Analyzing Accuracy of GPS Data for Vehicular Parameters

Gaurav Sood, Jordan Willis and Arunita Jaekel
School of Computer Science, University of Windsor
Windsor, Canada
{sood111, willisj, arunita}@uwindsor.ca

Shaik Nazeer and Dominic Paulraj
Arada Systems
Windsor, Canada
{dominic, snazeer}@aradasystems.com

***Abstract*— Currently, most vehicular parameters, such as speed acceleration and heading, are obtained from GPS data. In this paper our goal is to investigate an alternate source of vehicular information for collision avoidance systems and driver assistance applications, which is more accurate, reliable and has significantly lower time lag. We have implemented an architecture, which enables us to read, analyze, decode and store the real-time vehicular data from the vehicle's electric sensors. We have also presented and analyzed experimental data showing the errors in vehicular speed obtained from GPS readings, under different driving conditions.**

Keywords—Intelligent Transportation Systems (ITS); Vehicle to Vehicle (V2V) Communication; Controller Area Network (CAN); GPS; On Board Unit (OBU)

I. Introduction

Currently, most of the collision avoidance algorithms rely on GPS [1] for the real-time situation parameters of the vehicles. For example, the collision warning system presented in [2] and [3], calculates *Time to Collision (TTC)* between two vehicles using GPS information (e.g. longitude, latitude, speed, heading, altitude and acceleration) exchanged using Vehicle to Vehicle (V2V) [4] communications. The data obtained from GPS signals is significantly less accurate during bad weather conditions, in urban downtown areas and tunnels because of the distorted or no GPS signals received in these conditions. In addition, the polling rate of current GPS receivers is low with intervals up to 200 milliseconds, which may not be adequate for real-time systems like collision avoidance systems. Also, the current source of information like GPS provides access to only limited number of vehicular parameters like speed and acceleration. This has led to the development of collision avoidance algorithms with limited scope and accuracy. All these factors contribute to inaccurate calculation of TTC [2], which result in wrong or no warnings of the impending collisions.

In this paper, we implement a software application to extract relevant information e.g. speed, engine rpm and brake status directly from the vehicle instead of present sources e.g. GPS. Positioning information like longitude, latitude, altitude and heading still must be taken from GPS. In modern vehicles, all the electric devices of the vehicle are connected to each other by a bus network called Controller Area Network (CAN) [5]. The vehicle's electric devices communicate with each other through CAN protocol, which helps the devices to access the most recent information from various components. Extracting data directly from vehicle's sensors result in more accurate data with reduced time lag. The higher polling rate of the data from the CAN protocol (up to 5 milliseconds), also helps in capturing the minute changes in the values of the vehicular parameters

Our software module named *CAN Ethernet* [6] reads, analyzes, decodes and stores the CAN data read from the CAN bus in real-time. The software runs as a server application to act as a source of CAN data and enables up to five different client connections at same time. We are able to extract a number of vehicular parameters like speed, brake status, steering angle, yaw rate, accelerator pedal position, headlight status, wiper status. All these parameters information can prove very useful in developing and improving collision avoidance systems and driver assistance applications.

In the remainder of this paper, we focus specifically on vehicular speed and compare the speed extracted using our proposed software to the corresponding GPS values. We investigate how the accuracy of the GPS data varies under different driving conditions and report our findings.

II. Extracting information from CAN bus

Our application software is designed to send specific commands to an Ethernet dongle, which are interpreted to the CAN specific format and used to request raw data from the CAN bus. After receiving a legitimate command, electronic devices connected to the CAN bus respond by sending the requested data. After receiving the raw data, the application analyzes the received raw data at bit level, and decodes it to a human readable form. The decoded data is then made available for collision avoidance algorithms and other applications.

The final decoded value of the vehicle's sensors is stored in a data structure, which is continuously updated with new values. This application accepts client connections from the other applications requiring CAN information. Upon receiving a request our application dumps whole data structure to the client, providing real time CAN information to collision avoidance applications and other driver assistance applications.

III. Test results and analysis

Testing of our software was carried out on a Jeep Grand Cherokee (Year: 2007), using an OBU (Locomate), from Arada Systems [7]. The OBU was connected to the vehicle's

978-1-5090-0264-1/15/$31.00 ©2015 IEEE

DOI 10.1109/ICCVE.2015.4

CAN bus as well as a GPS antenna. Testing was carried out in parking lots, downtown areas and highways to check the behaviour of GPS signals at different locations and for different speeds. All field tests were done in and around the Detroit, MI metropolitan area in USA.

A. *Test Results at a Parking Lot*

The first test was conducted at a parking lot on a clear sunny day, so we expected comparatively less interference from atmospheric conditions on GPS signals. The vehicular speed extracted from the CAN bus data and the GPS signals are denoted as *CAN speed* and *GPS speed* respectively Fig. 1 shows the values of *CAN speed* and *GPS error (= GPS speed – CAN speed)* taken at specific intervals. The values of *CAN speed* and *GPS error* are on recorded on the left y-axis and right y-axis respectively, with total number of records on the x-axis. We found that the *GPS error* can reach up to -7 km/h when the vehicle is braking and coming to a stop from a speed of 20km/h. Normally, *GPS error* ranges from -5 km/h to 5 km/h tends to increase significantly when the vehicle is either accelerating or braking as compared to when it is running at a constant speed.

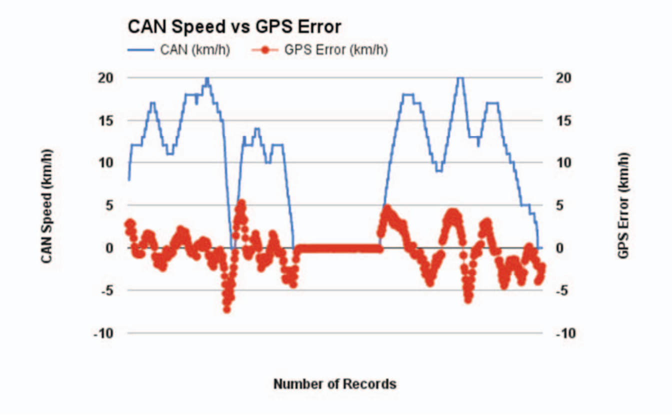

Fig. 1. Comparison of CAN Speed and GPS Error at a Parking Lot

B. *Test Results at Downtown*

The second test was conducted in downtown locations. Fig. 2, shows how *GPS error* varies with *CAN speed.* The results show a constant error of around +5 km/h when the vehicle was accelerating from 0 km/h to 35 km/h. The *GPS error* reaches to maximum of -10 km/h during the braking event of the vehicle, when the speed came down from 35 km/h to 0 km/h.

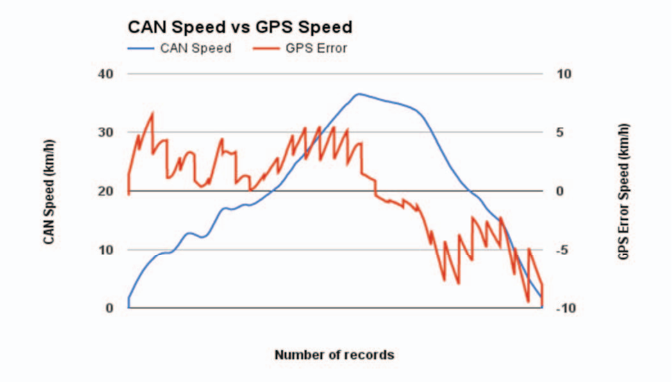

Fig. 2. Comparison of CAN Speed and GPS Error at Downtown

C. *Test Results on a Highway*

The third test was conducted on I-75 highway, on a day with overcast conditions. Fig. 3 shows that the *GPS error* is between 0 to -5 km/h when the vehicle was going at around 110 km/h. The *GPS error* reaches to maximum of -12 km/h when the vehicle decelerates from 95km/h to 70 km/h.

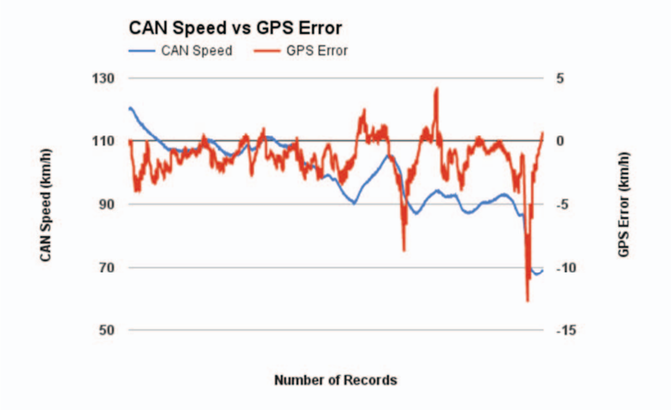

Fig. 3. Comparison of CAN Speed and GPS Error on a Highway

IV. CONCLUSION

In this paper, we present a software application for extracting speed, acceleration and other relevant data directly from the vehicle's sensors. Such an application will help to provide a better alternative to existing sources of vehicular information like GPS. In particular, we have focused on vehicular speed to show the discrepancy between values read from the CAN bus and those obtained from the GPS. One important factor contributing to this error is the lower polling rate for GPS signals. Our preliminary results show that the errors in the speed given by the GPS signals are higher during the events of the vehicle's acceleration and braking. So, a more accurate and consistent knowledge of the vehicle's speed will help to generate timely warnings. We are continuing our current work to develop improved collision avoidance algorithms, based on more accurate information and additional parameters obtained using our software.

ACKNOWLEDGMENT

This work has been supported in part by Ontario Centres of Excellence (OCE) and Arada Systems, Windsor.

REFERENCES

[1] http://www.loc.gov/rr/scitech/mysteries/global.html

[2] A. Gumaste, R. Singhai, and A. Sahoo, "IntelliCarTS: Intelligent car transportation system," Proc. IEEE LANMAN, June 2007.

[3] A. Gumaste, R. Singhai, and A. Sahoo, "Intelligent vehicular transportation system (InVeTraS)," in Australasian Telecommunication Networks and Applications Conference, December 2-5 2007.

[4] T. Samad, and A. M. Annaswamy, "Vehicle-to-vehicle/vehicle-to-infrastructure control from the impact of control technology," IEEE Journals and Magazines, www.ieeecss.org, 2011.

[5] R. Bosch, "CAN specification version 2.0," at http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/can2spec.pdf

[6] G. Sood, "Architecture for extracting data from vehicular sensors," M.S. thesis, School of Computer Science, Univ. of Windsor, 2015.

[7] .http://www.aradasystems.com