Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A CAN-based Distributed Control System for Autonomous All-Terrain Vehicle (ATV) ⋆

Woonhyuk Baek * **Seyong Jang** ** **Hoin Song** * **Soontae Kim** **
Bongsob Song * **Dongkyoung Chwa** **

* *Department of Mechanical Engineering, Ajou University, Suwon, Korea, (e-mail:may, ajoutech, bsong@ajou.ac.kr).*
** *Department of Electronic Engineering, Ajou University, Suwon, Korea, (e-mail:seyong, saxomaster, dkchwa@ajou.ac.kr).*

Abstract: This paper presents development of a longitudinal controller for an autonomous All-Terrain Vehicle (ATV). The developed ATV is a Controller Area Network (CAN) based distributed control system including multiple processors. Before developing the longitudinal controller, it is shown that the worst case response time of messages via CAN is bounded by appropriate assignment of priorities to all messages. Then, a control model for longitudinal control of ATV is proposed and validated experimentally. Finally, the longitudinal controller for ATV, based on a nonlinear control technique so-called Dynamic Surface Control (DSC), is designed and validated via simulation whether it can compensate for the worst case time delay and packet loss resulting from CAN communications.

NOMENCLATURE

C_a Aerodynamic drag coefficient
F_r Rolling resistance force
F_a Aerodynamic drag force
J_e Moment of inertia of engine
J_w Moment of inertia of axle and wheel
R_g Effective gear ratio
T_b Brake torque
T_e Engine torque
v Longitudinal velocity of ATV
m Total weight of ATV
h Effective wheel radius
α Angle of throttle control motor
β Angle of Brake control motor
μ Rolling resistance coefficient
ω_w Angular velocity of wheel
ω_e Angular velocity of engine

1. INTRODUCTION

Single bus based distributed systems have been developed in many real time control applications such as automobiles, aircrafts, and industrial automation(Farsi et al. [1999]). The single bus based on multiplexing network allows sharing information among various intelligent processors in the framework of multi-master systems. Among single bus network technologies, Controller Area Network (CAN) was developed as in-vehicle network by Bosch in 1980s and has been applied to not only vehicles but also other many distributed systems.

While there was a great success in the CAN-based distributed control system of automotive vehicles, one of the commonly perceived problems with CAN is its inability to bound the response time of messages in a real-time fashion, especially for lower priority messages (Heffernan and Bohannon [2001]). To solve the problem, two approaches have been considered in the literature: one is to improve software of a CAN application layer, e.g., intelligent scheduling (Audsley et al. [1993], Heffernan and Bohannon [2001]), dynamic ID allocation (Baek et al. [2006]). The other is to compensate for time delay of CAN using networked control system technologies, e.g., stability for networked control (Zhang et al. [2001]) and observer-based compensation (Luck and Ray [1991], Seiler [2001]).

The ultimate objective of autonomous ATV mentioned in this paper is to serve it as an agent in multi agent systems for environment monitoring. While the development of the autonomous ATV has not been completed and the studies have continued, two preliminary results are introduced in this paper. One is to describe how a CAN-based distributed control system using multiple Digital Signal Processors (DSP) is designed and developed for the autonomous ATV. Furthermore, it will be shown that the worst case response time of messages via CAN is bounded under appropriate assignment of priorities to all messages. The other is to design a longitudinal controller for ATV. The controller is based on a nonlinear control technique so-called Dynamic Surface Control (DSC), and will be validated via simulation whether it can compensate for the worst case time delay and packet loss which may occur via CAN.

2. DISTRIBUTED CONTROL SYSTEM

2.1 Hardware Layout

To develop the autonomous ATV, additional hardware such as sensors, actuators, and processors have been im-

⋆ This work was supported by grant No. R01-2006-000-11373-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

Table 1. Hardware for the ATV system

Type	Model	Manufacturer
Incremental Encoder	TRD-SH1024B	Koyo Electronics
Absolute Encoder	TRD-NA1024NW	Koyo Electronics
IMU	MTi	Xsense
DC Motor	IG-42GM	D & J Corp.
Laser Scanner	LMS291-S05	SICK AG
PC/104 CPU board	MOPSlcd7	KONTRON
PC/104 CAN board	CAN-AC2-104	SOFTING AG
DSP	TMS320F2811	Texas Instruments
GPS	VBOX III	Racelogic

Fig. 1. Hardware layout for autonomous ATV

plemented as shown in Fig. 1. In the case of longitudinal control, rotary encoders are used to measure engine speed and wheel speed. Also, a laser scanner is implemented to detect obstacles, Inertial Measurement Unit (IMU) is used to measure motion status of ATV, and absolute position and velocity can be obtained from a Global Positioning System (GPS). Finally, two DC motors and rotary encoders are implemented to control throttle and brake. For more detailed specification, you may refer to Table 1.

If the control system is centralized, i.e., all data and information are transmitted to single main processor, there are more possibility that rate of data loss and time delay increase due to limits of computation capability of processor and/or data communication (Kanakina and Tobagi [1987], Elliott and Boucher [1994]). Therefore, as shown in Fig. 2, processors using seven DSPs are distributed for our system. For instance, DSP 2 is connecting with both throttle and brake control motors, and DSP 3 is with engine and wheel speed sensors. This distributed configuration allows us to process measurement data from sensors independently. Furthermore, the DSPs transfer either processed data or measurement data via CAN bus and share information among them. It is noted that PC104 is connected and used to acquire all data from CAN bus for longitudinal control later.

2.2 Software Structure

As shown in Fig. 1, the used processors are classified into twofold: one is to process the data from sensors and the other is either to calculate control input or to monitor faults in the system. In the case of longitudinal control as shown in Fig. 3(a), the software structure is divided into three layers. In the interface layer, event triggered

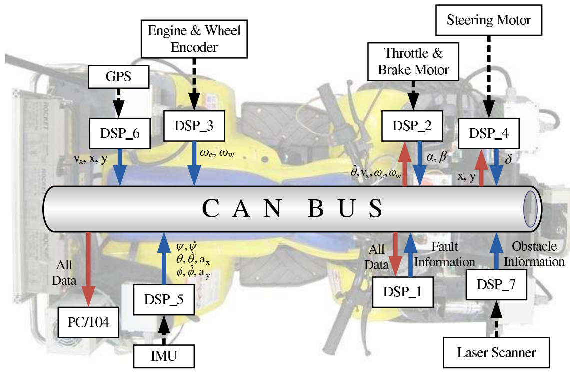

Fig. 2. Layout of distributed processors

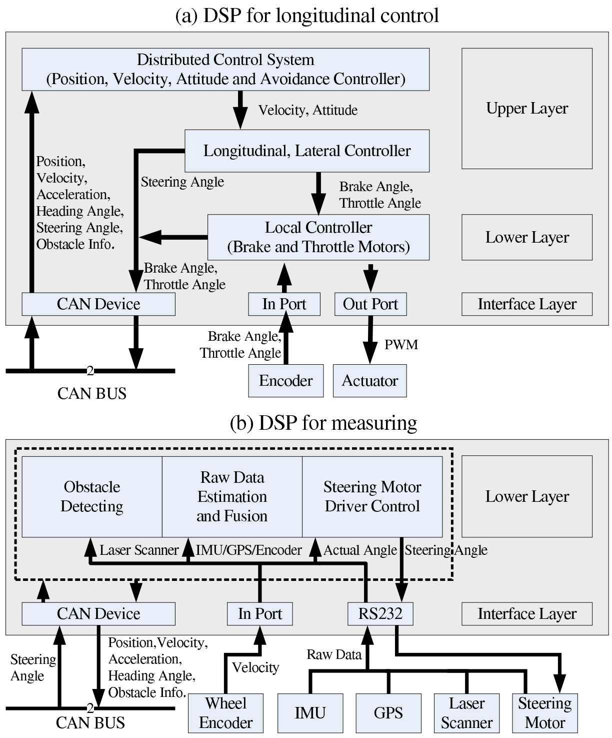

Fig. 3. Software structures of processors

data from CAN bus and analog data are stored to each variables. In the lower layer, the position controllers for both throttle and brake control motors are designed. Finally, in the upper layer, the longitudinal controller calculates the desired angles of throttle and brake control motors to track the desired velocity and/or position using the variables coming from interface layer. Furthermore, the desired position or trajectory is generated based on other information from CAN bus such as obstacle information and/or GPS information.

In the case of data acquisition as shown in Fig. 3(b), the software structure to get information from sensors is divided into two layers. First, the data from sensors are obtained via RS232 communication and the analog data are stored to variables in the interface layer. In the lower layer, some level of signal processing such as obstacle detection, data fusion, and filtering is performed. Then, the processed information are transmitted every transmission time via interface layer or CAN.

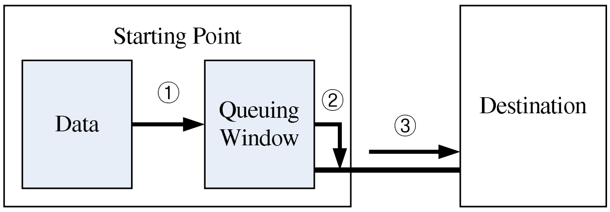

Fig. 4. Process of transmission

2.3 Analysis of CAN Bus

The CAN is a serial communication protocol, which efficiently supports distributed real-time control with high level of security, and is a carrier sense broadcast bus where a number of processor are connected to the bus via an interface (Tindell et al. [1995], Corrigan. [2002]). That is, if more than one processor tries to transmit data (or messages) at the same time, then all processors detect this, wait for a determined time period, and try again until the bus is idle. Therefore, in order to avoid collision of messages in a more systematic approach, the priority of a message is determined by an unique arbitration ID so-called *identifier*.

If two or more messages approach at the same time and see a dominant bit of identifier on the bus then that message transmission is stop and wait the idle state of bus. While there has been much work about the analysis of CAN model, we use a simple CAN model with fixed priorities (Tindell et al. [1995]). As shown in Fig. 4, there are three processes in the CAN model. First, a message is queued into the queuing window by processor. Second, the processor checks the state of CAN bus. If there is a message using the bus, the queued message wait until the state is idle. Finally, the queued message is transmitted. The time delays of each process are denoted jitter(J_m), queuing delay(t_m) which include J_m, transmission delay(C_m).

Now, we summarize the analysis of simple CAN model proposed by Tindell et al. [1995]. The worst-case of response time is the longest time gap between the queuing of a message and the arrival at destination of a message. R_m is made of two delays as

$$R_m = t_m + C_m$$

C_m is a function of the number of bytes in a message. When the baud rate is 1Mbps and 8bytes of the message is sent, the C_m is about 130μs (Tindell et al. [1995]). t_m is made of two terms; blocking time and $interference$. And it is defined as

$$t_m = B + \sum_{\forall j \in hp(m)} \left\lceil \frac{t_m + J_j + \tau_{bit}}{T_j} \right\rceil C_j$$

where τ_{bit} is the time taken to transmit a bit on CAN, B is the blocking time, $hp(m)$ is the set composed of all the messages in the system of higher priority than message m, and T_j is the period of a given message j.

As shown in Fig. 5, if the worst-case response time is occurred at $(k{+}1)$-th transmission (i.e., $t_m(k) < t_m(k{+}1)$) then the interval of two arriving times is the longest (i.e., $T_m < \mathrm{dt}(k)$). In this case, if we assume that τ_{bit}=8 byte, T_j=10 ms for all j, C_j=130 μs for all j, and B is assumed to be constant in the same processor and transmission

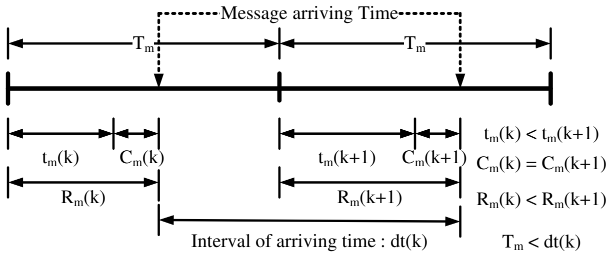

Fig. 5. Definition of delay

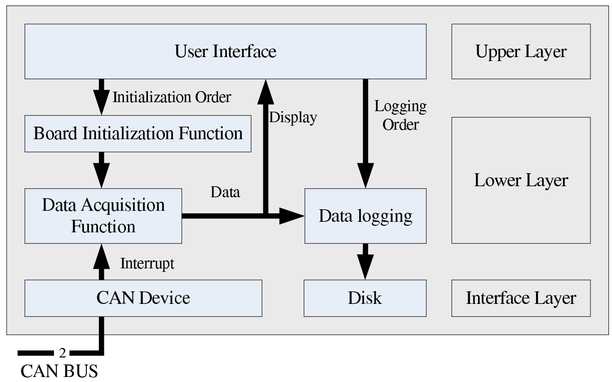

Fig. 6. Software structure of monitoring program

function, then the worst-case response time is determined by the priority of each message.

As shown in Fig. 6, the monitoring software is developed to obtain the maximum interval of arriving time. It is divided into three layers. The upper layer interfaces between user and PC104, i.e. Human Machine Interface (HMI), and the lower layer acquires data delivered by CAN and it allows us to analyze both time delay and packet loss. For instance, Table 2 shows test results obtained through analysis of about 60,000 messages with respect to 13 different messages when the baudrate is set as 1 Mbps. The subscript of M_1-M_{13} represents an identifier, M_1 is the lowest identifier (i.e., the highest priority message) and M_{13} is the highest identifier (i.e., the lowest priority message). In Table 2, the column of $\mathrm{dt}_{\mathrm{ave}}$ shows that the average interval is close to 10 ms. It is noted that M_3-M_6 are the obstacle information coming from the laser scanner with about 293 ms of sampling time. Due to its own hardware characteristics, the minimum sampling time can be modified arbitrarily. Therefore, based on Table 2, it can be estimated that the maximum time delay is 1.114 ms and the maximum rate of packet-loss is 0.015 % for the given CAN system.

3. CONTROL MODEL AND ITS VALIDATION

Only a longitudinal control model will be discussed in this paper and the proposed model will be validated through experimental driving tests.

3.1 Longitudinal Control Model

Using the Newton's 2nd law, a longitudinal control model for autonomous ATV can be derived as follows (Gerdes [1996])

Table 2. Analysis results of CAN messages

No.	Message	dtave(ms)	dtmax(ms)	loss(%)
M_1	$\alpha,\ \beta$	10.00022	10.142	0.015
M_2	$\omega_e,\ \omega_w$	9.99982	10.125	0
M_3	obstacle info.	293.269	293.552	0
M_4	obstacle info.	293.202	293.571	0
M_5	obstacle info.	293.216	293.610	0
M_6	obstacle info.	293.273	293.576	0
M_7	x	9.99970	10.962	0
M_8	$v,\ y$	10.00035	11.053	0.005
M_9	$\dot{v}$	10.00076	11.109	0.015
M_{10}	$\psi,\ \dot{\psi}$	10.00006	10.374	0.002
M_{11}	$\dot{\theta},\ a_x$	10.00011	11.114	0.002
M_{12}	$\dot{\phi},\ a_y$	10.00018	10.954	0.002
M_{13}	$\theta,\ \phi$	10.00018	10.777	0.005

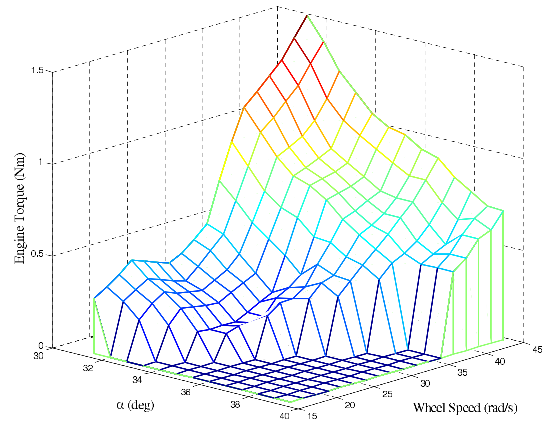

Fig. 7. Empirical map of engine torque

$$J_{eq} \cdot \dot{v} = \frac{T_e(\alpha, \omega_w)}{R_g} - T_b(\beta) - h \cdot F_f \tag{1}$$

where J_{eq} and F_f can be calculated as following equations

$$J_{eq} = \frac{J_e + R_g^2(J_w + m \cdot h^2)}{R_g^2 \cdot h}$$
$$F_f = F_a + F_r. \tag{2}$$

Furthermore, T_e is a function of α and ω_w and can be obtained empirically as shown in Fig. 7, and T_b is assumed to be a empirical function of β. It is noted that the empirical engine map is obtained through experimental driving tests because the engine map data haven't been provided by the manufacturer.

Next, the rolling resistance and aero dynamics resistance force in (2) can be described as

$$F_r = \mu m g, \quad F_a = \frac{1}{2} C_a v^2. \tag{3}$$

The coefficients μ and C_a are estimated by minimizing the error between the measured velocity and one calculating from the proposed ATV model under the driving scenario of no-braking slowdown. When the coefficients are chosen as $\mu = 0.02$, $C_a = 0.5487$, the maximum error between estimated and measured velocity is about ± 5 % (See in Fig. 8).

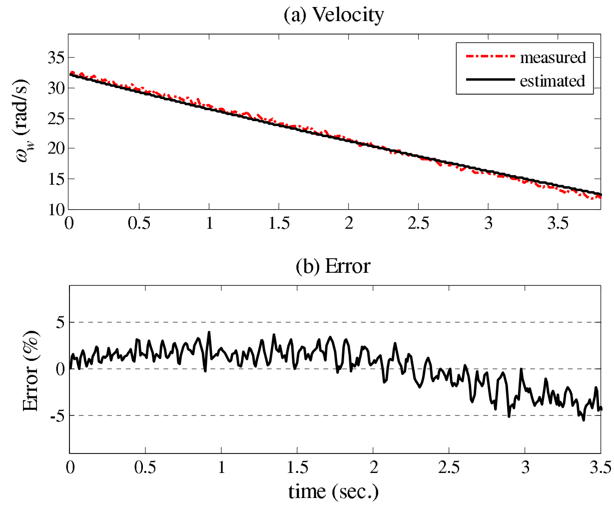

Fig. 8. Estimation of F_r and F_a via experiments

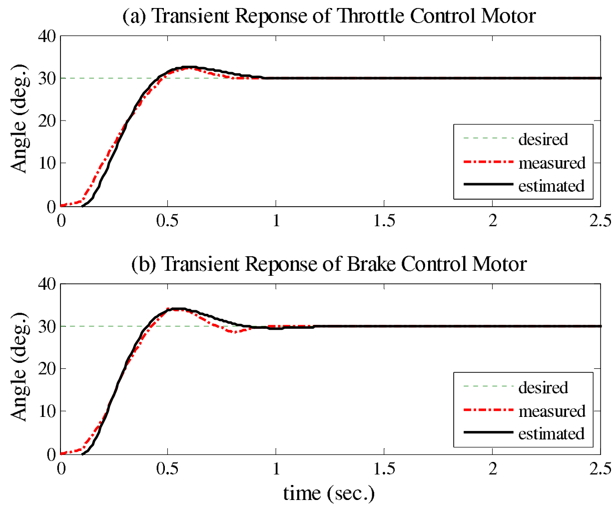

Fig. 9. Transient response of throttle and brake control motors

3.2 *Position Controller*

As shown in Fig. 3(a), two position controllers to track the desired angle positions of throttle and brake control motors are necessary to be designed. If PID control is used, the time responses of two motor angles are shown in Fig. 9. If both PID controllers and motors are assumed as actuators, they are approximated as a second order system to reduce complexity respectively. For instance, once PI controllers are determined, throttle and brake control motors can be modeled as follows:

$$\ddot{\alpha} + 9.32\dot{\alpha} + 64.16\alpha = 64.16\alpha_{in}(t - 0.1)$$
$$\ddot{\beta} + 8.989\dot{\beta} + 69.3\beta = 69.3\beta_{in}(t - 0.1) \tag{4}$$

where α_{in} and β_{in} are the command angles to position controllers of throttle and brake control motors, respectively.

3.3 *Experimental Validation of a Longitudinal Control Model*

To validate the longitudinal control model experimentally, velocity calculating from the control model will be com-

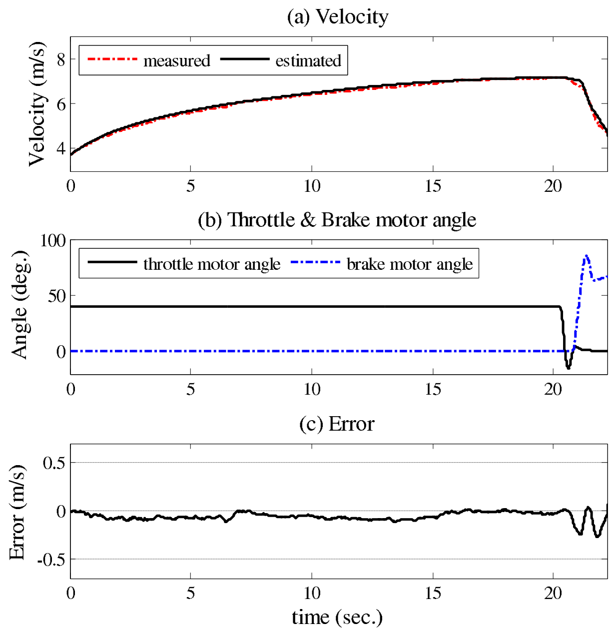

Fig. 10. Experimental validation of a control model

pared with the measured one under given throttle and brake angle. As shown in Fig. 10(b), the driving scenario is given as follows: an angle of throttle control motor is given as 40 *deg* up to 20 second, then the angle of brake control motor is changed to 70 *deg*. Fig. 10(a) shows that the velocity increases up to 7 m/s and two velocity responses (measured and estimated) are compared each other. For the given driving scenarios, the estimation error is within ± 0.3 m/s in a term of velocity (see in Fig. 10(c)).

4. DESIGN OF A LONGITUDINAL CONTROLLER

In this section, a longitudinal controller is designed based on the experimentally validated ATV model. Then it will be validated via simulation whether the designed controller can compensate for both time delay and packet loss, which in general occurs through CAN communications.

4.1 Longitudinal Control for Speed Following

One of nonlinear control techniques called Dynamic Surface Control is applied to design a longitudinal controller for speed following. It was first applied to longitudinal control for an autonomous passenger vehicle (Gerdes [1996]). It can be extended and applied for longitudinal control of ATV due to their similarities. Therefore, a concise mathematical derivation of the controller will be described here.

Suppose the desired velocity is v_{des}. Then, the error surface S_1 is defined as $S_1 \equiv v - v_{des}$. After differentiating S_1 and using (1)

$$\dot{S}_1 = \dot{v} - \dot{v}_{des} = \frac{1}{J_{eq}} \left(\frac{T_e}{R_g} - T_b - h \cdot F_f \right) - \dot{v}_{des}, \qquad (5)$$

the desired engine torque for converging S_1 to arbitrary boundary is calculated as follows

$$T_{e,des} = R_g \left\{ J_{eq} \left(\dot{v}_{des} - \lambda_{1e} \cdot S_1 \right) + h \cdot F_f \right\} \qquad (6)$$

Table 3. Control model parameters

Parameter	Value	Parameter	Value
h	0.205	m	183
μ	0.02	C_a	0.5487
J_e	0.00236	J_w	0.25234

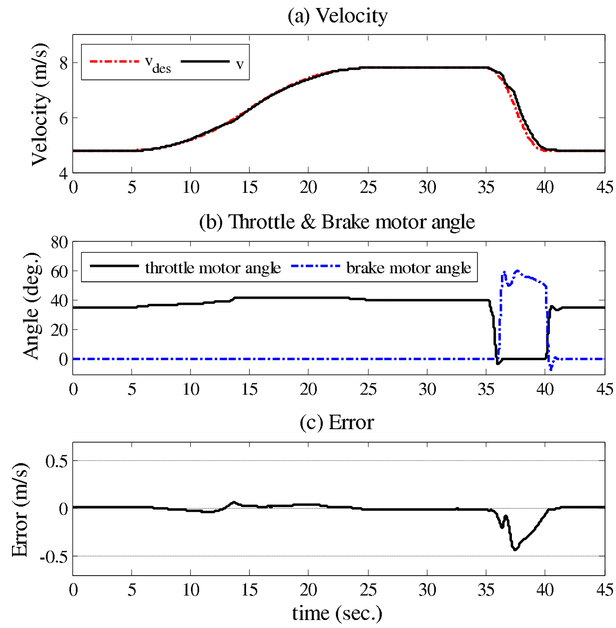

Fig. 11. Performance of a proposed controller

where λ_{1e} is the controller gain for engine control. Then, the desired angle of throttle control motor can be derived using an inverse function of $T_e(\alpha, \omega_w)$ in (1).

Similarly, the desired brake torque is calculated as follows

$$T_{b,des} = -\left\{ J_{eq} \left(\dot{v}_{des} - \lambda_{1b} \cdot S_1 \right) + h \cdot F_f \right\} \qquad (7)$$

where λ_{1b} is the controller gain for brake control. Finally, the desired angle of a brake control motor can be calculated using a heuristic relationship between the angle of the brake control motor and brake torque. More detailed theory about stability and synthesis problems can be referred to Song et al. [2002].

4.2 Simulation Result

The performance of the longitudinal controller is validated via simulations based on an experimentally validated control model, and the corresponding model parameters are listed in Table 3. Also, the packet loss and time delay models are added to simulate CAN based on the analysis in section 2.3. That is, it is assumed that the packet loss rate is 1% and maximum time delay is 2 *ms*.

When the desired velocity is changing from 4.8 m/s to 7.8 m/s, the velocity tracking error is within 0.4 m/s (see in Fig. 11). It can be concluded that the proposed controller is robust enough to compensate for packet loss and time delay whose model is obtained experimentally.

5. CONCLUSION

In this paper, the CAN-based distributed control system for autonomous ATV was presented. First, hardware layout and software structure was introduced and the performance of CAN was analyzed in terms of time delay and packet loss rate. Second, the longitudinal control model was proposed and the corresponding model parameters were estimated by a least square method. Then, the control model was validated by comparing simulation results with experimental ones. Finally, a longitudinal controller based on DSC was designed and its performance was validated via simulations in the presence of packet loss and time delay due to CAN.

As a future work in the near term, the longitudinal controller will be validated via driving tests and a lateral controller will be implemented for fully autonomous ATV. Furthermore, the ATV will be served as an intelligent agent for an environment monitoring system based on multi agents.

REFERENCES

M. Farsi, K. Ratcliff, and M. Barbosa. An overview of controller area network. *Computing & Control Eng. Journal*, pages 113–120, 1999.

K. Tindell, A. Burns, and A. J. Wellings. Calculating controller area network (CAN) message response times. *Control Eng. Practice*, volume 3, no. 8, pages 1163–1169, 1995.

D. Heffernan and A. Bohannon. Real-time design in a distributed control network application layer environment. *IEE Proc. Software*, volume 148, no. 5, pages 149–1155, 2001.

N. Audsley, A. Burns, M. Richardson, K. Tindell and A. Wellings. Applying new scheduling theory to static priority preemptive scheduling. *Software Eng. Journal*, volume 8, no. 5, pages 284–292, 1993.

K. H. Baek, W. S. Lee, K. N. Ha, B. J. Son, S. Lee, and K. C. Lee. Improvement of transmission delay using dynamic ID allocation algorithm in CAN communication. *Proc. of KSAE Fall Conference*, 1492–1498, 2006.

W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control System Magazine*, pages 84–99, February 2001.

R. Luck and A. Ray. An obserber-based compensator for distributed delays. *Automatic*, volume 26, no. 5, pages 346–352, 2006.

P. J. Seiler. *Coordinated control of unmanned aerial vehicles.* Ph. D thesis, U. C. Berkeley, 2001.

H. R. Kanakina and F. A. Tobagi. On distributed computations with limited resources. *IEEE Trans. Computers*, volume C-36, no. 5, pages 517–528, 1987.

S. J. Elliott and C. C. Boucher. Interaction between multiple feedforward active control systems. *IEEE Trans. Speech and Audio Processing*, volume 2, no. 4, pages 521–530, 1994

S. Corrigan. *Introduction to the controller area network (CAN).* Application Report SLOA101, Texas Instruments, 2002.

J. C. Gerdes. *Decoupled design of robust controllers for nonlinear systems; as motivated by and applied to coordinated throttle and brake control for automated highways.* Ph. D. thesis, U. C. Berkeley. 1996.

B. Song, J. K. Hedrick, and A. Howell. Robust stabilization and ultimate boundedness of dynamic surface control systems via convex optimization. *International Journal of Control*, volume 75, no. 12, pages 870–881, 2002.