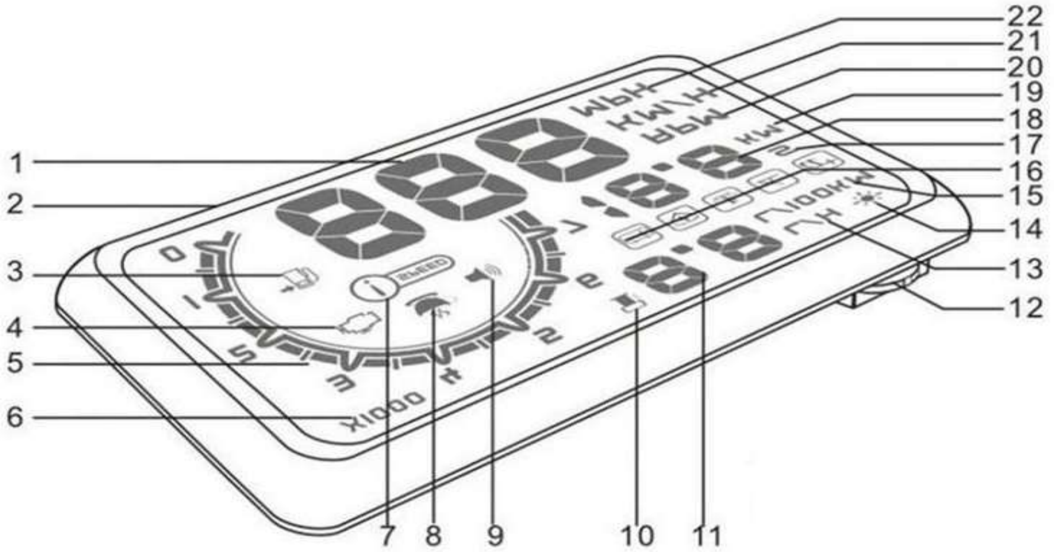
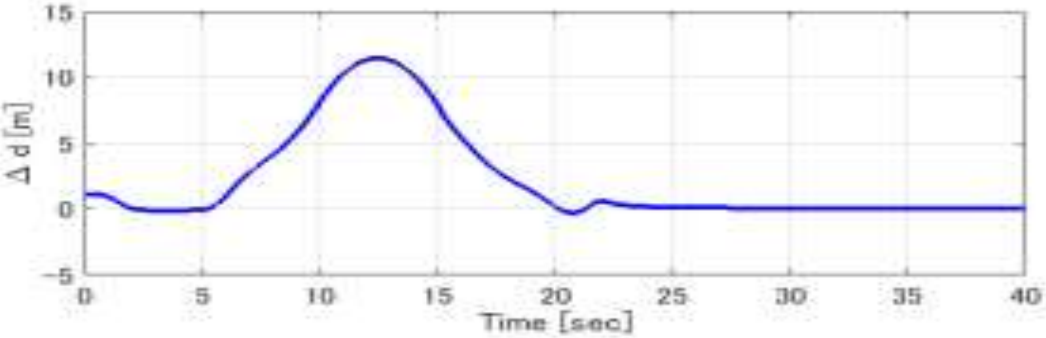


Figure 1 : Pinout diagram of Speed Limit

1-Date 2-Date Interface 3-Refuel 4-Engine failure 5-Engine speed
6-Engine icon 7-Speed alarm 8-Rest 9-Sound switch 10-Fuel consumption
11-Fuel date 12-Dial buttons 13-Fuel per hour 4-Light sensor 15-Fuel per 100km
16-Voltage, Throttle positison, Temperature, Lgnition angle, Acceleration 17-Seconds
18-Date 19-Mileage 20-Reel per minute 21-Kelio meter per hour 22-Mile per hour



.

Figure 2: Inter-vehicle distance following error $y1$

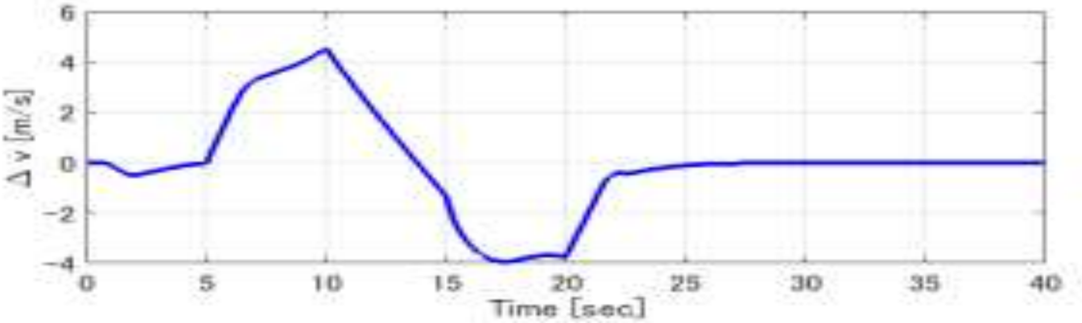


Figure 3 : Relative velocity $y2$

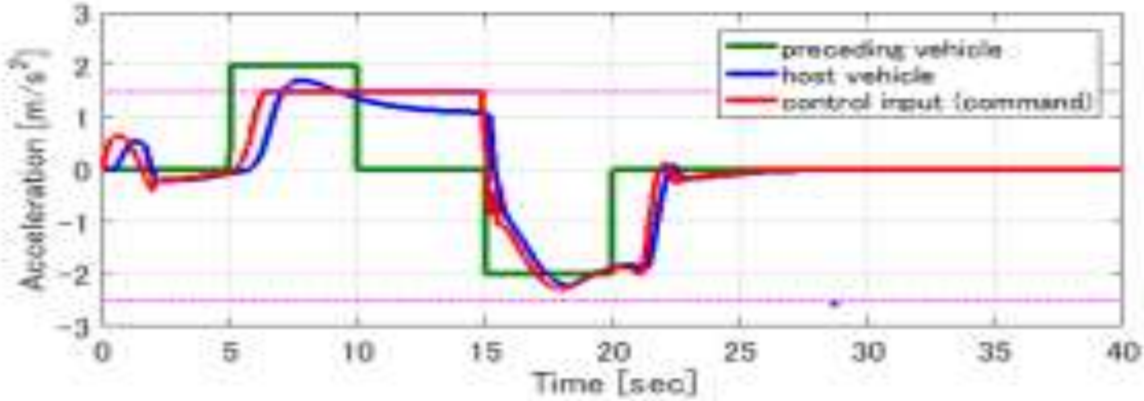
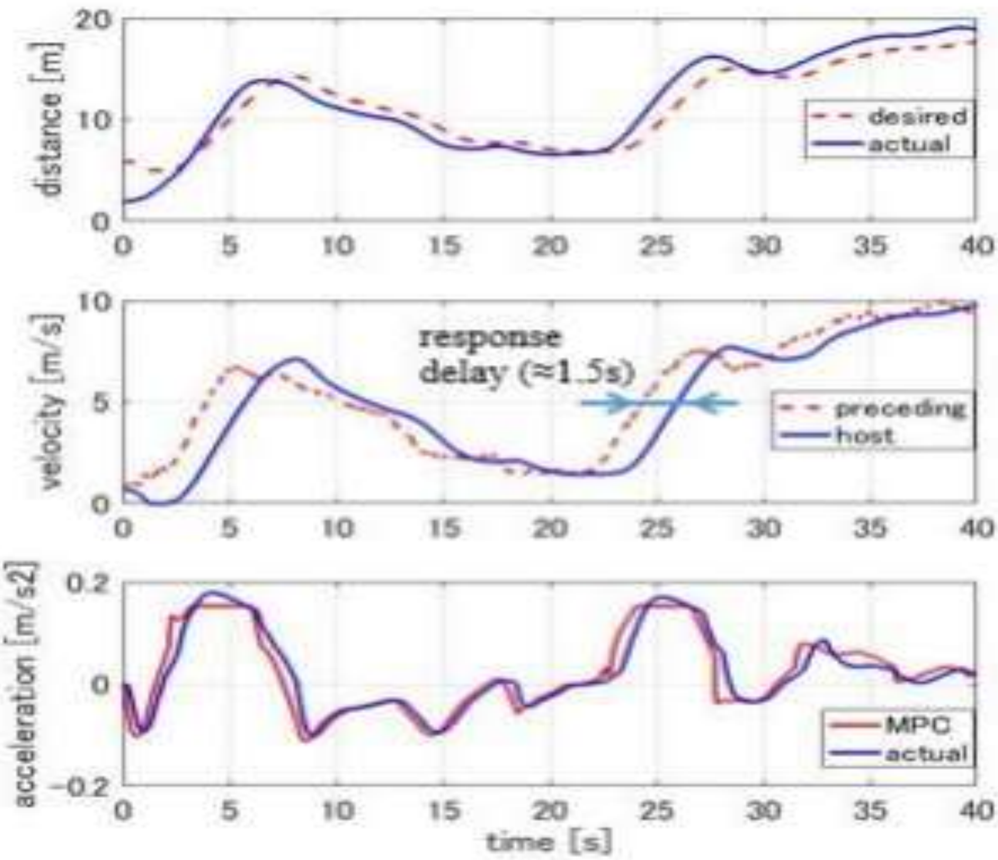


Figure 4: Acceleration $\dot{v}p$, y3, and u



scen

Figure 5: experimental result during traffic jam

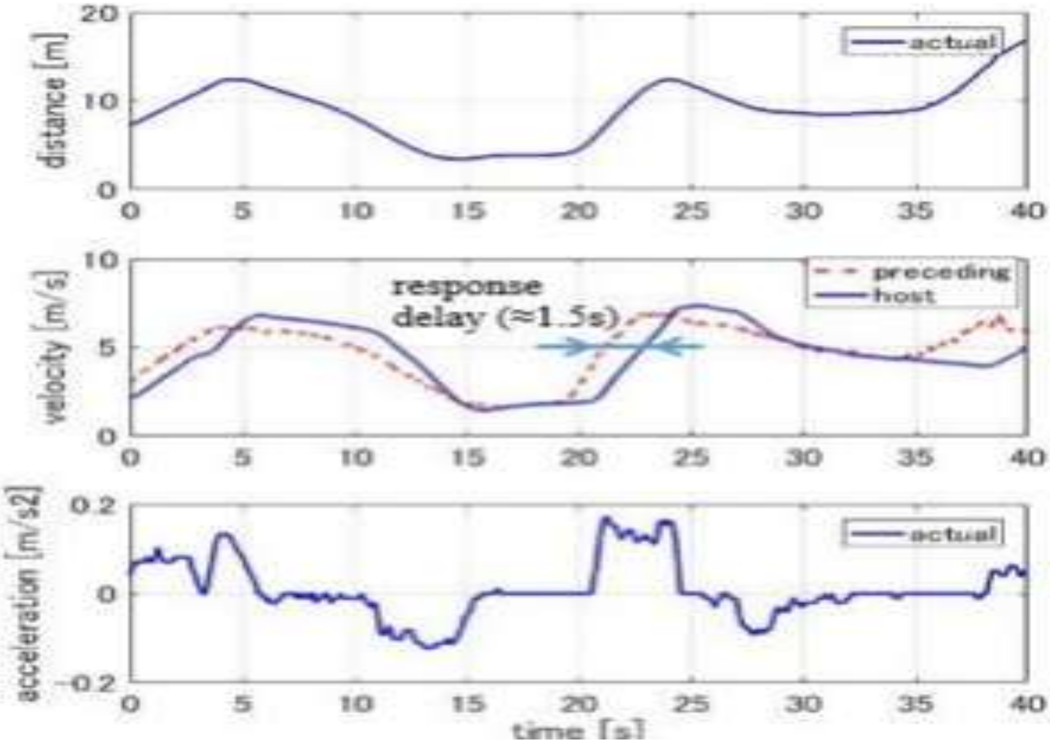


Figure 6: Vehicle behavior driven by normal driver

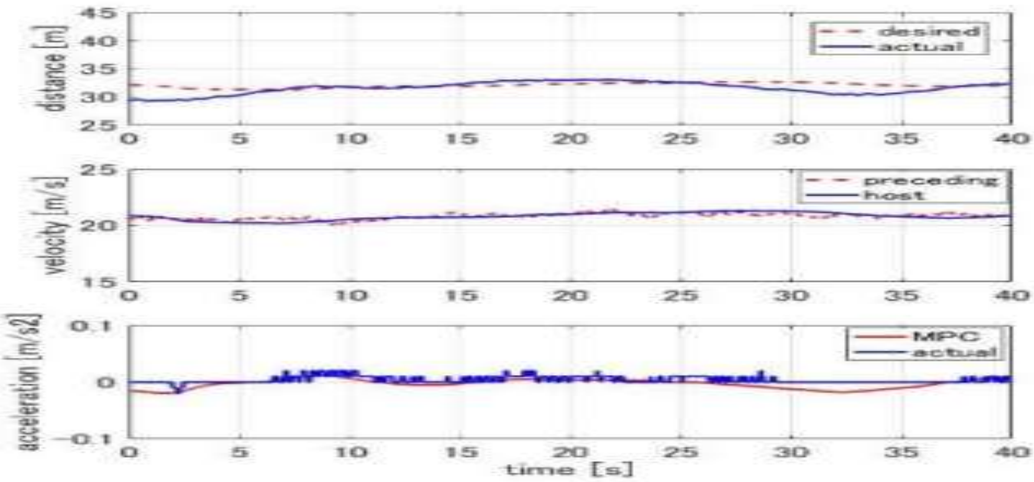


Figure7: Experimental result during high speed driving

Table 1: Applied Frequency and Measured Frequency

Applied Frequency	**Measured Frequency**
50.12 Hz	50.05 Hz
101.31 Hz	101.25 Hz
200.88 Hz	200.65 Hz
300.08 Hz	299.90 Hz
400.60 Hz	399.85 Hz
600.32 Hz	600.45 Hz
800.62 Hz	798.48 Hz
1000.98 Hz	999.15 Hz

Table 2: Calculation of speed

Frequency	**Speed**
0 Hz	0.00 km/hr
100 Hz	5.13 km/hr
400 Hz	20.52 km/hr
800 Hz	41.04 km/hr
1200 Hz	61.57 km/hr

Table 3: Accuracy in speed measurement

Reading from vehicle speedometer	Reading shown by our system
0 (km/hr)	0
10 (km/hr)	9.98
20 (km/hr)	19.99
30 (km/hr)	30.01

SOURCE CODE

APPENDIX: SOURCE CODE

```
main()
{
  int i, j = 0, gd = DETECT, gm;
  initgraph(&gd,&gm,"C:\\TC\\BGI");
  settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
  outtextxy(25,240,"Press any key to view the moving car");
  getch();
  setviewport(0,0,639,440,1);
  for (i = 0; i <= 420; i = i + 10, j++)
  {
    rectangle(50+i,275,150+i,400);
    rectangle(150+i,350,200+i,400);
    circle(75+i,410,10);
    circle(175+i,410,10);
    setcolor(j);
    delay(100);
    if (i == 420)
      break;
    clearviewport();
  }
  getch();
  closegraph();
  return 0;
}
//start global variables
char str[10] = "000";
unsigned char body_temp_str[7]="000 ";
unsigned char heart_beat_str[4]="00 0";
unsigned char in_char =0;
unsigned char out_char =0;
float time = 1;
float speed = 0;
// End global variables
// LCD module connections
sbit LCD_RS at PORTB0_bit;
sbit LCD_EN at PORTB1_bit;
sbit LCD_D4 at PORTB2_bit;
sbit LCD_D5 at PORTB3_bit;
sbit LCD_D6 at PORTB4_bit;
```

```
sbit LCD_D7 at PORTB5_bit;
sbit LCD_RS_Direction at DDB0_bit;
sbit LCD_EN_Direction at DDB1_bit;
sbit LCD_D4_Direction at DDB2_bit;
sbit LCD_D5_Direction at DDB3_bit;
sbit LCD_D6_Direction at DDB4_bit;
sbit LCD_D7_Direction at DDB5_bit;
// End LCD module connections
//start buttons
#define button_PIN1 PINC
#define button_DDR1 DDRC
#define button_PORT1 PORTC
#define button1_1 2
#define button1_2 3
#define button1_3 4
void init_buttons(){  //initialise buttons
button_DDR1 &= ~((1<<button1_3)|(1& lt;<button1_2)|(1<&l t;button1_1));
button_PORT1 |= (1<<button1_3)|(1< ;<button1_2)|(1<< button1_1);
}
//End buttons
//Start Sensors
#define sensors_PIN1 PINC
#define sensors_DDR1 DDRC
#define sensors_PORT1 PORTC
#define sensor1_0 0
#define sensor1_1 1
//define sensor1_2 2
void init_sensors(){ //initialise sensors
sensors_DDR1 &= ~((1<<sensor1_0)|(1& lt;<sensor1_1));
sensors_PORT1 &= ~((1<<sensor1_0)|(1& lt;<sensor1_1));
}
//End sensors
// Start outputs
#define outputs_PORT1 PORTD
#define outputs_DDR1 DDRD
#define buzzer 0        //pin number
//#define relay 1        //pin number
void init_outputs(){    //initialise outputs
outputs_DDR1 |= 1<<buzzer;
//outputs_DDR1 |= 1<<relay;
}
//End outputs
/******************** ************************** ************************** * MAIN PROGRAM
********************* ************************* ************************* void main(){
  Lcd_Init();                    // Initialize Lcd
  init_buttons();
  init_sensors();
  init_outputs();
  outputs_PORT1 &=~(1<<buzzer); //buzzer OFF
  Delay_ms(100);                 //wait for UART Initialize
  Lcd_Cmd(_LCD_CLEAR);              // Clear display
  Lcd_Cmd(_LCD_CURSOR_OFF);         // Cursor off
  Lcd_Out(1,1,"HWAY SPD CHECKER");
  Lcd_Out(2,1,"Initiati ng.......");
  Delay_ms(1000);
< div style="background:#eeeeee; border:1px solid #cccccc; padding:5px 10px">  while(1){
//////////////////// /////////////ENTRY start//////////////////// /////////////
  unsigned time = 1;
  unsigned int speed = 0;
  if((sensors_PIN1 & (1<< sensor1_1)))
```

```
{   outputs_PORT1 |= (1<<buzzer); //buzzer ON
    Lcd_Out(2,1,"SP D calculating..");
    Delay_ms(10);
    outputs_PORT1 &=~(1<<buzzer); //buzzer OFF
    while(1) {
        time++;
        if((sensors_PIN1 & (1<< sensor1_0))) {
            break;
        }
        Delay_ms(10);
    }
   speed = (1000/time);
   FloatToStr(speed, str);
   Lcd_Cmd(_LCD_CLEAR);              // Clear display
   Lcd_Cmd(_LCD_CURSOR_OFF);         // Cursor off
   Lcd_Out(1,1,"HWAY SPD CHECKER");
   Lcd_Out(2,1,"SPEED=&q uot;);
   Lcd_Out(2,7,"          ");
   Lcd_Out(2,7,str);
   Delay_ms(2000);


   if(!(button_PIN1 & (1<< button1_1))) {
     if(speed < 41) {
        Lcd_Cmd(_LCD_CLEAR);              // Clear display
        Lcd_Cmd(_LCD_CURSOR_ OFF);        // Cursor off
        Lcd_Out(1,1,"HI GHWAY SPD CHECKER");
        Lcd_Out(2,1,"SP EED =");
        Lcd_Out(2,7,"          ");
        Lcd_Out(2,7,str);
        Lcd_Out(2,13,"L OW ");
        outputs_PORT1 |= (1<<buzzer); //buzzer ON
        Delay_ms(5000);
        Delay_ms(5000);
        Delay_ms(5000);
        Delay_ms(5000);
        outputs_PORT1 &=~(1<<buzzer); //buzzer OFF
        goto start;
     }
   }
   else {
        if(speed > 60) {
        Lcd_Cmd(_LCD_CLEAR);              // Clear display
        Lcd_Cmd(_LCD_CURSOR_ OFF);        // Cursor off
        Lcd_Out(1,1,"HI GHWAY SPD CHECKER");
        Lcd_Out(2,1,"SP EED =");
        Lcd_Out(2,7,"          ");
        Lcd_Out(2,7,str);
        Lcd_Out(2,13,"H IGH");
        outputs_PORT1 |= (1<<buzzer); //buzzer ON
        Delay_ms(5000);
        Delay_ms(5000);
        Delay_ms(5000);
        Delay_ms(5000);
        outputs_PORT1 &=~(1<<buzzer); //buzzer OFF
        goto start;
#include<LiquidCrystal.h>
                              constint rs = 7, en = 6, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
                              LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
                              int sen1=11;
```

```
int sen2=12;
unsignedlong t1=0;
unsignedlong t2=0;
float velocity;
voidsetup()
{
  lcd.begin(16, 2);
pinMode(sen1,INPUT);
pinMode(sen2,INPUT);
  Serial.begin(9600);
  lcd.setCursor(0,0);
  lcd.print(" Speed Detector ");
}

voidloop()
{
while(digitalRead(sen1));
while(digitalRead(sen1)==0);
  t1=millis();
while(digitalRead(sen2));
  t2=millis();
  velocity=t2-t1;
  velocity=velocity/1000;//convert millisecond to second
  velocity=(5.0/velocity);//v=d/t
  velocity=velocity*3600;//multiply by seconds per hr
  velocity=velocity/1000;//division by meters per Km
for(int i=5;i>0;i
  {
   lcd.setCursor(3,1);
   lcd.print(velocity);
   lcd.print(" Km/hr   ");
delay(500);
   lcd.setCursor(3,1);
   lcd.print("            ");
delay(500);
  }
}

        }
      }
    }
    start:
    Lcd_Out(2,1,"NO VEHICLE detected");
  }
}
/******************** ************************** *************************** End of File
********************* ************************* **************************
```