Analysis of Effects of Time-Delay in an Inverted Pendulum System Using the Controller Area Network

Sung-Min Cho* and Suk-Kyo Hong**

*Department of Electronics Engineering, Ajou University, Suwon 442-749, Korea
(Tel: +82-31-219-2489; Email: benedictcho@empal.com)
**Department of Electronics Engineering, Ajou University, Suwon 442-749, Korea
(Tel: +82-31-219-2489; Email: skhong@ajou.ac.kr)

Abstract: In this paper, the design of the network system using the CAN and the analysis of effects of time delay in the system are presented. A conventional implementation technique induces many problems because of the amount and complexity of wiring and maintenance problems. The network system reduces these problems, but it cause another problem; time delay. Time delay in a sampling time does not have much effects on the system, but time delay over the sampling time changes the control frequency and ended up makes the system unstable. It is verified that time delay between each parts has different effects on the entire system. The results from this paper will be a base for studying algorithms to reduce effects of time delay in the system using the CAN.

Keywords: Controller Area Network, CAN, Network System, Inverted Cart Pendulum, Time Delay

1. Introduction

Today's industrial control applications contain a large number of electronic devices and come to be more complex. With conventional implementation techniques, data is exchanged by means of dedicated signal lines, but this is becoming more difficult and expensive as control functions become ever more complex.[3]

Networking also known as multiplexing, is a method for data transferring in a distributed electronic devices via a serial data bus. Controller Area Network (CAN) is one of the well-known bus technologies. It provides some build-in features, such as high level data security, priority based media access control, short message length, error registration and indication.

The network system unavoidably has time delay during exchanging messages between each parts. Some works for time-delay systems like Fig.1 have been reported in the literature. Hörjel [5] used Bluetooth in real-time control loop and suggested control algorithms for compensating time delay. Xiaodong and Yanjun [6] proposed methods to calculate the message delivery time.

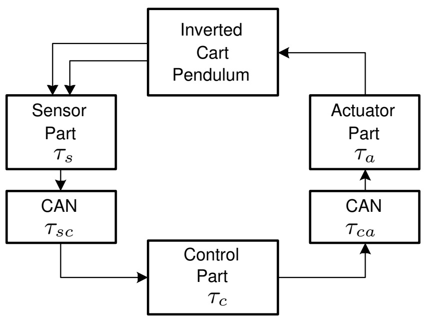

Fig. 1. The General Setup of the Control Application

Fig.1 shows the general setup of the control algorithm through the CAN and time delay occurring at each parts. The total delay is the total time it takes to complete one control loop. Here the round trip delay time can thus be described as[5]:

$$\tau = \tau_s + \tau_{sc} + \tau_c + \tau_{ca} + \tau_a \tag{1}$$

The main objective is to analyze effects of these time delay in an inverted pendulum system using the CAN.

In this paper, we verify it is possible to control an inverted pendulum system using the CAN at first. Next, the value of time delay in equation (1) is calculated by equation (2) and measured from the system. Last, we discuss about effects of time delay in the network system used in this paper.

2. Controller Area Network

CAN was originally developed in the 1980s for the interconnection of control components in automotive vehicles.[4] CAN enabled a huge reduction in wiring complexity and, additionally, made it possible to interconnect several devices using a single pair of wires, allowing data exchange between them at the same time. Although specifically conceived for the automotive industry, nowadays it is widely used in automation mainly due to the low price of CAN communication solutions. The basic features of CAN are:

- High-speed serial interface: CAN is configurable to operate from a few kilobits per second right up to 1Mbit/s transmission rates.
- Low-cost physical medium: CAN operates over a simple twisted wire pair, therefore cabling a CAN network is inexpensive compared to other cabling.
- Fast reaction times: The ability to transmit information without requiring a token or permission from a bus arbiter results in extremely fast reaction times.
- Multi-master : Using CAN it is simple to broadcast information to all or a subset of nodes on the bus.
- Error detection and correction: The high level of error detection and number of error detection mechanisms provided by the CAN hardware.

2.1. Communication modes and data exchange

CAN is a broadcast bus with priority base access to the medium and non-destructive collision resolution: two or more nodes try simultaneously to transmit a message, the lowest priority frames lose the contention. But, the highest priority message successfully destination without being destroyed by the collision. Nodes do not possess an address. A message contains and identifier, unique to the whole system, that serves two purposes:

- Assigning a priority for the transmission (the lower the numerical value of the identifier, the greater the priority during the arbitration phase) Allowing message filtering upon reception
- Data segmented in several frame may be transmitted periodically, sporadically or on-demand.

Due to the medium access technique (Priority-based Carrier Sense Multiple Access/Collision Detection), the maximum data rate that can be achieved essentially depends on the bus' length. For example, the maximum data rate for 30 and 500 meter buses are respectively 1Mbit/sec and 100kbit/sec

2.2. Format of a CAN telegram

The CAN message format uses 11-bits identifiers at an standard 2.0A format; however, an extended CAN format (2.0B format) uses 29-bit identifiers instead. CAN controllers supporting the extended format will in general also work with the standard format communication using 11-bit identifiers.

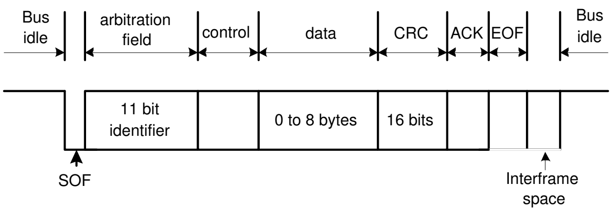

Fig. 2. Format of the Standard CAN Telegram

A message in the standard format begins with the start of frame (SOF). This followed by the arbitration field which contains the identifier of the CAN telegram and is used to arbitrate access to the bus. Also part of the arbitration field is the RTR bit (remote transmission request) which indicates whether the frame is a request frame (without any data, this type of message is used to trigger a transmission by another node) or a data frame.

The control field contains the IDE bit (identifier extension), which indicates whether the frame is a standard format frame or an extended one. Next comes the data field which can be from zero to eight bytes in length.

The acknowledge(ACK) field comprises an ACK slot bit and an ACK delimiter bit. The ACK slot is transmitted as a recessive bit and receivers that retrieve the message correctly overwrite this field with a dominant bit.

The end of frame field (EOF) denotes that the frame terminated. Finally, the intermission space represents the minimum number of bit periods that need to elapse following the frame before another station is allowed to transmit a message.

2.3. Arbitration

CAN uses the carrier sense multiple access with collision detection (CSMA/CD) mechanism in order to arbitrate access to the bus. It uses a priority scheme based on numerical identifiers in order to resolve collisions between two nodes wishing to transmit at the same time.

When two or more nodes wish to transmit, they sense the bus and if there is no bus activity, they begin to transmit their message identifier. At the same time, they also monitor the bus levels. If one node transmits a recessive bit on the bus and the other transmits a dominant bit the resulting bus level is a dominant bit. Therefore, the node transmitting a recessive bit will see a dominant bit on the bus and stop transmitting any further information. Any node that has lost during the arbitration process then waits until the bus becomes free before trying to retransmit its message. This process is shown in Fig.3

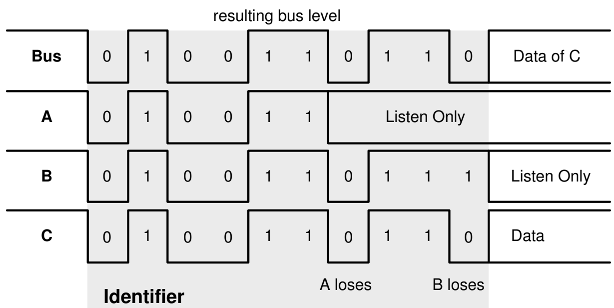

Fig. 3. Arbitration Mechanism in the CAN

2.4. The transmission time of the CAN

The data frame is the most useful type of all frames used in the CAN. The standard data frames contains from 44 to 108 bits. And, up to 23 stuff bits can be inserted in a standard data frame, depending on the data-stream coding. Furthermore, Interframe space following with the end of the data frame consists of 3 recessive bits. The overall maximum data frame length is between 47 bits to 134 bits for a standard frame. If the baud rate of the CAN bus is 1M bit/s, the transmission time(τ_m) of message N should be from 47 μs to 134 μs.

Because of the arbitration showing the Fig.3, nodes that lost the contention attempt to arbitrate with other nodes after the current transmission. This period of time is the arbitration time(τ_{ab}) of the message. And in order to put a message on the bus, a processor needs the preparation time (τ_p). Through these time of messages, the delay time can be looked for.

If a system makes up of m messages, the delay time(τ_n) of message N(priority n) is [6]:

$$\tau_n = (\tau_p + \tau_m) + \tau_m(n-1)(0 \leq n \leq m) \qquad (2)$$

3. An Inverted Cart Pendulum

An inverted cart pendulum was selected due to the basis of one's unstable nature. the Pendulum was suitable as a testbench for control experiments. If time delay in the CAN gives much effects to the pendulum, it will fall down. The inverted cart pendulum by the Quanser is used for experiments.

3.1. The model of an Inverted Cart Pendulum

This section describes the mathematical model for an inverted cart pendulum that was used when deriving the control laws.

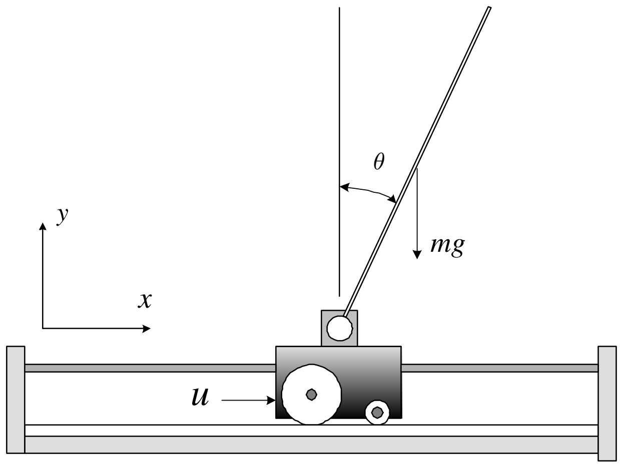

Fig. 4. Inverted Cart Pendulum Coordinate

The linearized equations describing the dynamics of the inverted cart pendulum[2] shown in Fig.4 are written next:

$$Ml\ddot{\theta} = (M+m)g\theta - u \tag{3}$$

$$M\ddot{x} = u - mg\theta \tag{4}$$

For implementation, the motor dynamics should be included into these pendulum dynamic equations. The dynamic equations including the motor dynamics are the following:

$$Ml\ddot{\theta} = (M+m)g\theta + b\dot{x} - aV_m \tag{5}$$

$$M\ddot{x} = -mg\theta - b\dot{x} + aV_m \tag{6}$$

where

$$a = K_iK_g/R_mr,\ b = K_iK_{emf}K_g/R_mr^2$$

From the obtained equations, the form of state-space equation can be represented in:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (7) \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \quad (8) \end{aligned}$$

where

$$\mathbf{x} = \begin{bmatrix} \theta & \dot{\theta} & x & \dot{x} \end{bmatrix}^T, u = V_m$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ (M+m)g/Ml & 0 & 0 & K_iK_{emf}K_g/R_mr^2Ml \\ 0 & 0 & 0 & 1 \\ -mg/M & 0 & 0 & -K_iK_{emf}K_g/R_mr^2M \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & -K_iK_g/R_mrMl & 0 & K_iK_g/R_mrM \end{bmatrix}^T$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

x	:	`the cart position`
θ	:	`the pendulum angle`
M	:	`the cart mass`
m	:	`the pendulum mass`
l	:	`the pendulum length`
g	:	`the acceleration of gravity`
u	:	`the input`
V_m	:	`the voltage applied to the motor`
K_i	:	`the motor torque constant`
K_g	:	`the motor gear ratio`
R_m	:	`the motor resistance`
K_{emf}	:	`the motor back-emf constant`
r	:	`the radius of the motor pinion gear`

Fig. 5. Definitions of Model of the Pendulum.

3.2. Stabilization of the pendulum

The objective in this work is to stabilize the pendulum in an upright position. The control law used to stabilize the pendulum is

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \tag{9}$$

The matrix **K** of the optimal control vector can be calculated in many ways. Here LQR method is used. It allows two design parameters, a Q matrix and R matrix. Q and R are penalty matrixes for the states x and the input u. Suitable Q and R matrixes were found to be:

$$\mathbf{Q} = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.01 \end{bmatrix} \tag{10}$$

From two matrixes, the matrix **K** of the optimal control vector can be founded to be:

$$\mathbf{K} = \begin{bmatrix} -107.200 & -30.513 & -6.324 & -19.407 \end{bmatrix} \tag{11}$$

4. The Process Exchanging Messages

Fig.6 shows the process that exchanges messages in the CAN. The system consists of three parts: the sensor part, the control part and the actuator part. For exchanging data, each parts need at least two mailboxes. Here mailbox0 and mailbox4 are used.

The sensor part sends its message including a message ID[0x5F] to the bus through Mailbox4. Then, the control

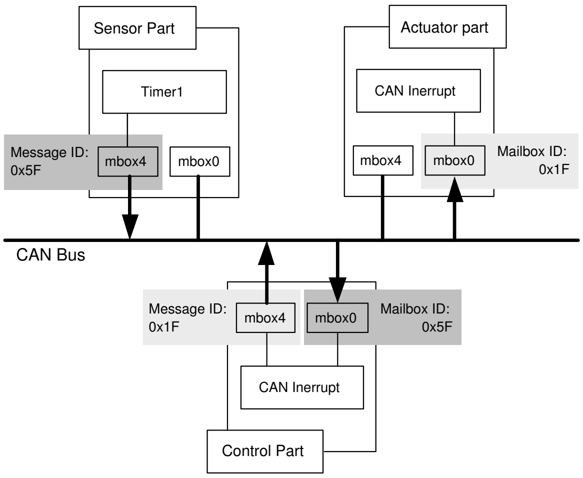

Fig. 6. The Process Exchanging Messages

part and the actuator part can receive the message with the message ID from the bus. It is the same as a mailbox ID[0x5F] of the control part. It means Mailbox0 of the control part can only receive the message from the sensor part. After finishing to process the message, new message for an actuator including a message ID[0x1F] is putted on the bus through Mailbox4 of the control part. Because the message ID and a mailbox ID[0x1F] of the actuator part have the same identifier, Mailbox0 of the actuator part can accept the message from the bus.

5. Hardware Design

Fig.7 shows the system that uses DSP controllers, TMS320LF2407, in order to consist of each parts; the sensor part, the control part and the actuator part.

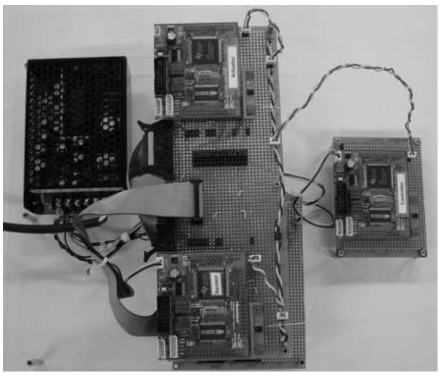

Fig. 7. Network System for an Inverted Cart Pendulum

The DSP controller has an integrated CAN controller inside. But, It needs a connection to the transceiver, SN65HVD230, to be attached to the CAN Bus. No counters like HCTL2020 are necessary for counting encoder pulses because the DSP controller itself provides the function counting pulses. And Digital-Analog Converter (DAC) should be used to drive a motor of the pendulum. MAX507 that has 12-bit parallel bus and bipolar output ($\pm 5V$) is used here.

6. Experiment

6.1. transmission time of the Network System

At the Section 2.4. the delay time was theoretically calculated through the equation (2). In this section, practical values were found by an experiment.

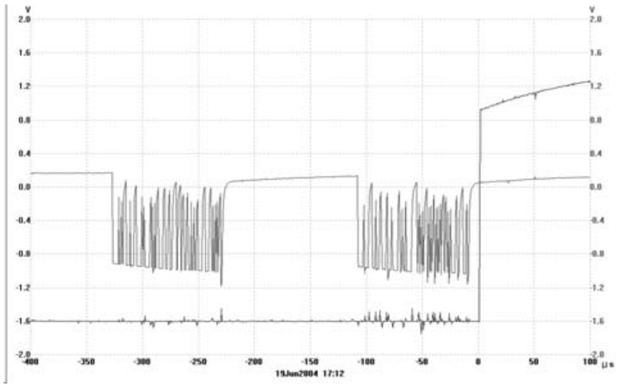

Fig. 8. Transmission Time (1M baud rate)

In Fig.8, we can find that the transmission time of first message is 106.3 μs(τ_{sc}) and second message is 107.9 μs(τ_{ca}). And each parts need time (τ_a, τ_c, τ_s) for computing. It shows Table 1.

Table 1. Computation time at each parts.

	Sensor(τ_s)	Control(τ_c)	Actuator(τ_a)
Time	83.47μs	112.7μs	50.21μs

The entire transition time that it takes to complete one control loop can be found out; It takes 460.97 μs.

If there are n messages having higher priority than a message in a part, the message in the part cannot be on the bus until the whole messages are transmitted. In Fig. 9, the second message has higher priority than the third. So, the transition time of a message from the control part to the actuator part is changed; It takes 213.6 μs. If the preparation time added in the transition time, it comes close to the result of the equation (2).

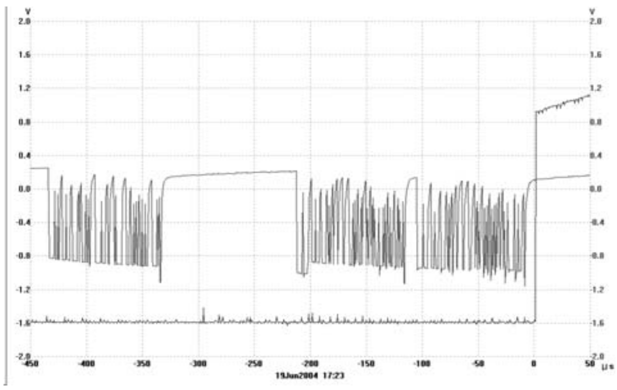

Fig. 9. Arbitration Time(1M baud rate)

6.2. Effects of time delay in the Inverted Cart Pendulum

This section presents the effects of time delay in the inverted cart pendulum. Even though the transmission time is existing between nodes in order to exchange messages, the transmission time confirmed from section 6.1. is too small to have an effect on the system. Fig.10 is shown the fact that the inverted cart pendulum can be controlled without a decline in the system performance.

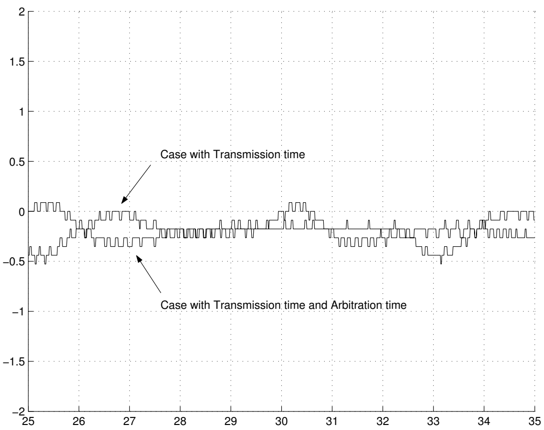

Fig. 10. Angle of the Pole of the Inverted Pendulum.

The network system used for controlling the inverted cart pendulum operates with a sampling time; 1 ms. All control gains and parameters were calculated being based on this sampling time. If the sampling time is changed, they also have to be modified.

Fig.11 presents the effects of time delay occurred in sampling time of the system. In this case, time delay is less than the sampling time. Therefore, all parts in the network system can work in the sampling time. It means that time delay in the sampling time doesn't have an effect on the system.

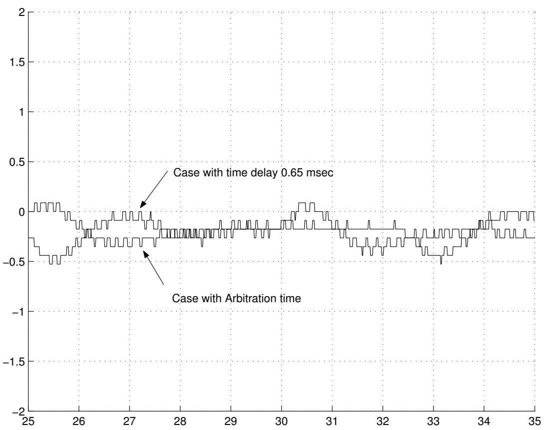

Fig. 11. The Effects of Time Delay Occurred in Sampling Time.

If 30 nodes are connected on the bus and they send messages at the same time, a message having the lowest priority has to wait about 3 ms according to the arbitration rule of the CAN. If it happens periodically, some parts in the network system cannot work in the sampling time. It makes the system unstable. It can be confirmed in Fig.12 and Fig.13.

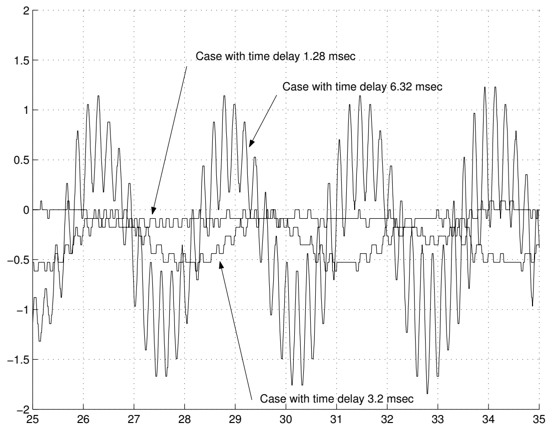

Fig. 12. The Effects of Time Delay between the sensor Part and the control Part.

Time delay(τ_{sc}) occurred between the sensor part and the control part has an effect on these two parts simultaneously. The control part cannot receive messages every sampling time, and the control part sends a message to the actuator part according to a system control frequency that is changed by time delay. If time delay is over 6.32 ms here, control of the system is impossible. This effect is shown in Fig.12.

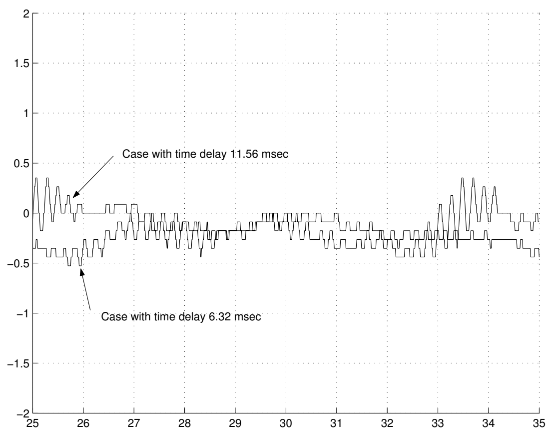

Fig. 13. The Effects of Time Delay between the control part and the actuator part.

Time delay(τ_{ca}) between the control part and the actuator part has an effect on just the actuator part. It means that the control part is normally working in the sampling time but the actuator part isn't. The actuator part cannot get some messages from the control part. Here even though time delay is over 11.56 ms, the system does not become unstable. Fig.13 shows this effect.

It can be find out that Time delay(τ_{sc}) between the sensor part and the control part makes the system more unstable than time delay(τ_{ca}) between the control part and the actuator part in the network system for controlling an inverted cart pendulum.

7. Conclusion

This paper shows that it was possible to make a network system including the CAN in a control loop in order to control an inverted cart pendulum. Through the network system, we found out that time delay occurring in the sampling time does not have an effect on the control frequency of all parts in the system, but time delay occurring over the sampling time changes the control frequency of some parts and makes the system unstable. Furthermore, time delay between two parts has an different effect on the network system, according as time delay occurs between the sensor part and the control part or between the control part and the actuator part. In future study, a algorithm will be developed to compensate these time delay.

References

[1] Wolfhard Lawrenz, *CAN System Engineering From Theory to Practical Application*, Springer, 1997.

[2] Katsuhiko Ogata, *Modern Control Engineering*, Prentice Hall, 1997.

[3] Lamia Chaari, Nouri Masmoudi, Lotfi Kamoun, "Electronic Control in Electric Vehicle Based on CAN Network," *IEEE SMC*, 2002.

[4] M.Farsi, K.Ratcliff and Manuel Barbosa, "An overview of Controller Area Network," *Computing & Control Engineering Journal*, pp. 113-120, June 1999.

[5] Andreas Hörjel, "Bluetooth in Control,", January 2001.

[6] NI Xiaodong, Zhang Yanjun, "Determining Message Delivery Delay of Controller Area Network," *IEEE*, pp. 767-771, 2002.

[7] Luís Miguel Pinho, Francisco Vasques, "Timing Analysis of Reliable Real-Time Communication in CAN Networks," *IEEE*, pp. 103-112, 2001.

[8] Robert Bosch, "CAN Specification 2.0," Part A and B, September 1999.