Agilent Engineering Excellence Program: Collision Avoidance System

Ng Weisheng Joseph

***Abstract*— The development of a suitable test bed for an automobile collision avoidance system is accomplished on a 1:10 scale model car. The test bed here refers to the final demonstrable product. The system integrates sensors placed on the vehicular body to detect a number of critical parameters which adequately describe the vehicular dynamics. Various collision avoidance algorithms are implemented using the real time data attained from the sensor system. Visual and auditory feedback systems which adequately give warnings to mitigate impending collisions are implemented in real time. The implemented hardware setup successfully detects and warns the driver of possible future collisions thereby allowing him or her to apply the brakes manually to avoid the impending collision.**

The development of the test bed is accomplished by using Agilent's U2353A data acquisition unit (DAQ) and Agilent's Visual Engineering Environment pro (VEE pro). Various other sensor components to detect linear distance and rotation speed have also been integrated into the system. The final product effectively showcases the capability of Agilent's DAQ and VEE pro's flexibility in integrating various sensor components to develop a test bed for an automobile collision avoidance system.

I. Introduction

THE motivation for the development of collision avoidance (CA) systems is the improvement which they can bring to vehicular safety. In the UK, it was found that 75% of all crashes occurred at speeds lower than 20mph (32km/h) [1]. In the US, the report "Traffic Safety Facts 2005" released by the National Highway Traffic Safety Administration (NHTSA) found that 43% of all vehicular-vehicular collisions were rear end collisions.

Low speed crashes indicate the need for a CA system well suited for in city driving. Rear end collisions make up a bulk of most vehicle to vehicle collisions and therefore deserve special attention. CA systems should be adequately employed to mitigate such collisions.

The push for automobile CA systems to improve vehicular safety has created a large market for non-entertainment automotive electronics. According to In-Stat, a provider of information resources and analytical assets, the world market for non-entertainment automotive electronics was estimated at US$36.8 billion in 2005 and is expected to reach US$52.1 billion by 2010. Automobile safety and convenience systems accounted for 50.3% of the global market in 2005 and were valued at US$18.5 billion. Growth from driver assistance systems such as collision avoidance systems and lane departure systems is expected to be strong [2].

The objective of this project is the development of a CA system using Agilent VEE pro and U2353A DAQ to emulate the CA system on a car. The tasks involved include understanding the operations of Agilent VEE pro and U2353A DAQ, the implementation of the test bed for the CA system and the development of the software required for the task to be carried out. The main focus of the project will be the development and implementation of a front end CA system. Of secondary focus is the development of lateral and rear CA systems, both of which work on the same principles as the front end CA system. Here, only the front end CA system will be discussed in detail.

The front end CA system implemented consists of an SRF08 ultrasonic module to detect preceding vehicular range and an axle rotary encoder for obtaining host vehicle speed. A PIC microcontroller is used in conjunction with VEE pro and the DAQ to develop the experimental test bed. The implemented test bed uses a stationary 1:10 scale model radio controlled (RC) car to emulate the CA system of a car. CA algorithms have been modified and implemented in order to calculate the warning parameter *w*. Visual and auditory systems have been put in place to give varying levels of warning to the driver as set by *w*.

This technical paper is organized as follows – Section II presents the design and integration of the proposed test bed. Section III documents the analysis and implementation of the CA algorithm. In section IV, the software flow structure for the test bed is discussed. In section V, the optimization and evaluation of the test bed is covered. This paper ends with a conclusion in section VI.

II. Design of Experimental Test Bed

A. Agilent VEE pro and the U2353A DAQ

The use of both VEE pro and the U2353A are central to this project. Hence, the test bed must be designed with these 2 systems in mind. Agilent VEE pro is a powerful visual engineering environment which interfaces with instruments from any vendor using GPIB, LAN, USB, RS-232, VXI and other interfaces and buses. It provides a fast, easy and intuitive solution for the measurement and analysis of data from the interfaced instruments. It supports popular

Manuscipt received on the May 16, 2008. This work is supported by Agilent Technologies.

Ng Weisheng Joseph is with the Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore (email: joseph.ng@nus.edu.sg)

software tools like .NET framework and ActiveX and comes embedded with the MATLAB script and the MathWorks Signal Processing Toolbox.

The U2353A DAQ is a high performance multifunction data acquisition unit. It comes with an array of analog to digital converters, digital to analogue converters and digital input and output channels.

B. *Proposed platform for test bed*

The suitability of a number of sensors for front end CA systems was reviewed. These include radar, laser and ultrasonic sensors. Of the all the sensor systems considered, radar is the only sensor which has seen implementation on commercial automobiles. It is by far the most robust system encountered. However, the complexity for the development and integration of such a sensor within the allocated time and resources poses a tremendous challenge. An example of this test bed is shown in Fig. 1 where a motion belt simulator imitating the motion of the preceding vehicle is placed in front of the integrated radar module.

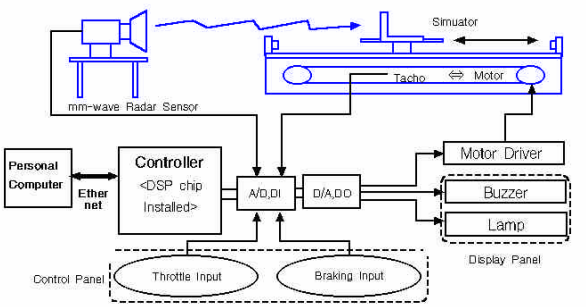

Fig. 1. Schematic diagram for hardware in loop simulation of a front end CA system. [3]

Much research work is still being done on laser CA sensors. These systems have yet to make their presence felt in the commercial automotive market. Present research on laser CA systems has yet to prove the effectiveness of them for use in automobile CA systems [4]. Ultrasonic sensors on the other hand have mainly been used for side and rear CA systems due to limitations in their speed resolution.

Considering all the above factors, the only option available is the scaling down of the test bed to involve a non-automotive platform. The proposed platform is a stationary 1:10 scale model RC car embedded with an ultrasonic sensor. There are a number of benefits for such as system. Its maximum speed can be lower easily if needed to cater to the capabilities of the ultrasonic sensor. Its scale makes it large enough for the incorporation of multiple sensors onto the car body. Its compactness makes it easily demonstrable in an enclosed space. The simulated CA system takes place in a controlled environment, hence making results obtained much more repeatable. The RC car is by far the most affordable mobile platform to incorporate the CA system into. This is its main attractive feature.

C. *Integration of sensor systems onto test bed platform*

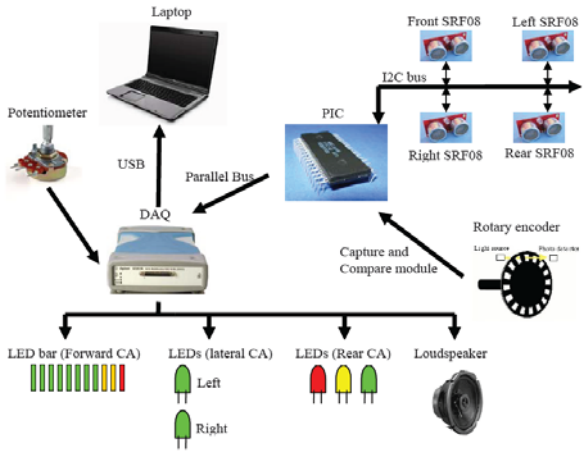

Fig. 2. Schematic diagram for final integrated test bed

The final integrated test bed is depicted in Fig 2. The 'Front SRF08' ultrasonic sensor shown in Fig 2 is placed at the front bumper of the RC car while the rotary encoder is embedded into the rear axle of the left wheel. The SRF08 ultrasonic sensor is connected via the I^2C bus to a PIC 16F877 microcontroller. The focus of the project was on the delivery of a working concept of an emulated CA system. Hence, due to the limited time available, the optimization of usage of the hardware components via the implementation of the I^2C bus on the DAQ was not carried out. The PIC is used as it comes with a built in I^2C bus controller, making data communication more reliable and robust. Periodic ranging operations are performed by the SRF08 sensor and the time between these ranging operations measured. This enables the calculation of the relative velocity of the leading vehicle.

An optical rotary encoder is used. Rotation of the encoder intermittently blocks the infra red (IR) beam between an IR transmitter and receiver pair, producing a TTL periodic square-wave ranging from 0V to 5V. The period of the waveform is measured by the timer in the PIC's capture and compare module (CCP). From the frequency of the square-wave, the speed of the host vehicle can be calculated.

The PIC obtains the raw values of the lead vehicle range and host vehicle speed and sends these values in batches to the DAQ via a uni-directional parallel bus. The parallel bus is 8 bits wide and comes with 2 control lines. The DAQ sends the obtained data via the USB cable to the laptop running VEE pro. The implemented CA algorithm in VEE calculates the warning parameter w from the raw values of lead vehicle range and host vehicle speed. Visual and audio feedback is given to the driver in the form of a graduated LED bar and loudspeaker respectively. The intensity of the visual and audio warnings given is varied according to the level of danger as calculated by w. These visual and audio feedback systems are controlled by the analog output channels of the DAQ.

III. CA Algorithms Analysis and Implementation

A number of CA algorithms have been developed. A major challenge in this area is to give ample warnings which do not serve to de-sensitize the driver. Most of these algorithms are derived from kinematics equations and involve the issuing of a warning should the range R be less than the warning range R_w. These algorithms also come with an overriding range R_o. If the range R is less than R_o, the CA system automatically applies the brakes. Two such algorithms will be considered here – the Mazda and Honda algorithm.

A. The Mazda Algorithm

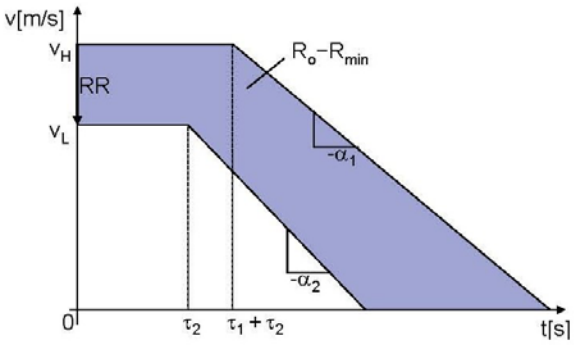

Fig. 3. Velocity time graph for the Mazda algorithm [5]

The Mazda algorithm is a conservative algorithm which takes into account the hypothetical worst case scenario. Fig 3 shows the velocity time plots for the lead and host vehicle. Initially, the host and lead vehicles begin with the velocity v_H and v_L respectively. The lead vehicle starts to brake after a time delay of τ_2 at the deceleration level of α_2, while the host vehicle only starts to brake after an additional time delay of τ_1, and at the deceleration level of. α_1. Both vehicles continue exhibiting these characteristics until they come to a complete standstill with their bumper to bumper distance being R_{min}. The shaded area shown in Fig 3 would be $R_0 - R_{\min}$, where R_o is the overriding range. R_o is given by the equation below [6]:

$$R_0 = v_H \cdot \tau_1 - RR \cdot \tau_2 + \frac{v^2{}_H}{2\alpha_1} - \frac{v^2{}_L}{2\alpha_2} + R_{\min} \qquad (1)$$

RR here refers to the ranging rate or relative velocity between the host and lead vehicle and is given by $v_L - v_H$. The system provides warning when the range R becomes less than the warning range R_w but greater than the overriding range R_o, where $R_w = R_o + \varepsilon$. Automatic braking is applied when $R < R_o$. A plot of R_o against $-RR$ and host vehicle velocity is shown in Fig 4. Fig 4 is obtained with the use of the following parameters: $\alpha_1 = 6\text{m/s}^2$, $\alpha_2 = 8\text{m/s}^2$, $\tau_1 = 0.1\text{s}$, $\tau_2 = 0.6\text{s}$, $R_{min} = 5\text{m}$. In Fig 4, the critical headway distance d_{br} refers to the overriding range R_o, while the relative velocity v_{rel} is equal to $-RR$. Only the areas of

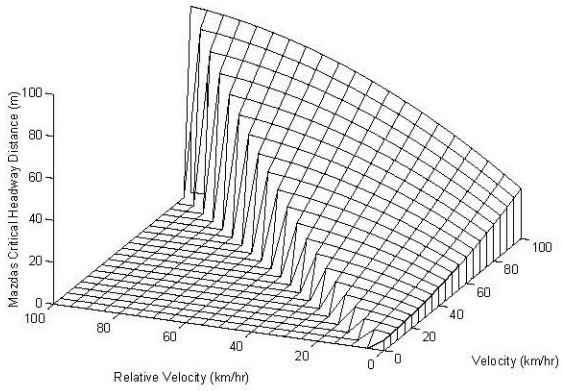

Fig. 4. Mazda critical headway distance versus relative velocity and host velocity (referred to here as 'velocity'). [7]

interest are plotted. When $v_{rel} > v_H$, the lead vehicle velocity is less than zero. This means that the lead vehicle is moving in the opposite direction. Since we take that to mean the detection of an oncoming vehicle in the opposite lane, such a value is ignored and a warning is not given, thus d_{br} is set to 0. When $v_L > v_H$, d_{br} and v_{rel} become negative. This means that the host vehicle can be "in front" of the lead vehicle and when the brakes are applied, still come to a stop at a distance of R_{min} from the lead vehicle, making the plot not meaningful. Since we are only considering the case whereby the host vehicle moves along a given trajectory in one given direction, v_H would be always greater or equal than 0.

B. The Honda Algorithm

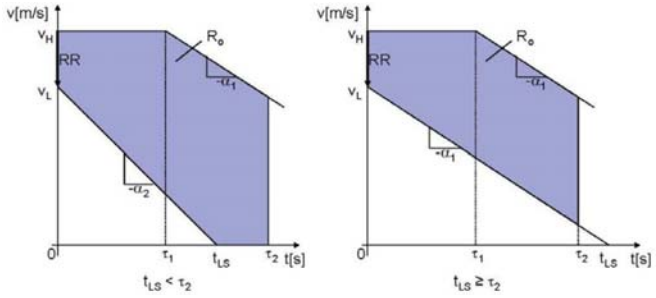

Fig. 5. Velocity time graph for the Honda algorithm [5]

The Honda algorithm is a less conservative algorithm which does not intend to avoid all collisions. Fig 5 shows the velocity time plots for the lead and host vehicle. The algorithm uses 2 scenarios depending on whether the lead vehicle stops before or after the time delay τ_2. It is assumed that the lead vehicle brakes constantly and has a deceleration of α_2 (if the lead vehicle stopping time $t_{LS} = {v_L}/{\alpha_2} < \tau_2$) or α_1 (if $t_{LS} \geq \tau_2$). The host vehicle continues traveling at its prevailing speed until a time delay of τ_1, at which it brakes constantly with a deceleration of α_1. The overriding range

R_o of these 2 scenarios is the minimum range needed to avoid collisions until τ_2 in both situations. It is given by the shaded area in Fig 5.

The equations for the 2 scenarios are given as follows [8]:

$$R_0 = \tau_2 \cdot v_{rel} + \tau_1 \cdot \tau_2 \cdot \alpha_1 - 0.5 \cdot \alpha_1 \tau_1^2 \qquad t_{LS} \geq \tau_2 \quad (2)$$

$$R_0 = \tau_2 \cdot v_H - 0.5 \cdot \alpha_1 (\tau_2 - \tau_1)^2 - \frac{v_L^2}{2 \cdot \alpha_2} \qquad t_{LS} < \tau_2 \quad (3)$$

The symbols used in Fig 5 and equations 2 and 3 are similar to those used in the Mazda algorithm unless otherwise stated. The system gives a warning when the range R becomes less than the warning range R_w and applies the brakes automatically when R is less than R_o. The warning range is given as:

$$R_w = -2.2 \cdot RR + 6.2 \qquad (4)$$

A plot of R_o against $-RR$ and host vehicle velocity is shown in Fig 6. Fig 6 is obtained with the use of the following parameters: $\alpha_1 = \alpha_2 = 7.8 m/s^2$, $\tau_1 = 0.5s$, $\tau_2 = 1.5s$. In Fig 6, the critical headway distance d_{br} refers to the overriding range R_o, while the relative velocity v_{rel} is equal to $-RR$.

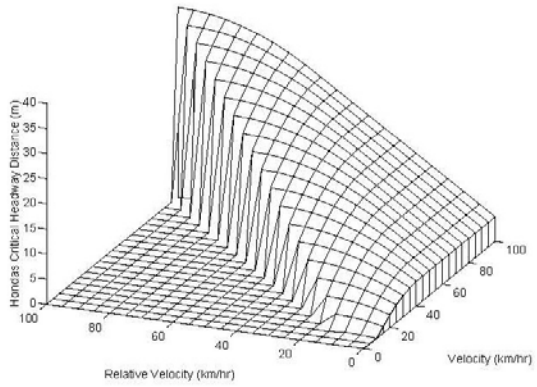

Fig. 6. Honda critical headway distance versus relative velocity and host velocity (referred to here as 'velocity'). [7]

It can be seen from comparing Fig 5 and Fig 6 that Honda's algorithm gives a smaller R_o, as compared to Mazda's algorithm.

C. Implemented Algorithm

The proposed CA algorithm does not have an overriding feature i.e. the brakes of the RC car will not be automatically applied under any circumstance. This is due to concerns of overriding systems interfering with driving during critical situations, hence compounding risks involved instead of contributing to vehicular safety. However, it shares similar concepts with both the Mazda and Honda algorithms. The implemented algorithm uses a conservative R_w and a non-conservative R_o to provide a fine balance between giving an adequate range of warnings while preventing the driver from becoming de-sensitized. The algorithm is based on a non-dimensional warning parameter w, where

$$w = \frac{R - R_0}{R_w - R_0} \qquad (5)$$

R is the bumper to bumper range between the host and lead vehicles; R_w is the warning range and R_o is the critical range. If $w > 1$, the lead vehicle is beyond the warning range i.e. $R > R_w$, no warnings of any sort are given. If $a < w < 1$, where a is the audio warning parameter and $a > 0$, this means that $R < R_w$, but $R > R_o$. As the value of w decreases from 1 to a, the intensity of visual warnings given increases. This is represented by a graduated LED bar shown in Fig 2. When $w < a$, auditory feedback via the loudspeaker is given on top of the visual display.

The equations for R_w and R_o are modified forms of the Mazda and Honda algorithms respectively.

$$R_w = \frac{1}{2 \cdot \alpha}\left[v^2 - (v - v_{rel})^2\right] + \tau_A \cdot v + \tau_B \cdot v_{rel}' + R_{\min} \quad (6)$$

$$R_0 = \frac{1}{2 \cdot \alpha}\left[v^2 - (v - v_{rel})^2\right] + \tau_C \cdot v \qquad (7)$$

Where v is the host vehicle velocity, $v_{rel} = -RR$. The host and lead vehicle decelerations are assumed to be α for all cases. $\alpha = 2.16 m/s^2$; $R_{min} = 0.04m$; $v_{rel}' = |v_{rel}|$; $\tau_A = \tau_{hum} + \tau_{sys}$; $\tau_B = \tau_{hum}$; $\tau_{hum} = 0.5s$. The value of τ_{sys} is the system delay calculated within VEE. For the Honda algorithm, only the case for $t_{LS} < \tau_2$ (equation 3) is used since the maximum value of t_{LS} given by v_{max} / α for the RC car is found to be 0.32s, which is less than $\tau_2 = 1.5s$. The velocity time graph of the modified Honda algorithm is shown in Fig 7.

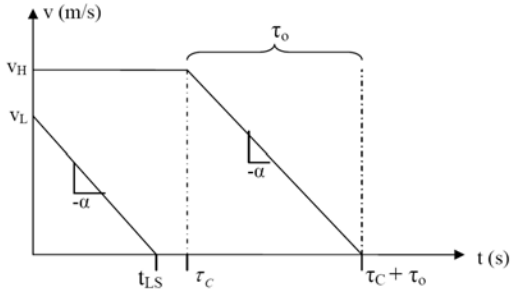

Fig. 7. Velocity time graph for the modified Honda algorithm

The delay time in the automated braking system τ_C is equivalent to τ_1 in Fig 5. $\tau_C = \tau_1 = 0.5s$. The maximum host vehicle stopping time if found to be $\tau_C + \tau_0 = 0.5 + {}^{v_{H(Max)}}\!/_{\alpha} = 0.85$. Since this is less than $\tau_2 = 1.5s$, the constant τ_2 is ignored in the formulation of equation 7. The critical range R_o is found by subtracting the area under the v_L line from the area under the v_H line in Fig 7.

In order for the implemented algorithm to work, R_o has to be less than R_w for all values of v_{rel} and v. This is to ensure that w only becomes negative when $R < R_o$.

IV. Software Flow Structure for Test Bed

A. Software flow structure for the PIC

The block diagram in Fig 8 shows the software flow structure implemented on the PIC microcontroller.

First the PIC input/output ports, capture and compare (CCP) module and I2C module are initialized and configured. The SRF08 sensor is initialized with a max

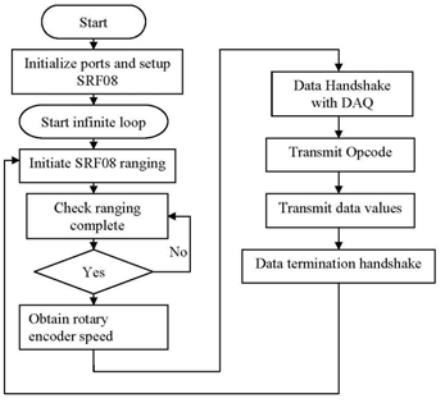

Fig. 8. Block diagram for PIC software flow structure.

analogue gain 94 (lowest setting possible) and a maximum range of 3 meters.

In the infinite loop, the range of the lead vehicle is obtained. Next, the speed of the host vehicle is calculated using the PIC CCP module. Finally, a data ready handshake is performed between the PIC and the DAQ to signal that the data is ready for transmission. The opcode for the transmission is then sent across to indicate the nature of the data being transferred. The data is then loaded byte by byte onto the parallel port. Once all data has been sent from the PIC to DAQ, a final data termination handshake is given to indicate the success of the data transfer and terminate the transmission.

B. Software flow structure for VEE pro

The block diagram in Fig 9 shows the software flow structure implemented in VEE pro. VEE pro directly controls all the functions for the DAQ.

First, 10 digital input/output ports are initialized for the 8-bit data and 2-bit control parallel port. Analogue output ports for the LED bar driver and loudspeaker are initialized. The timer of the DAQ is initialized and begins counting. Next the global variables used throughout the VEE pro program are initialized. In the infinite loop, the DAQ checks if the data batch containing the ranging information and host vehicle speed from the PIC is ready. If the data batch is not ready for transmission, past values of the host vehicle velocity and relative velocity (obtained from the data_matrix) are used to calculate the warning range, critical range and warning parameter. The appropriate level of visual and auditory warning is then given via the LED bar and loudspeaker outputs.

If the data batch is found to be ready, then a data level handshake is done with the PIC. The DAQ reads the first byte of sent data – the opcode. If the opcode is 11h, this indicates that the batch of data sent contains the ranging and speed values. If the opcode is not found to be 11h, then an 'opcode error' message is raised, indicating that there is error in the initial stage of the data transmission.

An opcode of 11h leads to the performance of a set instruction routine. The count value of the timer register is measured. The initial measurement of the timer value indicates the time it takes to reach from the start of the program to the present point. All subsequent measurements of the timer indicate the time t_{range} between successive reception of ranging and speed values from the PIC i.e. the ranging time. The value of t_{range} is equal to τ_{sys}. After measurement, the timer is cleared, reset and started again.

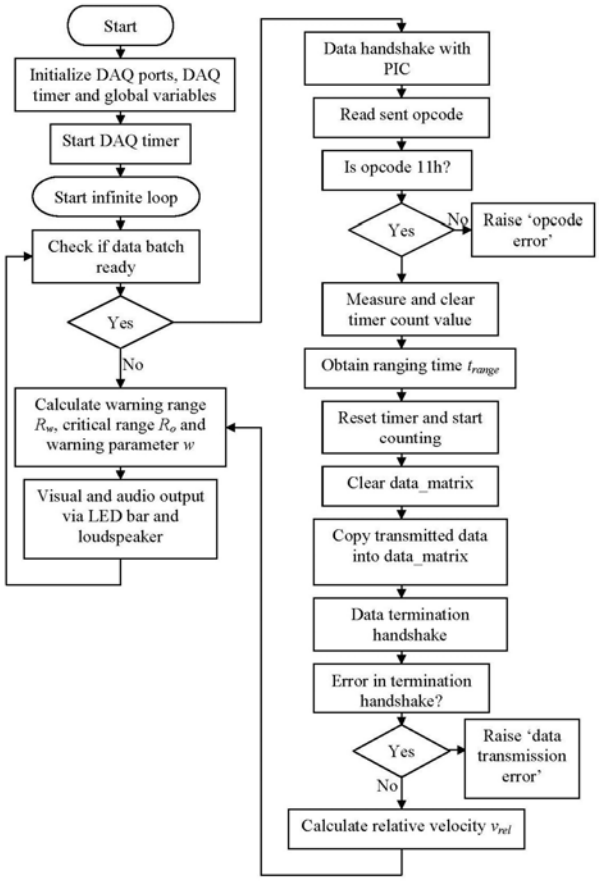

Fig. 9. Block diagram for VEE software flow structure.

Next, all values in the global variable data_matrix are set to 0. This would enable easier error detection should an error in data transmission occur. The data is then transmitted byte by byte over the parallel bus from the PIC to the DAQ and stored directly into the data_matrix. These data values contain the raw values of the range of the leading vehicle and the speed of the host. Completion of data transmission is indicated by a data termination handshake. If an error is detected in this handshake, then the 'data transmission error' message will be raised, indicating that errors have occurred during the course of the data

transmission.

Using the present and previously obtained value for the lead vehicle range ($R_{present}$ and R_{past} respectively), the relative velocity v_{rel} between the host and lead vehicle can be calculated as follows: $v_{rel} = (R_{present} - R_{past}) / \tau_{sys}$.

After calculating v_{rel} , the program loops back to calculate the new value of the warning range, critical range and warning parameter using these recently obtained values. The appropriate visual and audio feedback is output via the LED bar and loudspeaker.

V. Test Bed Optimization and Evaluation

The performance of the CA front end system was found to be severely dependant on the system delay τ_{sys}. This is directly dependant on the speed and amount of data transferred over the parallel bus. Optimizations to the system

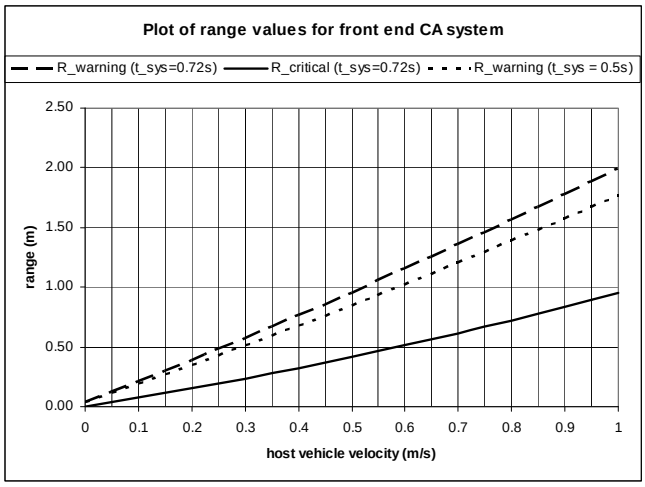

Fig. 10. Plot of R_w (R_warning) and R_o (R_critical) against v_{host}

have yielded a τ_{sys} of 0.63 to 0.72 seconds. Taking the worst case scenario where τ_{sys} is always 0.72 seconds, the values of R_w and R_o needed for CA system to function effectively can be calculated. A plot of R_w, R_o (for τ_{sys}=0.72s) and R_w (for τ_{sys}=0.5s) against v_{host} is shown in Fig 10. A τ_{sys} of 0.5s represents the optimal system delay.

High accuracy in the range readings obtained from the SRF08 is desired. An optimal accuracy ranging from 85% to 90% was obtained for a maximum range of 1.5m. For τ_{sys}=0.72s the corresponding maximum host vehicle velocity is found to be 0.77m/s from Fig 10.

The rotary encoder was optimized to yield detection of the lowest possible speed. By reducing the timer clock input frequency and increasing the resolution of the hardware, the minimum speed detectable was reduced to 0.127m/s.

The above findings add considerable constraints to the implemented test bed. Although the maximum cruising speed of the RC car is 2.22m/s, reliable performance of the system will only be achieved when the speed ranges from 0.127m/s to 0.77m/s. As mentioned earlier, 75% of all crashes occur at speeds lower than 32km/h [1]. If we take the maximum cruising speed for automobiles to be 90km/h, this would mean that 75% of crashes can be avoided if the RC car speed is lower than 0.79m/s. Hence, although the implemented system cannot avoid high speed crashes, it can remain effective at low speeds.

The main constraint arises from the choice of sensor for the front end CA system. The limited range which can be achieved by the ultrasonic sensor due to the unreliability of longer range readings coupled with the time delays in-between ranging operations have directly imposed the above mentioned system constraints.

Improvements to the system can be made by introducing several major factors which the present system does not take into account. These include the acceleration and tire-road coefficients of the RC car. The latter would be an important indicator of the road conditions. These factors play an important role in many of the automobile CA systems which are currently in research.

VI. Conclusion

Although the functionality of the test bed is limited by the constraints mentioned, the test bed effectively implements a CA system using Agilent's U2353A DAQ and VEE pro software within the set constraints. This successfully demonstrates the capabilities of Agilent's DAQ and VEE pro in the development of applications for CA systems.

Acknowledgment

The author would like to thank Assistant Professor Ha Yajun, Assistant Professor Le Minh Thinh and the services and support rendered by Agilent Technologies for their advice and guidance.

References

[1] British Motor Insurance Repair Research Centre (ThatCham), "The car we couldn't crash", vol 3 no.2, February 2008

[2] Auto Electronics market to exceed US$50 Billion by 2010: http://www.instat.com/press.asp?ID=1752&sku=IN0603375RE

[3] Yi Kyongsu, Woo Minsoo, Kim Sung Ha, Lee Seong-chul, "Experimental investigation of a CW/CA System for automobiles using Hardware-in-the-loop simulations", Proceedings of the American Control Conference, volume 1, p 724-728 (1999)

[4] An-Ping Wang, Jie-Chang Chen, Pao-Lo Hsu, "Intelligent CAN based automotive collision avoidance warning system", Proceedings of the 2004 IEEE International Conference on Networking, Sensing and control, volume 1, p 146-151 (2004)

[5] Yizhen Zhang, "Engineering Design Synthesis of Sensor and Control Systems for Intelligent Vehicles", California Institute of Technology (2006)

[6] A. Doi, T. Butsuen, T. Niibe, T.Yakagi, Y. Yamamoto, and H. Seni. "Development of a Rear-End Collision Avoidance System with Automatic Braking control", JSAE Review, vol. 15, p335-340, October 1994

[7] Peter Seiler, Bongsob Song, J. Karl Hendrick, "Development of a collision avoidance system", *SAE Special Publications, ITS Advanced Controls and Vehicle Navigation Systems,* **volume 1332,** p97-103 (1998)

[8] Y. Fujita, K. Akuzawa, and M. Sato, "Radar Brake System", Annual meeting of ITS America, vol. 1 p95-101, March 1995