CAN Network Based Longitudinal Velocity Measurement Using Accelerometer and GPS Receiver for Automobiles

K. Arun Venkatesh, N. Mathivanan

University Science Instrumentation Centre, Madurai Kamaraj University, Palkalai Nagar, Madurai, 625021, India, arunvenkateshk@gmail.com, nmvanan@yahoo.com

A design of Controller Area Network (CAN) based longitudinal velocity measurement system using MEMS accelerometer and GPS receiver is presented. CAN is a serial communication protocol which efficiently supports distributed control system with high level of security. The system consists of a master node and a slave node built around LPC1768 and LPC2129 microcontrollers respectively. The master and slave nodes are linked with CAN bus. The slave node gets velocity data from GPS receiver and transfers to the master node. The master node samples the accelerometer output, and saves the sampled data and the velocity data received from the slave node on a microSD card. A LabVIEW program has been developed to study the measured parameters in offline by applying a Kalman filter and computing the estimate of longitudinal velocity. Typical measurements with the present system in a standard driving maneuvers and computation of estimate of longitudinal velocity using the LabVIEW program are presented. The measurement system produces comparable results with the conventional meter.

Keywords: Accelerometer, CAN network, GPS, LabVIEW, Kalman filter

1. INTRODUCTION

NOWADAYS, GPS based navigation systems are common. GPS receivers are used to assist navigation by measuring the position, velocity and course angle. The GPS receivers are also used to measure side slip and roll [1], [2]. In places like underground tunnels, the GPS signals will not be available. Also in urban environment GPS signals can be very weak. In this case the position accuracy derived from GPS receiver might be lower than required. To improve GPS performance in terms of availability and reliability in weak signal environment and during GPS signal outages it is common to use external sensors such as accelerometers, gyroscopes, odometers, doppler radars etc [3]. Also multiple sensors are used to measure a single physical quantity to minimize error produced by each sensor. A low-cost MEMS based accelerometer can be used along with GPS receiver to measure the velocity of an automobile [4], [5], [6]. MEMS accelerometers are being used in measuring acceleration, velocity and vibration of a vehicle [7]. A Controller Area Network (CAN) based measurement system for measuring longitudinal velocity of an automobile using GPS receiver and MEMS accelerometer is designed.

CAN is a two-wire serial communication protocol which provides efficient and reliable communication between sensor, actuator, controller, and other nodes in real-time applications. CAN is a broadcast bus designed to operate up to 1 Mbps and in case of bus failure, CAN nodes support retransmission of data. CAN is efficiently used for automotive electronics by networking engine control units, anti-skid systems, anti-lock braking systems, safety features like airbags, monitor emission, steering performance and brake fluid levels [8]. The CAN networks are being used in Electronic Stability Control (ESC) system, which enables the stability of a car to be maintained during critical maneuvering and to correct potential under steering or over steering [9]. The proposed CAN network based measurement system follows a master slave approach. The master node sends a reference message to the slave node and the slave node responds to the message with the measurement data.

2. HARDWARE DESCRIPTION

The system consists of a master node and a slave node built around LPC1768 and LPC2129 microcontrollers respectively. Fig.1. shows the circuit diagram of the measurement system.

The slave node is used to acquire the velocity data, v_{GPS}, measured by the GPS receiver. It includes an ARM7 based LPC2129 microcontroller, a GPS receiver module (GPS-634R), a RS232 driver/receiver (MAX232) and a CAN transceiver (MAX13050). The GPS receiver has 51 channel acquisition engine and 14 channel track engine, which is capable of receiving signals from up to 65 GPS satellites and transferring them into the precise position and timing information. The GPS receiver generates an output which includes position (latitude & longitude), velocity, course angle, UTC time, etc. The GPS receiver provides v_{GPS} values in kmhr^{-1}. Since the GPS receiver generates and transmits the output at 1 Hz in NMEA 0183 version 3.01 standard [10] through serial communication at 9600 baud rate [11], the output of the GPS receiver module is interfaced to UART0 port of the LPC2129. The signal levels of GPS receiver pins are RS232 compatible and LPC2129 pins are TTL compatible [12]. Hence, MAX232 is used for necessary signal conversion between the GPS receiver and LPC2129.

The master node includes an ARM CORTEX M3 based LPC1768 microcontroller, a 3-axis accelerometer (MMA7361L) and associated signal conditioners, a 2GB microSD card and a CAN transceiver (MAX13050). The master node samples the accelerometer output and receives v_{GPS} from the slave node. The master node further saves the accelerometer data and v_{GPS} on a microSD card. The sensitivity of the accelerometer sensor is kept at ±0.0204 V/ms^{-2} by applying logic low to the g-Select pin. The operating range of the sensor is ±58 ms^{-2} [13].

Fig.1. Circuit Diagram of the Measurement System

The accelerometer has a response time of 1 ms. The X output of the accelerometer is buffered and filtered with anti-aliasing filter. Since the bandwidth response of the sensor is given as 400 Hz for X output [13], the buffered X output is filtered with anti-aliasing filter of 400 Hz cut-off frequency before quantization to prevent aliasing. The buffer and the anti aliasing filter use LM324 op-amp. The output of the filter is applied at the analog input of LPC1768. The microcontroller has a built-in unipolar, 12-bit ADC operating with 3.3 V reference voltage and supports 8

analog input channels [14]. The ADC is used to quantize the outputs of the accelerometer. Since the ADC operates with +3.3 V reference, the resolution of the 12-bit ADC is 0.806 mV. Hence, the resolution of the measurement system is 0.0395 ms^{-2}. The resolution of the measurement system can be further increased to 0.00987 ms^{-2} by applying logic high to g-select pin of the accelerometer. The microSD card [15] is interfaced to the SPI port of the LPC1768. An UTP cable is used to provide the physical connection between the CAN transceivers.

3. SOFTWARE DESCRIPTION

A. Firmware

Two separate firmware for the master and slave nodes have been developed in 'C' using IAR Embedded Workbench IDE. The built-in CAN controllers of both nodes are initialized with 29-bit identifier (CAN 2.0B compliant), 12 CAN clocks of synchronization jump width, 3 CAN clocks of segment delays and for communication at 100 kbaud. The functions of firmware are listed below.

Slave node: The firmware in the slave node performs the following functions.

1. Initializes the on-chip CAN controller. Initializes the UART0 port for communication in 8N1 format at 9600 baud rate.

2. Reads UART0 and gets output data of the GPS receiver. Parses the GPS receiver data into v_{GPS} and writes the data and an error status byte into a buffer for transmission. If v_{GPS} data is not received properly, it indicates the same in the error status byte.

3. Checks if "REMOTE FRAME" is received from the master node and transmits the latest velocity data, i.e. v_{GPS}, and the error status byte in the buffer to the master node. It is implemented as an interrupt routine.

4. Repeats the steps 2 and 3 at an interval of one second.

Master node: The firmware in the master node performs the following functions.

1. Initializes the on-chip PLL to generate a core clock frequency of 100 MHz using a 12 MHz external crystal. Initializes the on-chip CAN controller, real-time-clock and SPI port.

2. Samples the analog input at 800 Hz sampling rate. The required timing signal for sampling is derived from internal free-running timer which runs on the system core clock.

3. Sends "REMOTE FRAME" to the slave node at the end of every second. Receives v_{GPS} data and the error status byte returned by the slave node.

4. Frames the data, i.e. the quantized accelerometer output Q_X, velocity data (v_{GPS}), and error status byte into a packet and stores in a file created in the microSD card [16].

5. The program creates a new file for every 5 minutes and saves data sampled during the period in the file. The 5 minute interval is arbitrarily chosen for comfortable data handling.

B. Velocity Estimation and Kalman Filter

The acceleration of the vehicle in the direction of X-axis, $a_X(t)$ in ms^{-2}, is given by

$$a_X(t) = \left(\left(\frac{Vref}{2^N - 1} \times Q_X(t) \right) - offset \right) \times \frac{1}{S} \quad (1)$$

Where,
Vref – reference voltage of the ADC which is 3.3 V
N – number of bits of resolution of the ADC which is 12
S – sensitivity of the accelerometer which is 0.0204 V/ms^{-2}
$Q_X(t)$ - quantized data of X output at time t
offset - output voltage of the accelerometer at rest in V

The velocity $v_{XA}(t)$ in ms^{-1} is obtained by integrating the acceleration [17], and is given by,

$$v_{XA}(t) = \int_0^t a_X(t)\,dt \quad (2)$$

The velocity computed from acceleration measurement with accelerometer, v_{XA} and the velocity provided by GPS receiver, v_{GPS} are used to compute the estimate of the longitudinal velocity, v_E, of the automobile. When valid v_{GPS} is not available, the system uses the v_{XA} alone to compute v_E. The sampling frequency of $v_{GPS}(t)$ is 1 Hz. The sampling frequency of $v_{XA}(t)$ is 800 Hz. Hence, average of $v_{XA}(t)$ in one second period is computed to represent the $v_{XA}(t)$ at 1 Hz sampling frequency. The velocity of the vehicle is estimated by combining $v_{XA}(t)$ and $v_{GPS}(t)$ using the Kalman filter. Also in discrete time sampled data system the measurements include additive random noises and the Kalman Filter removes the noises [18]. Fig.2. shows the removal of noise using Kalman filter. A brief introduction to Kalman filter is given below.
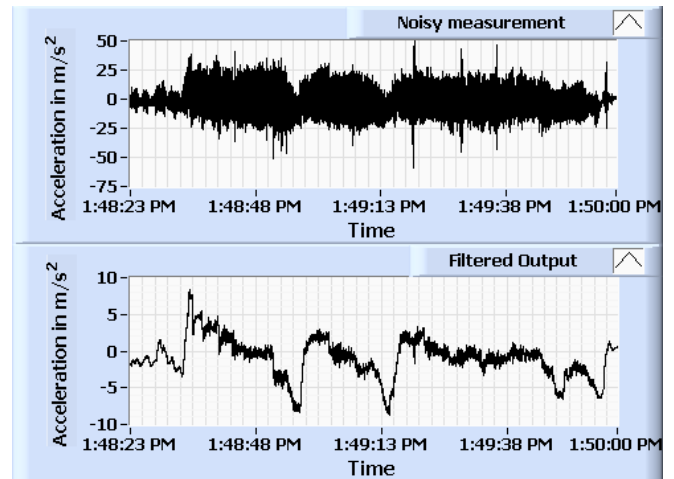

Fig.2. Removal of noise using Kalman filter

Kalman Filter: The Kalman filter is used for combining inexact forecast of a system's state with an inexact measurement of the state [18]. Also, it is widely applied on measurements in which two independent instruments with different noise variances measure a same parameter [19]. The Kalman filter time update equations are given as

$$\hat{x}^-[t] = x_1[t] + \hat{x}[t-1] \quad (3)$$

$$P^-[t] = P[t-1] + Q \qquad (4)$$

The Kalman filter measurement update equations are given as

$$K[t] = \frac{P^-[t]}{P^-[t] + R} \qquad (5)$$

$$\hat{x}[t] = \hat{x}^-[t] + K[t]\left(x_2[t] - \hat{x}^-[t]\right) \qquad (6)$$

$$P[t] = P^-[t]\left(1 - K[t]\right) \qquad (7)$$

Where,

$\hat{x}^-[t]$ - predicted state estimate at t

$\hat{x}_1[t]$ - V_{XA} at t

$\hat{x}_2[t]$ - V_{GPS} at t

$P^-[t]$ - predicted covariance estimate at t

Q - process noise covariance

$K[t]$ - Kalman gain at t

R - measurement noise variance

$\hat{x}[t]$ - posteriori state estimate at t (estimated velocity v_E)

$P[t]$ - posteriori covariance estimate at t

C. LabVIEW program

An application program has been developed on LabVIEW platform to analyze the measurement data on the microSD card. Fig.3., Fig.4. and Fig.5. show the front panel and block diagram of the application program. The application program performs the following functions.

1. Reads the microSD card and parses the measurement data into Q_X and v_{GPS}.

2. Computes v_{XA} from Q_X using the equations (1) and (2).

3. Estimates v_E by combining v_{XA} and v_{GPS} using the Kalman filter. If valid v_{GPS} data is not available, the program estimates v_E using v_{XA} and the Kalman filter is used to remove the process and measurement noises only.

4. Plots v_{XA}, v_{GPS} and v_E against time in waveform graph.
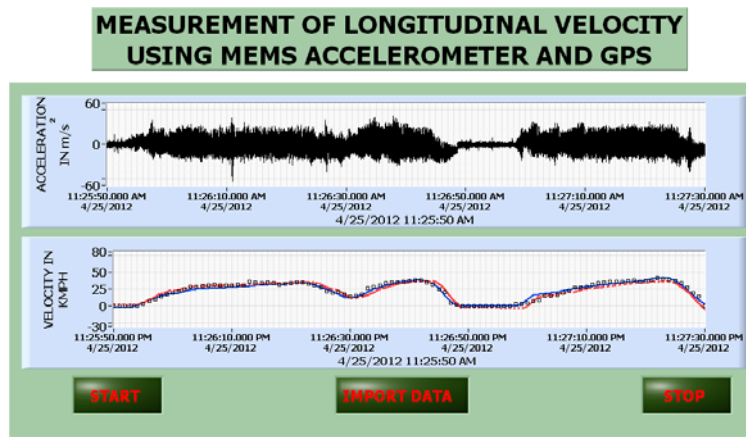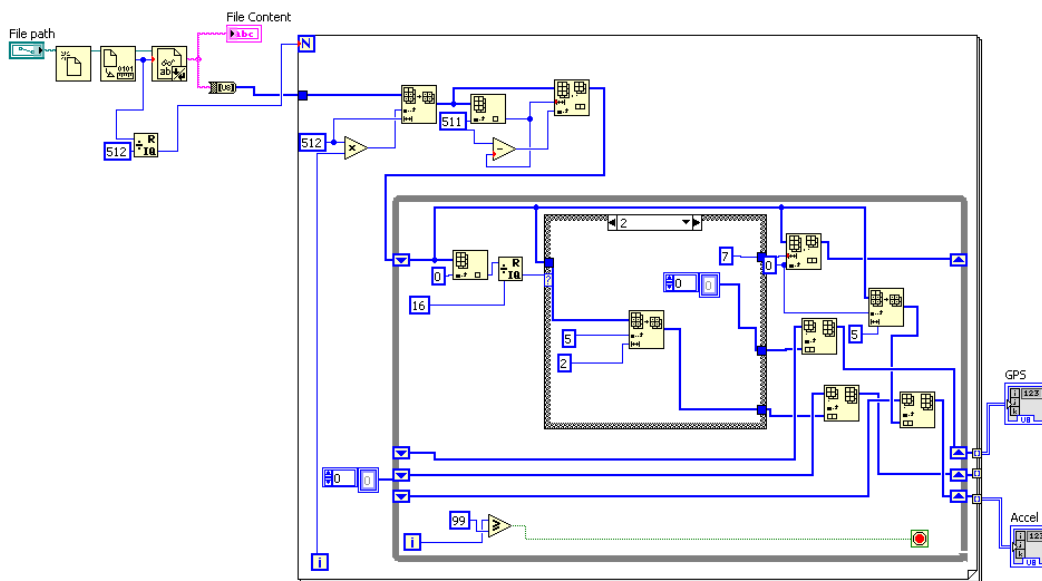

Fig.3. Front Panel of the LabVIEW program

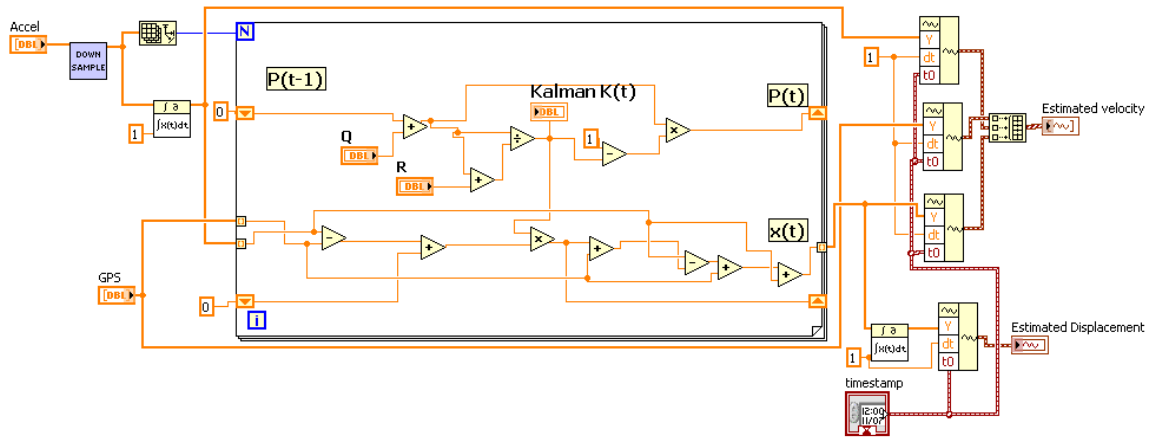Fig.4. Block diagram of the LabVIEW program to parse the measurement data from the file.

Fig.5. Block diagram of the LabVIEW program to estimate velocity and displacement using Kalman filter.

4. RESULTS AND DISCUSSION

The measurement system is mounted on an automobile and the automobile is driven in standard driving maneuvers. The measured data is analyzed in offline using the LabVIEW program and the following important features of the system are observed.

	Trial distance in m	MSE between V_E & conventional meter in m/s	MSE between V_{XA} & conventional meter in m/s
Trial 1	250	0.258	0.789
Trial 2	400	0.152	0.560
Trial 3	500	0.081	0.280
Trial 4	1000	0.250	0.812
Trial 5	2250	0.308	1.109

Table 1. Comparison with conventional meter

Comparison with Conventional meter: The automobile is driven on five different tracks inside the premises of Madurai Kamaraj University, India. The estimated velocity is compared with the velocity of the vehicle measured by the conventional meter. Since the conventional meter of the vehicle has very low resolution, a Hall-effect sensor and a data logger is fitted to the conventional meter to measure the velocity. The experimental procedure will be discussed in a future paper.

The mean square error between the estimated velocity v_E and the velocity measured using the conventional meter for each trial is computed and are listed in Table 1. The average of the mean square errors for the trials is found as 0.2098 m/s with variance 0.0084. Fig.6. shows the plot of velocity measured using conventional meter and the estimated velocity v_E for Trial 3. During this trial, the mean square error between v_{XA} and velocity measured using conventional meter is found as 0.280 m/s. The mean square error between the estimated velocity v_E and the velocity measured using the conventional meter is found as 0.081 m/s. It is found that the addition of GPS measured velocity improves the estimated velocity.
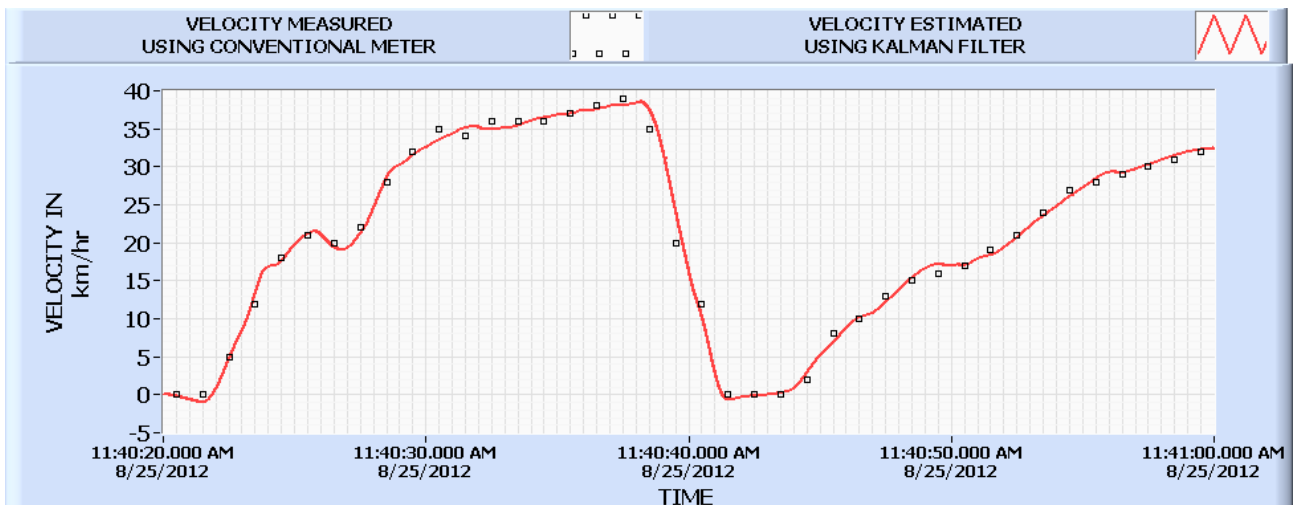

Fig.6. Velocity measured using conventional meter and the estimated velocity v_E.

Displacement Measurement: The estimated velocity data is used to estimate the total distance travelled. The displacement is estimated by integrating the estimated velocity. The automobile is driven in a dry asphalt surface for a distance of 400 m inside the campus of Madurai Kamaraj University, India. The displacement results are shown in Fig.7. The test is carried out multiple times and it is found that the error in the measurement of displacement lies within ±5 m in each test corresponding to a percentage error of 1.25%.

GPS Signal Outage: In a typical urban environment, it is common that the GPS receiver loses the tracking of satellites and may generate error measurements. In such case, the measurement system should follow the velocity measured with accelerometer. After the GPS receiver regains the tracking of satellites, the measurement system must include the GPS measured velocity in the estimation process. The slave node is disconnected from the master node for 20 s to simulate GPS signal outage [20]. Fig.8. shows the plot of estimated velocity during GPS signal outage. The results show that the v_E traces v_{XA} during the GPS signal outage.
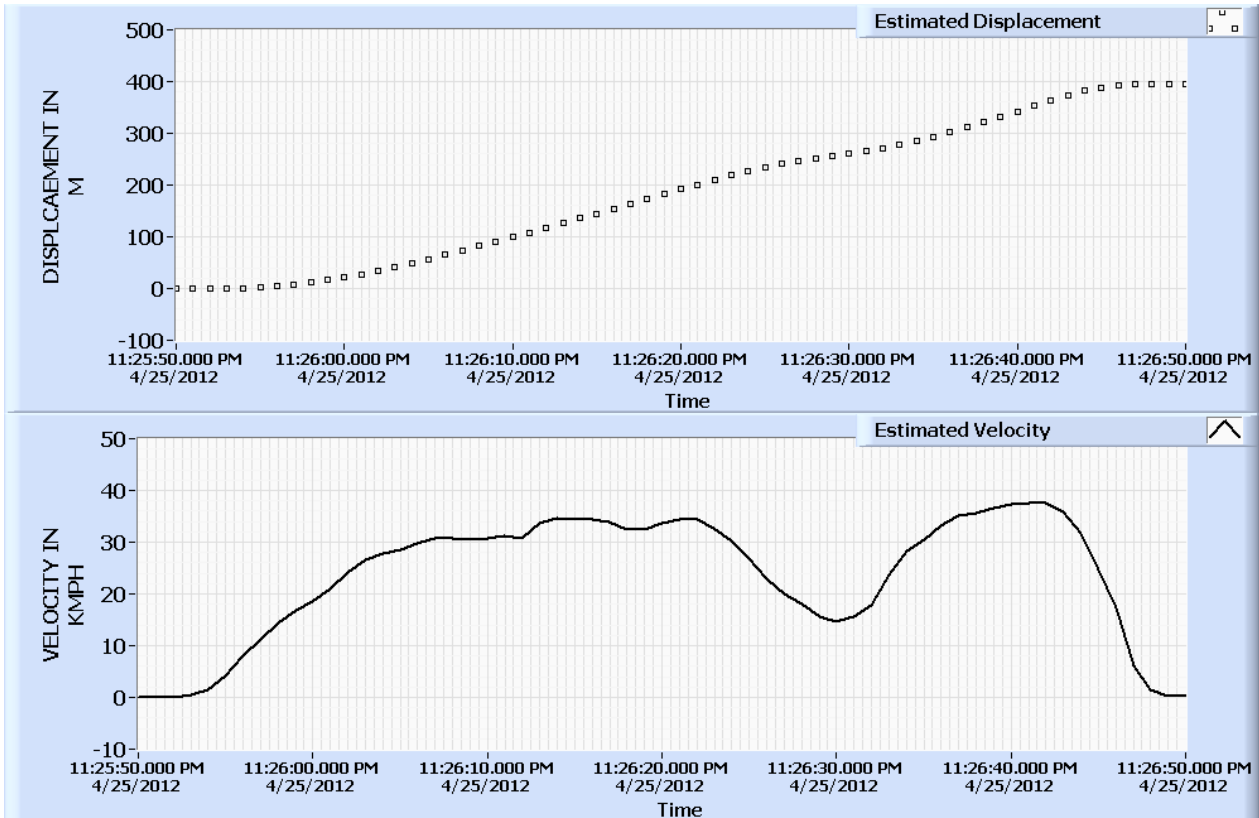

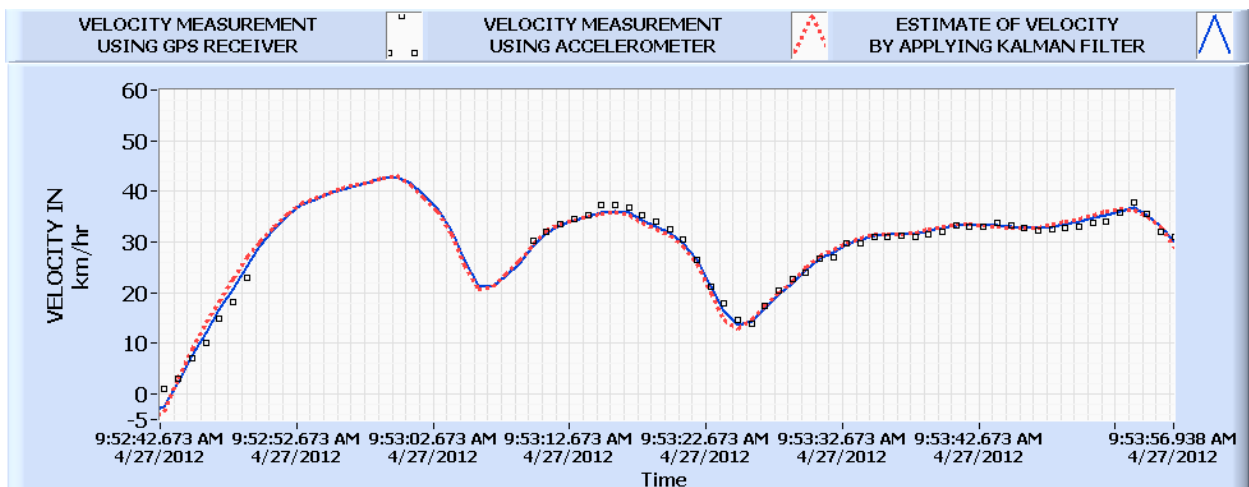Fig.7. Estimated displacement and velocity for 400m test drive.

Fig.8. Estimation of longitudinal velocity during GPS signal outage.

5. CONCLUSION

The system measures the longitudinal velocity using GPS and MEMS accelerometer. The CAN network used in the measurement system provides efficient communication between the nodes. During GPS signal outage, the system is able to track the position of the automobile using accelerometer. The mean square error between the estimated velocity and the velocity measured using conventional meter is as low as 0.08m/s. The estimated displacement has a percentage error of 1.25% which is considerably accurate. The system performance can be further improved by incorporating wheel speed sensors to provide additional measurement. The microcontroller in the master node can be programmed to implement the Kalman filter algorithm to measure the velocity dynamically. The measurement system can be used to acquire acceleration values in all three axes which can be used to measure the parameters like side slip, yaw rate, roll, pitch, etc of the automobile. Additional sensor like gyroscope can be added to CAN node to determine the yaw rate which is another important parameter in electronic stability control.

REFERENCES

[1] Ryu, J., Rossetter, E.J., Gerdes, J.C. (2002). Vehicle sideslip and roll parameter estimation using GPS. In *International Symposium on Advanced Vehicle Control (AVEC)*, Hiroshima, Japan.

[2] Bevly, D.M., Gerdes, J.C., Wilson, C. (2002). The use of GPS based velocity measurements for measurement of sideslip and wheel slip. *Vehicle System Dynamics,* 38 (2), 127–147.

[3] Davidson, P., Hautamaki, J., Collin, J. (2008). Using low-cost MEMS 3D accelerometer and one gyro to assist GPS based car navigation system. In *15th International Conference on Integrated Navigation Systems*, St. Petersburg.

[4] Grewal, M.S., Weill, L.R., Andrews, A.P. (2001). *Global Positioning Systems, Inertial Navigation, and Integration.* John Wiley & Sons.

[5] Farrell, J., Barth, M. (1998). *The Global Positioning System & Inertial Navigation.* McGraw-Hill.

[6] Acharya, A., Sadhu, S., Ghoshal, T.K. (2011). Train localization and parting detection using data fusion. *Transportation Research Part C: Emerging Technologies*, 19 (1), 75-84.

[7] Stein, G.J., Chmúrny, R., Rosík, V. (2011). Compact vibration measuring system for in-vehicle applications. *Measurement Science Review,* 11 (5), 154-159.

[8] Johansson, K.H., Törngren, M., Nielsen, L. (2005). Vehicle applications of controller area network. In *Handbook of Networked and Embedded Control Systems.* Birkhäuser Boston, 741-765.

[9] Tseng, H.E., Ashrafi, B., Madau, D., Allen Brown, T., Recker, D. (1999). The development of vehicle stability control at Ford. *IEEE/ASME Transactions on Mechatronics*, 4 (3), 223-234.

[10] SiRF Technology Inc. (2005). *NMEA Reference Manual*. Rev. 1.3 January 2005.

[11] Cyber i-Technologies Co., Ltd. (2010). *GPS-634R Technical Data Sheet*. Ver. 1.4.

[12] NXP B.V. (founded by Philips) (2008). *UM10114 – LPC21xx and LPC22xx User Manual*. Rev. 03 – 2 April 2008.

[13] Freescale Semiconductors, Inc. (2008). *±1.5g, ±6g Three Axis Low-g Micromachined Accelerometer – Technical Data*. Document Number MMA7361L, Rev 0, 04/2008.

[14] NXP B.V. (founded by Philips) (2010). *UM10360 – LPC17xx User Manual*. Rev. 2 – 19 August 2010.

[15] Coopmans, C., Yang Quan Chen (2008). A general purpose low-cost compact spatial-temporal data logger and its applications. In *IEEE AUTOTESTCON 2008,* 8-11 September 2008. IEEE, 64-68.

[16] Venkatesh, K.A., Mathivanan, N. (2012). Design of MEMS accelerometer based acceleration measurement system for automobiles. *Measurement Science Review,* 12 (5), 189-194.

[17] Liu, H.H.S., Pang, G.K.H. (2001). Accelerometer for mobile robot positioning. *IEEE Transactions on Industry Applications*, 37 (3), 812-819.

[18] Welch, G., Bishop, G. (2002). *An Introduction to the Kalman Filter*. TR-95-041. Department of Computer Science, University of North Carolina at Chapell Hill.

[19] Rojas, R. (2003). *The Kalman Filter*. Technical Report. Freie University of Berlin.

[20] Davidson, P., Vazquez, M.A., Piche, R. (2009). Uninterrupted portable car navigation system using GPS, map and inertial sensors data. In *13th IEEE International Symposium on Consumer Electronics*, Kyoto, Japan, 836-840.