Time Delay Analysis in Networked Control System Based Controller Area Network

Amira Sarayati Ahmad Dahalan[1], Abdul Rashid Husain[2], Mohd Badril Nor Shah[3], Muhammad Iqbal Zakaria[4]
[1,2.3,4]Faculty of Electrical Engineering, University Teknology Malaysia
amira_sarayati@yahoo.com, rashid@fke.utm.my, bad_z81@yahoo.com, iqbal_z04@yahoo.com

***Abstract*—The uses of data network in a control loop have gained increasing attentions in recent years due to cost effective and flexible applications. But, still the major challenges in this networked control system (NCS) are the delay of the data transmission throughout the communication network. Therefore, this paper presents an analysis of the time delay in NCS and also the development of the NCS platform based on well known communication networks, which is, Controller Area Network (CAN).**

***Keywords*—networked control system, time delay, controller area network**

I. INTRODUCTION

Network Control System (NCS) is a distributed feedback control system, which is composed by many nodes connected trough real time network. All definitions found in literature for an NCS have one key feature in common. This defining feature is that information is exchanged among control system components which are sensor, controller, and actuator in the form of information packages through a network.. This new approach differs to the traditional systems since the controller and the plant are physically separated and connected through an industrial network. NCS eliminates unnecessary wiring, thus reducing the complexity and the overall cost in designing and implementing the control systems.

Despites the advantage of the NCS there are challenges related to the development of NCS with inclusion of the communication network in the close loop control system. The time taken for the measurements by the system and to transmit a control signal to the actuator and other components in NCS depends upon the characteristics of the communication network.

However, the introduction of communication network in industrial systems can improve the efficiency of the control system by reducing time and costs of installation and maintenance. The well known communication network is Controller Area Network (CAN) which defines a medium access control protocol developed in the 1980s which is earlier aiming to design a bus system for the automotive applications. Among all the existing communication networks, CAN provide an advanced built-in feature, which makes it suitable for complex real-time applications. Some of these features are priority-based and high transmission speed. Because of this advantage and the multi-master work mode, CAN has been implemented widely in Network Control System application [1, 2].

One of the important issues arises in NCS development is the delay of data transmission between the components throughout communication network. The NCS constructed without considering this delay have a low performance in real time and control aspects. Therefore, the network configuration parameters such as the scheduling messages need to be considered since it is also influenced the NCS performance [3].

II. DELAY IN NETWORKED CONTROL SYSTEMS

The NCS operates over a network, therefore the data transfers between the controller and the remote system will induce network delays in addition to the controller processing delay. In other words, the delay appears when exchange data activity between sensors, actuators, and controllers through the networks. This dynamic activity will affect the performance of control system designed without considering it and even stability of the system. Besides, the signal processing and computational delays that depending on the scheduling protocol should be taken into account caused they also contribute to the total delay.

Furthermore, if all the components of the NCS are time-driven, there is an additional synchronization delay, because the components have to wait until the next sample instant until they can act. One of example in [4] the controller might receive a measurement from the plant $0.1h$ after the measurement is made, but it would have to wait until the next sample instant until the control signal would be transmitted to the actuator, which again would wait until the next sample instant before actuation of the control command.

The direction of the data transfer as the sensor to controller delay $\tau^{sc}$and the controller to actuator delay τ^{ca} can be seen in Figure 3. The delays are computed as

$$\tau^{sc} = t^{cs} - t^{se} \quad \text{(a)}$$

$$\tau^{ca} = t^{rs} - t^{ce} \quad \text{(b)}$$

where is t^{se} the time instant that the remote system encapsulates the measurement to a frame or a packet to be sent, t^{cs} is the time instant that the controller starts processing the measurement in the delivered frame or packet, t^{ce} is the time instant that the main controller encapsulates the control signal to a packet to be sent, and t^{rs} is the time instant that the remote system starts processing the control signal. In fact, both network delays can be longer or shorter than the sampling time T [5].

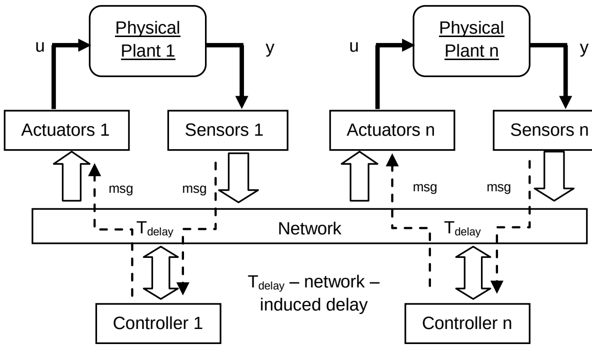


Figure 1. Structure of NCS with delays

The controller processing delay τ^c and both network delays can be lumped together as the control delay τ for ease of analysis. This approach has been used in some networked control methodologies. Although the controller processing delay τ^c always exists, this delay is usually small compared to the network delays, and could be neglected. In addition, the sampling periods of the main controller and of the remote system may be different in some cases.

The delays τ^{sc} and τ^{cc} are compose of at least the following parts [6].The waiting delay τ^w is the delay where the main controller or the remote system has to wait for queuing and network availability before actually sending a frame or a packet out. The frame time delay τ^w is the delay during the moment that the source is placing a frame or a packet on the network. While the propagation delays τ^w are the delay for a frame or a packet traveling through a physical media. The propagation delay depends on the speed of signal transmission and the distance between the source and destination.

Medium access control (MAC) protocol of the control network, delay in network can be constant, time varying, or even random. MAC protocols generally fall into two categories which are random access and scheduling [7]. Carrier sense multiple access (CSMA) protocols always applied in random access networks. Figure 2 illustrates various possible situations of network.

As an example, Figure 2 shows two nodes continually transmitting messages with respect to a fixed time line. A node on a CSMA network monitors the network before each transmission. For Case 1, the nodes transmission begins immediately when the network is idle; otherwise it waits until the network to be free. Collision occurs when two or more nodes try to transmit simultaneously. The way to resolve the collision is using a protocol dependent. For Case 2, it uses CSMA with a bitwise arbitration (CSMA/BA) protocol. It is known that messages are prioritized, when a collision occurred the message with the highest priority is transmitted and transmission of the lower priority message is terminated and will be retried when the network is idle. While for Case 3, CSMA with collision detection (CSMA/CD) protocol is employed. When there is collision, all of the affected nodes will back off and wait a random time, usually decided by the binary exponential back off algorithm [8] and retransmit. Packets on the networks are affected by random delays, and the worst-case transmission time of packets is unbounded. Therefore, CSMA/CD networks are generally considered nondeterministic. However, if network messages are prioritized likes Case 2, the higher priority messages have a better chance of timely transmission.

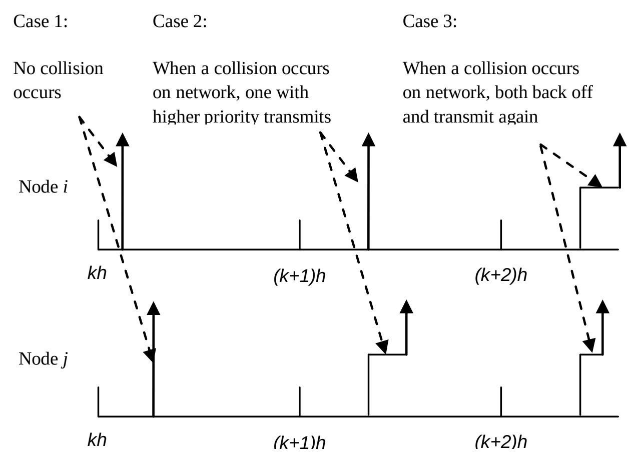


Figure 2. Timing diagram for two nodes on random access network

III. CONTROLLER AREA NETWORK

The communication network is the backbone of the NCS, therefore, the best network used in this work is the Controller Area Network (CAN) which is a multi-master serial bus that broadcast messages to all nodes in the network system. CAN was defined in International Standardization Organization (ISO) as a serial communication bus to replace the complex wiring harness with a two wire bus.The CAN system offers a transmission speed of up to 1 Mbit/s with error detection method for effective transmission [9].

A. Communication mechanism

CAN uses carrier sense multiple access protocol with collision detection (CSMA/CD) and arbitration on message priority as its communication protocol [9-11]. This communication protocol allows every node in CAN to monitor the bus network in advance before attempting to transmit a message. When no activity occurs in the network, each node has the same opportunity to transmit a message. Each type of message has a unique message identifier which serves as the name of the message as well as the priority code of the message. Lower numbered identifiers correspond to higher priority messages.

The message identifier is the first component of a message sent by a node to the bus. When a node wants to transmit a message it checks the bus to see if there is a message on the bus. If not, the node will begin its transmission immediately. If there is a message on the bus, then transmissions delayed until the current message is completed. Once the node finds the bus idle transmissions begin immediately. Transmitted messages are broadcasted to the bus and all nodes on the network are constantly listening to the bus. When a message appears on the bus, every node checks the message identifier. If this message identifier matches a receiving object header in the node, then the node will retrieve the message otherwise the message is ignored. When two or more nodes begin their transmission at the same time, a collision will occur. As two message identifiers are sent to the bus bit by bit, the dominant ‘0’ bit will overwrite the recessive ‘1’ bit, so the identifier with the smaller binary value will overwrite the identifier with larger value. While a node is transmitting, it listens to see if the bit on the bus is the same as it is sending. If what it hears is different from what it sent, a collision has occurred and it will stop its transmission immediately. The higher priority message will continue until it is completed sending package. After the higher priority message is completed, the lower priority message will try to transmit again. If there are other higher priority messages waiting to be transmitted, the lower priority message will lose arbitration again and again until all higher priority messages are done sending the package.

B. Scheduling of CAN messages

CAN bus use the data frame identifier to express the source of information and the priority. The frames are broadcasted on the bus and each station can decide if the message content is relevant by examining the message identifier of received frames. These identifiers have to be assigned statically during the design phase of the bus system to avoid ambiguity in the interpretation of the frame content. A global view of the complete system is needed in this process.

The data frame identifier can be presented both statically and dynamically. As stated in [12, 13] it can be divided into two. One is static priority scheduling algorithm which is fixed priority scheduling algorithm (FP), the rate monotonic scheduling algorithm (RMS) and the deadline monotonic scheduling (DMS). The other is dynamic-priority scheduling algorithm such as the earliest deadline first scheduling (EDF), the shortest idle time scheduling priority (LLF).

Rate monotonic scheduling (RMS) algorithm is a typical static scheduling algorithm, based on the different cycles of each task to allocate priority. The shorter the cycle, the higher priority task takes and also it is a fixed priority assignment algorithm. The task has been set priority before the allocation, so it will not change during the course of time.

EDF (Earliest Deadline First) is a type of dynamic scheduling algorithm is widely used in processors implementing the task. In order to achieve an effective scheduling, the priority is determined by the deadline of the task, so the tasks can be completed within the required deadline. The basic idea of the EDF scheduling algorithm is based on the absolute deadline to allocate priority dynamically. The higher priority is assigned with the shorter cut-off [14, 15 and 16].

IV. DEVELOPTMENT OF THE NCS PLATFORM BASED ON CAN BUS

Networked control system based CAN can be simulated using Matlab/Simulink with TrueTime toolbox [1]. TrueTime is good tools used for experimental platform for research on dynamical real-time controls system, by taking into account the effects of the execution of the control task and the data transmission on the controlled system dynamics. It provide computer and network block which execute user define thread, such as scheduling policies, controller task, network interface task, I/O tasks and message communication. TrueTime allow the researchers to study the compensation schemes that adjust the control algorithm based on measurement of actual timing variation [17].

In order to perform the analysis of CAN-based control system of the simulation platform is established by using three DC motor connected to CAN, controlled by one controller node with PD control algorithm as shown in Figure 3. The controller parameters are shown in Table 1. Tasks that performing data sampling from sensors are running periodically (clock driven), meanwhile tasks to calculate PD control signal and send it to particular node and also tasks to send control signal to actuators are event driven task which are trigged by interrupt signal. CAN data speed is set at 100kbps and all nodes are using Deadline Monotonic (DM) scheduling with assumption no clock drift phenomenon.

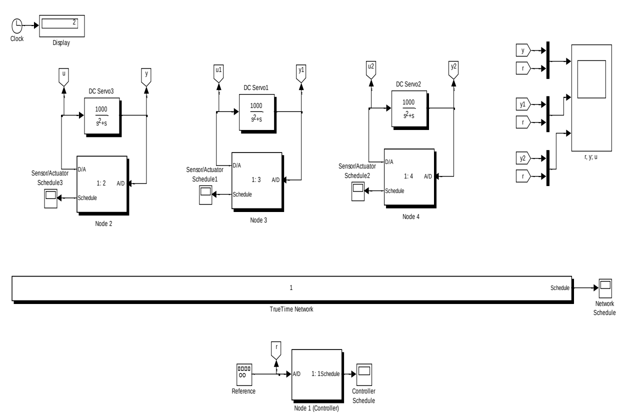


Figure 3. The test platform of NCS

TABLE 1

Parameter	Meaning	Values
K_P	Proportional value	1
K_D	Differential value	0.04

The PD parameters.

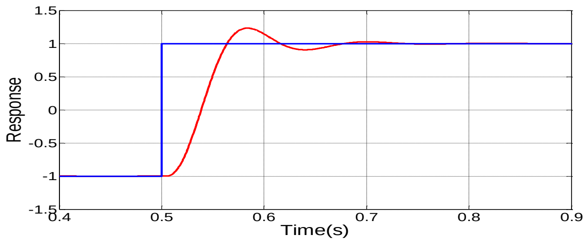
A. Simulation Result and Analysis

From this simulation works, the delays in the network can be observed by change the data speed or the baud rate of the CAN bus. Therefore, the result and analysis can be reviewed in 3 Cases. The graph Time versus Response in the figures below represents:

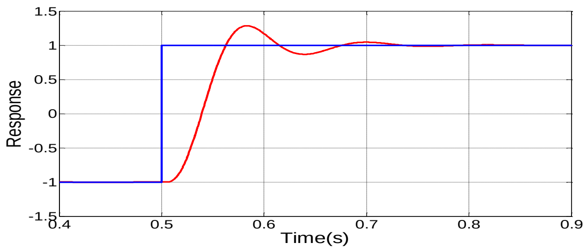
(a) DC motor 1
(b) DC motor 2
(c) DC motor 3

1) Case 1: The data speed set to 100kbps and using DM Scheduling.

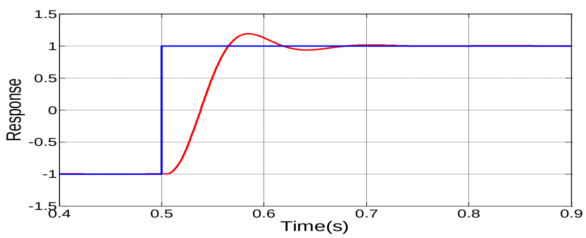
It is can be observed in Figure 4 that each DC motor produce slightly different result. DC motor 2 and DC motor 3 have higher overshoot and more oscillation from DC motor 1 due to longer delay from sensors to controller. Tables 2(a) show the sensors-controllers delays that are logged from simulation. The delays are found constant from the beginning until end of simulation.



(a)



(b)



(c)

Figure 4. Simulation result for *Case 1*

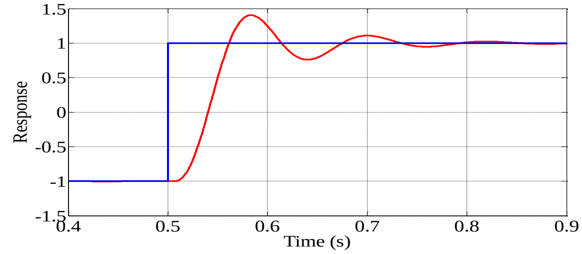
TABLE 2(A)

Node	Sensor-controllers delay (s)
DC motor 1	0.0008
DC motor 2	0.0016
DC motor 3	0.0024

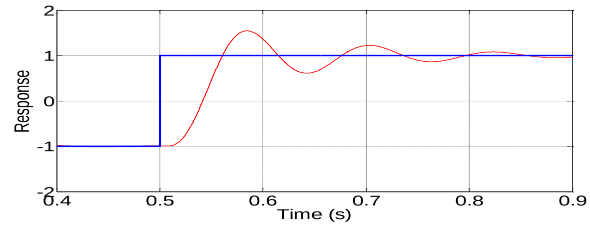
Sensor-controllers delay for corresponded node

2) Case 2: Data speed set to 50kbps and using DM scheduling

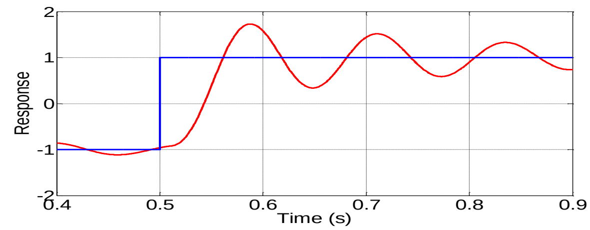
When CAN data speed is reduced to 50kbps, it can be observed the response of each DC motor become degrade due to longer sensor-controller delay as shown in Figure 5. Table 2(b) shows the sensors-controllers delays that are logged from this simulation.



(a)



(b)



(c)

Figure 5. Simulation result for *Case* 2

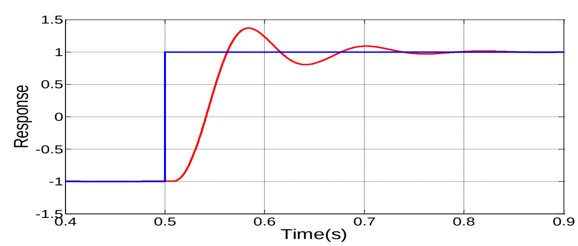
TABLE 2(B**)**

Node	Sensor-controllers delay (s)
DC motor 1	0.0016
DC motor 2	0.0032
DC motor 3	0.0048

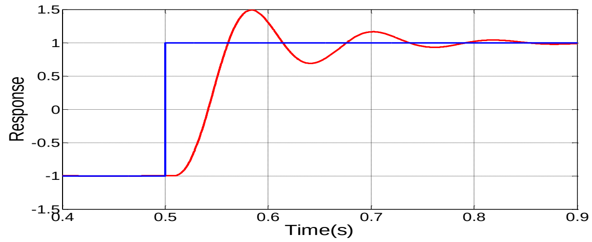
Sensor to controllers delay for *Case 2*

3) Case 3: Data speed set to 100kbps and interference.

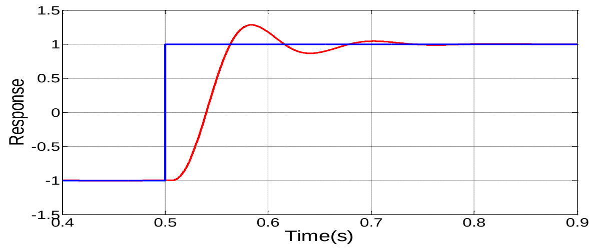
The system then is added an interference node which generate high priority task at random time with CAN speed transmission at 100kbps. This interference node represents the other nodes that execute high important task in the middle of system operation. It can be seen that the response of each DC motor experience higher overshoot and more oscillation as shown in Figure 6. Sensor-controller delays are found to be random.



(a)



(b)



(c)

Figure 6. Simulation result for *Case 3*

V. CONCLUSION

From simulation above, the sensor-controller delay has been observed. The result from *Case 1* with 100kbps data speed have a constant time delay rather than *Case 2* with 50kbps data speed resulting longer time delay. While for *Case 3*, with the data speed at 100kbps and interference node that generate high priority task, the time delay found to be random.

The simulation using other scheduling technique such as fixed priority (FP) and earliest deadline first (EDF) give same result as DM scheduling technique since there are only a few tasks to be executed inside each node.

It is concluded that delay in CAN network is influenced by CAN transmission speed and high priority random task. The delay also could become longer with the increasing number of nodes. The number of tasks inside nodes with different deadline should be increased to observe the scheduling technique effect to the delay. This research can be used as reference for further works that used a large number of tasks inside node to analysis the time delay.

REFERENCES

[1] Martion Andersson, Dan Henriksson, Anton Cervin. *TrueTime1.5—Reference Manual*, 2007.

[2] K. Tindell, A. Burns, A. J. Willings. *Analysis of Hard Real-time Communication.* Real-Time Systems, Vol.9, No.2, pp.147-171, 1995.

[3] Lian, F.L., Moyne, J.R. and Tilbury, D.M. *Network Design Consideration for Distributed Control Systems.* IEEE Transactions on Control Systems Technology, Vol. 10(2), pp. 297-307, 2002.

[4] Bollepally Raju. *Time Delay Compensation Schemes With Application To Networked Control System.* Master Thesis. National Institute Of Technology, Rourkela , 2007-2009.

[5] R. A Gupta, M. -Y Chow. *Networked Control System: Overview and Research Trends.* IEEE Transactions on Industrial Electronics, Vol 57, No 7, July 2010

[6] Lian, F.L., Moyne, J.R. and Tilbury, D.M. *Performance Evaluation of Control Network: Ethernet, Controlnet, and Device netsystems*. IEEE Control Systems Magazine, 21(1), 22-83, 2001.

[7] J.D. Spragins, J.L. Hammond, and K. Pawlikowski, *Telecommunications: Protocols and Designs*. Reading, MA: Addison-Wesley, 1991.

[8] A.S. Tanenbaum. *Computer Networks*, 3rd edition Upper. Englewood Cliffs, N.J.: Prentice-Hall, 1996.

[9] Pazul, K. *Controller Area Network (CAN) Basics*. Microchip Technology Inc, 1999.

[10] Chen, H.; Tian, J. *Research on the Controller Area Network*. International Conference on Networking and Digital Society, Vol. 2, pp.251 - 254, 2009.

[11] Steve Corrigan. *Introduction to the Controller Area Network (CAN)*. Texas Instrument, Application Report, 2008.

[12] GC. Walsh, H. YE, L G. Bushnell. Stability Analysis Of Networked Control Systems. *International Proceedings of the American Control Conf.* San Diego, California, pp.2876-2880, 1999.

[13] Y. Tipsuwan , Mo-Yuen Chow. *Control methodologies in networked control systems*. Control Engineering Practice, Vol.11, Issue 10, pp.1099-1111, 2003.

[14] H. Shokry, M. Shedeed, S. Hammad, M. Shalan, A. Wahdan. Hardware EDF Scheduler Implementation On Controller Area Network Controller. *Proceedings of 4th International Design and Test Workshop (IDT)*, pp.1-6, 15-17, 2009.

[15] P. Pedreiras, L. Almeida. *EDF Message Scheduling On Controller Area Network.* Computing & Control Engineering Journal, Vol.13, No.4, pp.163-170, 2002.

[16] S. Fuster, F. Rodriguez, A. Bonastre. Software-based EDF message scheduling on CAN network. *Proceedings of ICESS 2005-Second International Conference on Embedded Software and Systems*, pp.400-455, 2005

[17] Al-Hammouri, A., Branicky, M.S and Liberatore, V.*Co-simulation Tools for Networked Control Systems.* Springer-Verlag, Berlin, pp.16-29, 2008.