15th Triennial World Congress, Barcelona, Spain

TRAFFIC ANALYSIS OF A NETWORK-BASED CONTROL SYSTEM USING THE CAN

Duck-Jin Pae*, Hong-Ryul Kim*, Dae-Won Kim*, Hong-Seok Kim and Ho Gil Lee****

Dept. of Information Control Engineering, MyongJi University, KITECH***
{pdj666, museros, dwkim}@mju.ac.kr, hskim@cosmos.kitech.re.kr, leehg@kitech.re.kr

Abstract: This paper deals with an architecture of network-based control system using the CAN(Controller Area Network) protocol and its traffic analysis. It is difficult to determine an optimal network-based control architecture for a specific AGV(Autonomous Guided Vehicle) system with a manipulator arm. The fixed number of periodic messages to be occurred is pre-defined in the system. To determine whether the proposed system architecture is effective or not, we perform traffic analysis for the real-time communication of all messages. Through simulations, the range of transmission speed is found satisfying the required conditions and the permissible number of additional sensors is investigated for improving the system performance, when the sampling periods of analog sensors are determined under fixed condition that the transmission speed is over 500Kbps. *Copyright © 2002 IFAC*

Keywords: real-time, CAN(Controller Area Network), network-based control system, time delay, end-to-end communication.

1. INTRODUCTION

To overcome limitations and weakness of centralized control system, serial communication systems known as fieldbus systems have been developed. The limitations of fieldbus lie mainly in transmission expansion, a limited variety of topologies and transmission media. These limitations can be overcome by the network-based control system that distributed real-time control is possible (Schickhuber and McCarty, 1997).

Recently, in many systems of various application areas, such as airplanes, automobiles, building automations, and industrial automation systems, the network-based control system using fieldbus has been introduced. The network-based control system is usually composed of controllers, sensors, and actuators. The network-based control system can execute efficiently mutual functions between network components, such as multiple real-time controls and the exchange of information. Also, sensor signals and control signals generated by the network components are required to be transmitted in real-time to the corresponding network nodes (Halevi and Ray, 1988; Kwon and Kim, 1988). Hence, to handle efficiently and openly the data generated in the system, the system should be designed as an architecture including independent control units. This paper proposes an architecture of network-based control system applicable to an AGV(Autonomous Guided Vehicle) with a manipulator arm using the CAN(Controller Area Network) protocol, that is most commonly used in sensor level and distributed real-time control as a sort of fieldbus (Kim and Yoon, 1999). To analyze characteristics of the proposed architecture, modeling of latencies on communication using the CAN protocol and simulation are needed.

In the previous work, Tindell (1991, 1994a) defined four types of delays for latencies that can be generated on end-to-end communications. However, two types of delays such as media access delay and delivery delay, were modeled and analyzed when the token protocol and the priority bus were adopted.
Tindell (1994) studied models of two types of delays, transmission delay and queueing delay, and obtained the worst-case response time for the CAN protocol. and, this analysis was applied to an SAE benchmark. In this study, however, not all kinds of delays were considered.

In this paper, the mathematical models of delays for latencies on the CAN communication are defined in detail as four types. The worst-case response time is also calculated by them. Message from the nodes are scheduled to control efficiently the system including sensors and actuators, and analyzed to guarantee the real-time schedulability. As a result, a new architecture of the CAN-based control system is proposed and the performance of proposed system is analyzed through simulation according to the variation of transmission speed. And then, in order to improve the system performance, the permissible number of additional sensors is investigated when the sampling period of sensor is assumed.

The rest of the paper is organized as follows. In section 2, we briefly discuss the CAN protocol. In section, the mathematical models for message latencies on end-to-end communications are defined. In section, the worst case response times are calculated using the defined mathematical model and the analysis is described through simulation. In section, concluding remarks and future issues are mentioned finally.

2. OVERVIEW OF THE CAN PROTOCOL

The CAN was originally developed in the 1980s for the interconnection of control components in automotive vehicles. The CAN enables a huge reduction in wiring complexity and, additionally, makes it possible to interconnect several devices using a single pair of wires, allowing data exchange between them at the same time.

It was not long before this idea migrated from vehicles into such diverse areas as agricultural machinery, medical instrumentation, elevator controls, fairground rides, public transportation systems and industrial automation control components. It is because of its widespread use that the CAN semiconductors are inexpensive. Furthermore, since a large number of semiconductors produce CAN devices, the CAN technology is guaranteed well into the future.

The basic features of CAN are as follows:

- *High-speed serial interface*
- *Low-cost physical medium*
- *Short data lengths(max. 8bytes)*
- *Fast reaction times*
- *Multi-master and peer-to-peer communication*
- *Error detection and correction*

The CAN employs CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) 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. The identifier servers two purposes, filtering a message upon reception and assigning a priority to the message. The overheads of a CAN frame amount to a total of 47 bits(including 11 bits for the identifier field, 4 bits for a message length field, 16 bits for a CRC field, 7 bits for the intermission between frames). Some of these fields are 'bit stuffed': when five consecutive bits of the same polarity are sent, the controller inserts an extra 'stuff bit' of opposite polarity into the stream(this bit stuffing is used as part of the error signalling mechanism). Out of the 47 overhead bits, 34 bits are subject to bit-stuffing. The data field in a message(between 0 and 8 bytes) is also bit-stuffed. The size of the smallest CAN message is 47 bits, and the size of the largest one is 130 bits. Moreover, the CAN has a number of other features such as the error recovery protocol and the RTR(Remote Transmission Request) messages (Tindell, 1994b).

3. MATHEMATICAL MODELS OF THE TIME-DELAYS ON END-TO-END COMMUNICATION

The end-to-end communications delay can be classified into four major components: the generation delay, the queueing delay, the transmission delay, and the delivery delay (Tindell, 1994a). In this paper, we synthetically deal with four kinds of communication delays. In particular, we make in details the model of the generation delay that was not considered by previous works. Fig. 1 illustrates the time-delays on end-to-end communications.

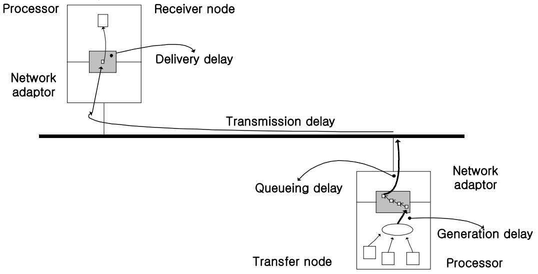

Fig. 1. CAN communication model

In this section, the delays on communication are calculated by using the mathematical models and the periodic messages generated from each node are scheduled by applying the DMS(Deadline Monotonic Scheduling) algorithm in which the deadlines of messages are less than or equal to their periods (Audesly, 1990). Deadline-monotonic priority ordering is similar in concept to rate-monotonic priority ordering (John and Lui, 1989). Priorities assigned to processes are inversely proportional to the length of the deadline. Thus, the process with the shortest deadline is assigned the highest priority and the longest deadline is assigned the lowest priority. This priority ordering defaults to a rate-monotonic ordering when the deadlines of messages are equal to their periods.

3.1. Basic Concept of Mathematical Modeling

The modelling of the worst-case response time of a task in the network-based control system is extended from the problem of computing the worst-case response time for a task *i*[10], in the local operating systems.

The worst-case time is represented by Eq.(1).

$$r_i = C_i + B_i + I_i \qquad (1)$$

Where r_i is the worst-case response time for a task *i* measured from the time the task is released, C_i is the worst-case computation time required by task *i* on each release, B_i is the blocking time from the lower priority tasks and I_i is the worst-case interference a task *i* can experience.
The total interference Ii is given by:

$$I_i = \sum_{\forall j \in hp(i)} \left\lceil \frac{r_i}{T_j} \right\rceil C_j \qquad (2)$$

Where *hp(i)* is the set of tasks with higher priorities and T_i is the period of task *j*. When equation(1) and (2) are combined, the unknown term r_i appears on both the left and right hand sides of the equation:

$$r_i = C_i + B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{r_i}{T_j} \right\rceil C_j \qquad (3)$$

From this Eq.(3), the mathematical models of each time-delay on end-to-end communications are derived.

3.2. Generation Delay

The generation delay is the time taken to process the task generated from the application of a node and to queue the message (Tindell, 1994a). Thus, the generation delay can be composed of the computation time and the interference time from the concept of Eq.(1). It is represented by Eq.(4).

$$r_i = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{r_i}{T_j} \right\rceil C_j \qquad (4)$$

Where r_i denotes the generation delay of the worst-case task *i*, T_j denotes the period of task *j*, and *hp(i)* represents a set of tasks with higher priority than task *i*. C_j and C_i represent the execution time of task *i*, *j* in the worst-case, respectively (Tindell, 1994c).

The priority of the tasks generated from each node is determined by using the DMS algorithm. The worst-case delay time can be calculated as Eq.(4) since each process includes the interference time by its upper process with higher priority, as the second term of Eq.(4).

3.3. Queueing Delay

The queueing delay is the time the message spends waiting to be removed from the queue by the communications device until the message occupies the network medium(Tindell, 1994; 1994a; 1994b). This time includes the time blocked from lower priority tasks and the time interfered by the execution of tasks with higher priority. The blocking time and the interference time are derived from of Eq.(1). The queueing delay is represented by Eq.(5).

$$t_m = B + \sum_{\forall j \in hp(m)} \left\lceil \frac{t_m + J_j + \tau_{bit}}{T_j} \right\rceil C_j \qquad (5)$$

The second term of Eq.(5) represents the delay of queueing the message to the queue.

$$B = \max_{\forall K \in lp(m)} (C_k) \qquad (6)$$

3.4. Transmission Delay

The transmission delay is the time taken for the message to be sent once it has been removed from the queue(Tindell, 1994; 1994b) and is represented by Eq.(7).

$$C_m = \left(\left\lfloor \frac{34 + 8s_m}{5} \right\rfloor + 47 + 8s_m \right) \tau_{bit} + \rho \qquad (7)$$

Where C_m denotes the transmission delay of the message to be transferred physically over the network bus, s_m is the size of a message, and ρ is the delay considering an electrical property of physical transmission medium, and is represented by an integer according to the property of a medium. The first term of Eq.(7) includes the overheads created from a CAN frame. When five consecutive bits of the same polarity are sent, the controller inserts extra 'stuff bit' of opposite polarity into the stream to check the error of the message. The total overheads are 47. Out of the 47 overhead bits, 34 are subject to bit-stuffing(Tindell, 1994b).

3.5. Delivery Delay

By expanding the definition of our previous work(Kim J.K. and Kim D.W., 1998), the delivery delay is defined as the amount of time taken to process the incoming data and deliver it to destination tasks(Tindell, 1991; 1994a) and is represented by the following Eq.(8).

$$D_H = C_H + I_H \qquad (8)$$

If the message is arrived at a node, it is assumed the processor handles a task by means of an interrupt. The first term of Eq.(8) is the time needed by ISR(Interrupt Service Routine). The second term represents the delay to handle the process generated on a node and is represented by Eq.(9).

$$I_H = \sum_{\forall j \in hp(H)} \left\lceil \frac{D_H + B_j}{T_j} \right\rceil C_j + \sum_{\forall m \in im(p)} P_m C_I \qquad (9)$$

Where B_j is the maximum time which the execution is arbitrarily delayed by the process j, P_m denotes the number of the packet, and C_I denotes the execution time of the ISR. The first term represents the time delayed by the processes with higher priority, the second term represents the execution time of ISR when a message is arrived at the message queue. Considering the above defined Eq.(4), (5), and (9), the identical terms appear on both the left and the right hand sides of the equations. To calculate this effectively, the initial value is set to 0. The calculation for all delays is solved in terms of the iterative technique(Audesly, 1994).

4. SIMULATIONS AND ANALYSIS RESULTS

The target system is assumed to be an AGV with a manipulator arm and the objectives of control is the position control of AGV and the motion control of manipulator arm. Throughout the section, the mathematical models for the time-delays of messages are defined.

This section applies it to an initial architecture of CAN-based control system and a new architecture of CAN-based control system proposed in this paper and deals with the guarantee of the real-time communications for the messages generated in the system by calculating the worst-case response time. The worst-case response time is calculated by the iterative execution technique with the mathematical models defined in the previous section under the following assumption conditions.

Assumption conditions:
- ·No queueing jitter
- ·No delay considering an electrical property of physical transmission medium

Through simulation, the range of transmission speed is found satisfying the required conditions and the permissible number of additional sensors for improving the system performance is investigated when the sampling periods of the sensors are assumed.

4.1. The CAN-based Control System Considering the Workload of a Node

This system has a gyroscope and an accelerometer for the position control of AGV, sonar sensors for the avoidance of obstacle, and a vision sensor for image data. Its architecture is shown in Fig. 2.

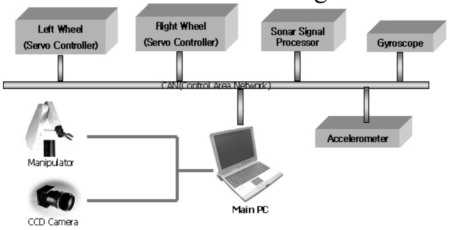

Fig. 2. The initial architecture of CAN-based control system for an AGV with a manipulator arm

Table 1 The definition of messages on the initial CAN-based control system

message	message size (byte)	period (μs)	deadline (μs)
1	8	5000	4900
2	8	5000	4950
3	8	5000	4999
4	8	10000	9990
5	2	2000	1990
6	6	2000	1999

This system architecture can minimize the calculation load as the main controller processing directly lots of data from the manipulator and the vision sensor. Assumed experimental conditions are represented in the following Table 1, however condition of aperiodic messages is not considered in this paper. From the analysis result of system performance according to variation of the transmission speed in the network system, we can confirm that the real-time scheduling is possible in case of over 250Kbps. It is shown in the following Fig. 3.

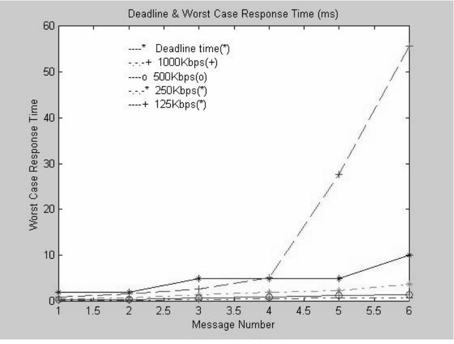

Fig. 3. The performance analysis of the initial CAN-based control system for an AGV with a manipulator arm

4.2. The Proposed CAN-based Control System

Fig. 4 represents an architecture of CAN-based control system applicable to a practical target system. The manipulator arm has 7 axes and its hand has 3 axes. To transfer the incoming data from the encoder of each axis via feedback loop, five CAN nodes are required if one CAN node is assumed to manage 2 axes servoing, and two CAN nodes are assigned to the right wheel and the left wheel. And, three CAN nodes are assigned to the unit of sensors, such as the unit of vision sensor, the unit of gyroscope and accelerometer sensors, and the unit of sonar sensors.

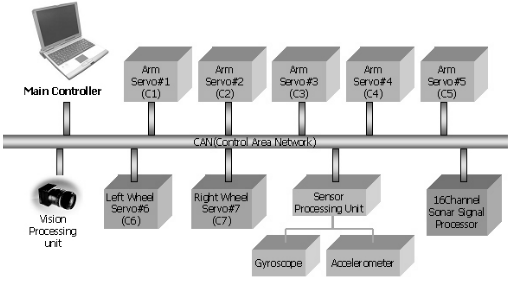

Fig. 4. The architecture of proposed CAN-based control system for an AGV with a manipulator arm

Table 2 The definition of messages on the proposed CAN-based control system

message	message size(byte)	period (μs)	deadline (μs)	source	destination
1	8	5000	4990	MC	C1
2	8	5000	4980	MC	C2
3	8	5000	4950	MC	C3
4	8	5000	4999	MC	C4
5	8	5000	4800	MC	C5
6	8	5000	4900	MC	C6
7	8	5000	4700	MC	C7
8	1	5000	4650	MC	Cn
9	8	10000	9900	S/S	MC
10	8	10000	9800	S/S	MC
11	8	10000	9999	S/S	MC
12	8	10000	9990	S/S	MC
13	8	2000	1990	G.A/S	MC
14	8	5000	9998	V/S	MC

These CAN nodes are designed with embedded processors. Thus, the number of CAN nodes in this system are totally 11 including the main controller and the number of generated messages are assumed to be 14. However, aperiodic messages generated on network are not considered in this paper.

The definition of messages generated from the proposed CAN-based control system are as the following Table 2. In Table 2, MC represents the main controller, Cn represents CAN nodes for servo controls, and subscript n denotes the number of total nodes. S/S represents the unit of processing sonar sensors, G.A/S does the unit of processing gyroscope and accelerometer sensors, and V/S does the unit of processing vision sensor.

The sampling periods for controlling each axis of the manipulator arm and for actuating the left and the right wheel are assumed to be 5ms, the sampling periods of the vision sensor and the sonar sensors are assumed to be 10ms, and the sampling period of gyroscopes and accelerometers is assumed to be 2ms. The range of transmission rate, in case of the CAN protocol, is from 125Kbps up to 1Mbps.

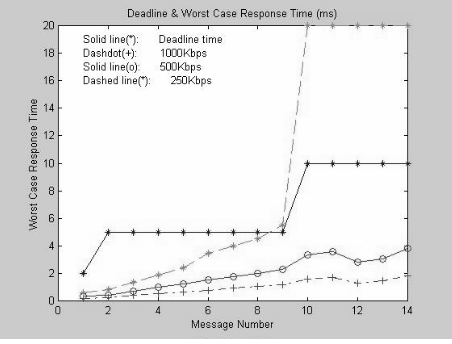

Fig. 5. The performance analysis of the proposed CAN-based control system

Fig. 5 shows the result of simulation for guaranteeing the real-time scheduling of messages according to the variation of transmission speed. In Fig. 5, the deadline of each message is represented by solid line(*). Thus, the real-time schedulability is guaranteed only when the transmission speed is over 500Kbps.

Moreover, we assume additional sensor nodes are needed in the proposed CAN-based control system. In this case, we may investigate how many additional nodes are permissible under the condition of satisfying the sampling period of messages generated from nodes. When the transmission speed is over 500Kbps, the following Fig. 6 and Fig. 7 show the results of simulation.

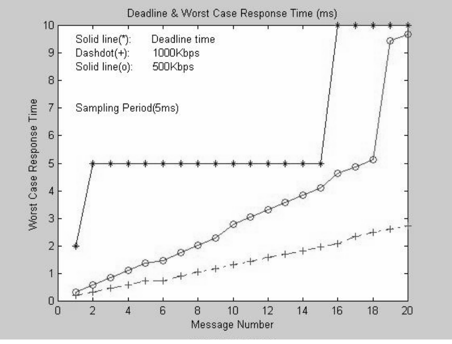

Fig. 6. The number of additional sensors when the sampling period is 5ms

In this simulation, we investigated the permissible number of additional sensor nodes satisfying the deadline condition when the size of message was assumed to be 8 bytes and the sampling period of sensors was assumed to be 5ms and 10ms, respectively. In case of 5ms, the maximum number of additional nodes was 6. In case of 10ms, the maximum number of them was 13.

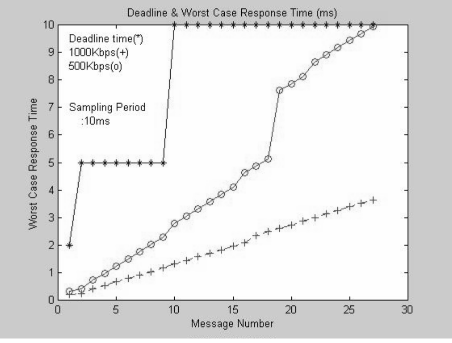

Fig. 7. The number of additional sensors when the sampling period is 10ms

5. CONCLUSIONS

The network-based control system has advantages of flexibility for expansion and inter-related functional performance between nodes compared with the centralized control system.

In this paper, the mathematical models of delays on the CAN communication were defined in details as four types, such as the generation delay, the queueing delay, the transmission delay, and the delivery delay. In particular, we derived the mathematical models of the time-delays from calculating the worst-case response time in the real-time computing system and made the model of the generation delay in details. The architecture of the CAN-based control system for the AGV with a manipulator arm was introduced and the worst-case response time was calculated by the defined mathematical models. From the result, we could confirm that the real-time scheduling was possible in case of over 250Kbps under given assumptions.

New architecture of the CAN-based control system was also proposed. To find the constraint conditions of proposed architecture, traffic analysis was performed. Through simulation, we found the range of transmission speed satisfying the required conditions. Also, we could determine the permissible number of additional sensors for expansion of system functionality under given assumptions

However, in this paper, we could not consider a control loop via network for controlling the AGV with a manipulator arm and did not deal with scheduling algorithm for effective processing of messages.

In the near future, hence, the research of control loop for an efficient control and the research of an architecture in case of considering the control loop will be performed. And, an approach to scheduling algorithm will be also made.

REFERENCES

Audesly, N.C., (1990). Deadline Monotonic Scheduling. YCS146, Dept. Computer Science, University of York. [online] available at http://cs.york.ac.uk/rts/.

Audesly, N.C., (1994). Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling. SEJ, Dept. Computer Science, University of York. [online] available at http://cs.york.ac.uk/rts/.

Halevi, Y., A. Ray, (1988). Integrated Communication and Control Systems. *Dynamic Systems, Measurement, and Control*, **Vol. 110**, pp. 367-381.

Hong, S.H., (1997). Management and Control of Fieldbus Network Traffic by Bandwidth Allocation Scheme. *Institute of Control Automation System engineering,* **Vol. 3**, pp. 77-88.

John Lehoczky, Lui Sha (1989). The Rate Monotonic Scheduling Algorithm: Exact Characterization And Average Case Behavior. *IEEE*, Dept. of Computer Science.

Kim, G.W., J.W. Yoon, (1999). Evaluation of Real-time Characteristics of CAN for Real-Time Distributed Control Systems. *Proceedings of the 14th KACC*, pp. 235-239.

Kim, J.K., D.W. Kim, (1998). An Approach to Implementation of an Assembly Cell and its Simulation Using the CAN(Controller Area Network). *Proceedings of the 13th KACC*, pp.1480-1483.

Kwon, W.H., Y.S. Kim, (1998). Network Protocols in Distributed Real-Time System. *ICASE special report*, pp.27-34.

Tindell, K.W., (1994a). Analysis of hard Real-Time Communications. YCS-222, [online] available at http://cs.york.ac.uk/rts/.

Tindell, K.W., (1994b). Guaranteed Message Latencies for Distributed Safety-Critical Hard Real-Time Control Networks. RTRG, Dept. Computer Science, University of York. [online] available at http://cs.york.ac.uk/rts/.

Tindell, K.W., (1994c). An Extensible Approach for Analyzing Fixed Priority Hard Real-Time Tasks. RTRG, Dept. Computer Science, University of York. [online] available at http://cs.york.ac.uk/rts/.

Tindell, K.W., A. Burns and A.J. Wellings, (1991). Guaranteeing Hard Real Time End-to-End Communications Deadline. RTRG 91102, Dept. Computer Science, University of York. [online] available at http://cs.york.ac.uk/rts/.

Tindell, K.W., A. Burns, (1994). Guaranteeing Message Latencies on CAN. *Proc. 1st International CAN Conference*.

Schickhuber, G., O. McCarthy (1997). Distributed Fieldbus and Control Network Systems. *COMPUTING and CONTROL ENGINEERING JOURNAL*.