A Scheduling Method for Network-Based Control Systems

Hong Seong Park, Yong Ho Kim, Dong-Sung Kim, and Wook Hyun Kwon

***Abstract*—This paper presents a scheduling method for network-based control systems with three types of data (periodic data, sporadic data, and messages). As a basic parameter for the scheduling method for network-based control systems, a maximum allowable delay bound is used, which guarantees stability of network-based control systems and is derived from characteristics of the given plant using the presented theorems. The presented scheduling method for network-based control systems can adjust the sampling period as small as possible, allocate the bandwidth of the network for three types of data, and exchange the transmission orders of data for sensors and actuators. In addition, the presented scheduling method guarantees real-time transmission of sporadic and periodic data, and minimum utilization for nonreal-time messages. The proposed method is shown to be useful by examples.**

***Index Terms*—Maximum allowable delay bound, network-based control system, real time, scheduling method, stability.**

I. Introduction

A NETWORK used at the lowest level of a process/factory communication hierarchy is called a fieldbus. Fieldbuses are intended to replace the traditional cabling between sensors, actuators, and controllers. In distributed control systems, a feedback control loop is often closed through the network, which is called a network-based control system (NBCS). An example of the NBCS is shown in Fig. 1. In the NBCS, various delays with variable lengths are occurred due to sharing a common network medium, which are called network-induced delays. These delays are dependent on configurations of the network and the given system. Those make the NBCS unstable.

Hence it is necessary to develop some methods to make these delays smaller and bounded, which are called scheduling methods for the NBCS. Fig. 2 shows a feedback control system with network-induced delays.

In feedback control systems, it is important that sampled data should be transmitted within a sampling period and stability of control systems should be guaranteed. While a shorter sampling period is preferable in most control systems, for some cases, it can be lengthened up to a certain bound within which stability of the system is guaranteed in spite of the performance degradation. This certain bound is called a maximum allowable delay bound (MADB). The MADB depends only on parameters and configurations of the given plant and the controller. It is noted that the MADB can be obtained from the plant model independent of network protocols, while the network-induced delays depend on network configurations. In addition, a faster sampling is said to be desirable in sampled-data systems because the performance of the discrete-time system controller can approximate that of the continuous-time system. But in NBCSs, the high sampling rate can increase network load, which in turn results in longer delay of the signals. Thus finding a sampling rate that can both tolerate the network-induced delay and achieve desired system performance is important in the NBCS design.

The MADB has been obtained from stability conditions of control systems. There have been some studies on the stability of the NBCS [1]–[3], but those were concerned with obtaining stability conditions of the system with a given delay. In the conventional systems with delay, there have been also some results on MADBs for stability [4]–[6]. In this paper, an MADB is obtained for the stability of the NBCS using the previous results in the conventional systems with delay. The derived MADB is used as a maximum bound of a sampling period in a control loop. That is, the sampling period determined by the proposed sampling period decision algorithm can be a value less than the MADB.

The network in the NBCS should handle three types of data: periodic data (or real-time synchronous data), sporadic data (or real-time asynchronous data), and message (or nonreal-time asynchronous data). All periodic data have to be transmitted within the respective sampling period to guarantee stability of control loops, while guaranteeing real-time transmission of sporadic data and minimum transmission of messages. That is, transmissions of three types of data have to be allocated in the sampling period. This bandwidth allocation method should be included in a scheduling method for the NBCS.

There have been some studies on scheduling methods for the NBCS in fieldbus networks [7]–[12]. In these papers [7]–[11], the MADB and controller delay time were not considered, which were important in control applications. A scheduling algorithm which allocates the bandwidth of a network and determines sensor data sampling periods was presented [12]; each control system had only single input and single output (SISO), only periodic data were considered, and the MADB was not obtained analytically. In [13]–[15], calculation methods of MADBs in networked control systems were presented. But the scheduling methods did not considered three types of data.

This paper presents a scheduling method for the NBCS considering three types of data, which can be used in multiinput and multioutput (MIMO) systems. The presented scheduling

Manuscript received August 31, 1999; revised November 22, 2000. Manuscript received in final form January 8, 2002. Recommended by Associate Editor M. Lemmon.

H. S. Park is with the Department of Electrical and Computer Engineering, Kangwon National University, Chuncheon, Korea (e-mail: hspark@cc.kangwon.ac.kr).

Y. H. Kim is with Teleman Multimedia, Inc., Alviso CA 95002 USA.

D.-S. Kim and W. H. Kwon are with the School of Electrical Engineering, Control Information Systems Lab, Seoul National University, Seoul 151–742 Korea.

Publisher Item Identifier S 1063-6536(02)03422-X.

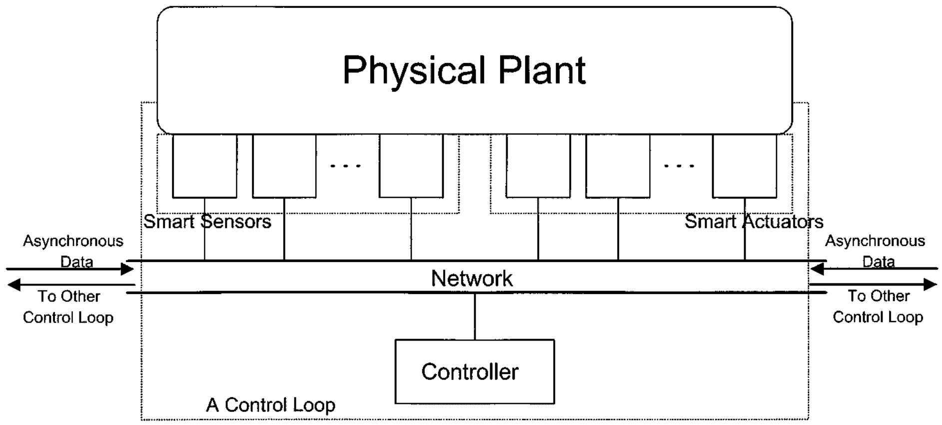

Fig. 1. A typical diagram of a network-based control system.

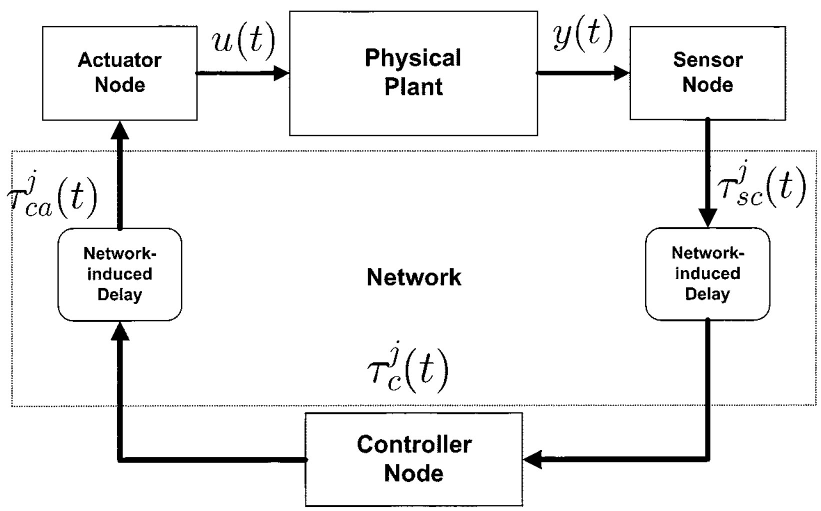

Fig. 2. A feedback control loop with network-induced delays.

method for the NBCS can adjust the sampling period as small as possible, allocate the bandwidth of network for three types of data, and the transmission order of data for sensors and actuators. In addition to those, the presented method can guarantee real-time transmission of sporadic and periodic data, and minimum transmission for messages. It is noted that the MADB is considered as a reference value of the sampling period used in the presented scheduling method. If the MADBs obtained by Lemmas 1 or 2 do not meet the requirements of scheduling, Lemma 3 or other methods should be used.

This paper is organized as follows. In Section II, the NBCS are analyzed and an MADB for stability of the NBCS is derived. In Section III, a scheduling method for the NBCS is presented, which allocates the bandwidth and determines the sampling period for the NBCS using the MADB. In Section IV, examples are given to show that the presented method is useful. Finally, the conclusions are given in Section V.

II. A Maximum Allowable Delay Bound for Stability in a Single Control Loop

In Fig. 2, many control loops can be connected using a single network medium. To simplify the analysis, the following notations are defined.

- P is the total number of loops that use the same medium.
- N^i_α is the number of α nodes in the ith loop. N, N_α, and N^i are the total number of nodes in the NBCS, the total number of α nodes in the NBCS, and the total number of nodes in the ith loop, respectively. Hereinafter, α can be C (controller), A (actuator), or S (sensor).
- T^j is a sampling period of the jth loop.
- $T^j_{\alpha_i}$ is the data transmission time of periodic data in the ith α node in the jth loop.
- T_β is an interval for transmission of β data or messages. Hereinafter, β can be P (periodic data), S (Sporadic Data), or N (messages).
- N^M_S is the maximum number of sporadic data which arrived during a basic sampling period. The basic sampling period means the minimum sampling period in all loops.
- T_{O_β} is the maximum overhead time to transfer β data or a message packet.
- T^j_D is the MADB in the jth loop.

In the above definitions, N^M_S should be integer and can be obtained from the maximum arrival rate of sporadic data in a basic sampling period.

The maximum overhead time (T_{O_β}) can be time-varying in some network protocols (for example, token control). T_{O_β} consists of a message overhead time ($T_{O_\beta}^M$) and a protocol overhead time ($T_{O_\beta}^P$). The message overhead time occurs because of buffering, packetizing, and transmission of additional data frames such as addressing fields, control fields, or a frame check sequence. The protocol overhead time occurs because of various medium access control methods such as polling or token passing. In additions, each overhead time can have different values according to periodic data, sporadic data, and messages. Hence the overhead time used in this paper is classified as $T_{O_\beta}^\gamma$, where γ can be M (message) or P (protocol), and β is given in the definition of notations.

For ease of explanations, the additional overhead time T_O^x is considered. T_O^x consists of the time for synchronization and the maximum overhead time of sums of overhead times consumed by nodes that do not transmit their data during the given finite period. The latter occurs mainly in the token passing mechanism.

The MADB is defined as the maximum allowable interval from the instant when sensor nodes sample sensor data from a plant to the instant when actuators output the transferred data to the plant. If the sampling period in the jth loop exceeds the given MADB, then stability of the overall system could not be guaranteed. In this case, the output of the plant could deviate from the desired trajectory, or the controlled system could behave in an unpredictable manner. Hence it is necessary to derive the MADB from the parameters and configurations of the given plant and the controller.

A. Maximum Allowable Delay Bound in Continuous-Time System

A plant in a single control loop j can be described in the following state-space form:

$$\begin{aligned}\dot{x}_p^j(t) &= F_p^j x_p^j(t) + G_p^j u_p^j(t)\\ y_p^j(t) &= H_p^j x_p^j(t)\end{aligned} \tag{1}$$

where $u_p^j(t) \in R^{N_A^j}$, $y_p^j(t) \in R^{N_S^j}$, $x_p^j(t) \in R^{D_{NP}^j}$. D_{NP}^j is the dimension of the plant in the control loop j. F_p^j, G_p^j, and H_p^j are matrices or vectors of appropriate sizes. A controller in the control loop j can be described by

$$\begin{aligned}\dot{x}_c^j(t) &= F_c^j x_c^j(t) + G_c^j u_c^j(t)\\ y_c^j(t) &= H_c^j x_c^j(t-\tilde{\tau}_c^j) + E_c^j u_c^j(t-\tilde{\tau}_c^j)\end{aligned} \tag{2}$$

where $u_c^j(t) \in R^{N_S^j}$, $y_c^j(t) \in R^{N_A^j}$, $x_c^j(t) \in R^{D_{NC}^j}$ and D_{NC}^j is the dimension of the controller in the control loop j. F_c^j, G_c^j, H_c^j and E_c^j are matrices or vectors of appropriate sizes. $\tilde{\tau}_c^j$ is computation time in the controller j, which satisfies $0 \le \tilde{\tau}_c^j \le \tau_{c,\max}^j$, where $\tau_{c,\max}^j$ is the maximum computation time in the controller j. For convenience, the computation time in the controller is treated in the same way as output delay. Because data from the plant to the controller and from the controller to the plant are transferred through the common communication network, communication delays exist. The communication delays in the control loop j are modeled as

$$\begin{aligned}u_c^j(t) &= y_p^j(t-\tilde{\tau}_{sc}^j)\\ u_p^j(t) &= y_c^j(t-\tilde{\tau}_{ca}^j)\end{aligned} \tag{3}$$

where $0 \le \tilde{\tau}_{sc}^j \le \tau_{sc,\max}^j$, $0 \le \tilde{\tau}_{ca}^j \le \tau_{ca,\max}^j$, $\tilde{\tau}_{sc}^j$ and $\tau_{sc,\max}^j$ are communication delay and maximum communication delay from sensors to a controller, respectively, and $\tilde{\tau}_{ca}^j$ and $\tau_{ca,\max}^j$ are communication delay and maximum communication delay from a controller to actuators, respectively.

Using (1)–(3), a control system in the control loop j can be described as

$$\begin{aligned}\dot{x}^j(t) = &\begin{bmatrix} F_p^j & 0\\ 0 & F_c^j\end{bmatrix} x^j(t) + \begin{bmatrix} 0 & 0\\ G_c^j H_p^j & 0\end{bmatrix} x^j(t-\tilde{\tau}_{sc}^j)\\ &+ \begin{bmatrix} G_p^j E_c^j H_p^j & 0\\ 0 & 0\end{bmatrix} x^j(t-\tilde{\tau}_{sc}^j-\tilde{\tau}_{ca}^j-\tilde{\tau}_c^j)\\ &+ \begin{bmatrix} 0 & G_p^j H_c^j\\ 0 & 0\end{bmatrix} x^j(t-\tilde{\tau}_{ca}^j-\tilde{\tau}_c^j)\end{aligned} \tag{4}$$

where $x^j(t) = [x_p^{j^T}(t) \;\; x_c^{j^T}(t)]^T$. Then the above equation can be rewritten as

$$\begin{aligned}\dot{x}^j(t) = F^j x^j(t) + F_1^j x^j(t-\tau_1^j) + F_2^j x^j(t-\tau_2^j)\\ + F_3^j x^j(t-\tau_3^j)\end{aligned} \tag{5}$$

where

$$F^j = \begin{bmatrix} F_p^j & 0\\ 0 & F_c^j\end{bmatrix}, \quad F_1^j = \begin{bmatrix} 0 & 0\\ G_c^j H_p^j & 0\end{bmatrix}$$

$$F_2^j = \begin{bmatrix} G_p^j E_c^j H_p^j & 0\\ 0 & 0\end{bmatrix}, \quad F_3^j = \begin{bmatrix} 0 & G_p^j H_c^j\\ 0 & 0\end{bmatrix}$$

$$\begin{aligned}&0 \le \tilde{\tau}_1^j = \tilde{\tau}_{sc}^j \le \tau_{sc,\max}^j = \tilde{\tau}_{1,\max}^j,\\ &0 \le \tilde{\tau}_2^j = \tilde{\tau}_{sc}^j + \tilde{\tau}_{ca}^j + \tilde{\tau}_c^j\\ &\quad \le \tau_{sc,\max}^j + \tau_{ca,\max}^j + \tau_{c,\max}^j = \tilde{\tau}_{2,\max}^j\end{aligned}$$

and

$$0 \le \tilde{\tau}_3^j = \tilde{\tau}_{ca}^j + \tilde{\tau}_c^j \le \tau_{ca,\max}^j + \tau_{c,\max}^j = \tilde{\tau}_{3,\max}^j.$$

Each control loop in the NBCS can be described as in (5) using three types of delays.

The following Lemma 1 can be derived from the existing results on the analysis of delayed systems [5], [16].

Lemma 1: Suppose that $(F^j + \sum_{i=1}^3 F_i^j)$ is asymptotically stable. Then a single control loop (5) in the NBCS is asymptotically stable if

$$\tau < \frac{\sigma}{\delta \sum_{i=1}^{3} \left\| F_i^j \left(F^j + \sum_{i=1}^{3} F_i^j \right) \right\|} \tag{6}$$

where

$$\tau = \tilde{\tau}_{i,\max}^j = \tilde{\tau}_{2,\max}^j, \quad \sigma = \frac{\lambda_{\min}(Q)}{2\lambda_{\max}(P)}, \quad \delta = \left[\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}\right]^{1/2}$$

$\|\cdot\|$ is the matrix norm induced by the vector norm. P, Q are the positive-definite symmetric matrices involved in the following Lyapunov equation:

$$\left(F^j + \sum_{i=1}^{3} F_i^j\right)^T P + P\left(F^j + \sum_{i=1}^{3} F_i^j\right) = -Q. \tag{7}$$

$\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are, respectively, the minimum and maximum eigenvalues of the matrix.

The proof of the above Lemma 1 is given in the Appendix.

This delay bound of each control loop will be used as a major parameter in the sampling period decision algorithm and the bandwidth scheduling algorithm, which are discussed in the next section. The value in the right side of (6) is called the MADB.

B. Maximum Allowable Delay Bound in Discrete-Time System

As an example of extension to the discrete-time case, the discrete-time control of an ideal continuous-time integrator in [3] is considered. The following continuous-time model of the system is considered

$$\dot{x} = u, \quad y = x. \tag{8}$$

The corresponding discrete-time model with sampling period is provided as

$$x(k+1) = x(k) + T_s u(k), \quad y(k) = x(k). \tag{9}$$

where $T_s = t(k+1) - t(k)$ is a sampling period and $t(k)$; $k = 0, 1, 2, \ldots$ are sampling times. Using a binary variable $\tau(k)$ to indicate the presence of the communication delay of length T_s in the kth time step, the static feedback law will be

$$u(k) = -g[(1-\tau(k))y(k) + \tau(k)y(k-1)]. \tag{10}$$

Hence, the closed-loop system is

$$x(k+1) = [1 - gT_s(1-\tau(k))]x(k) - gT_s\tau(k)x(k-1) \tag{11}$$

and using $z(k) := [x(k), x(k-1)]^T$ as a state vector, state-space form will be

$$z(k+1) = H(\tau(k))z(k) \tag{12}$$

where

$$H(\tau(k)) = \begin{bmatrix} \alpha(k) & -\beta(k) \\ 1 & 0 \end{bmatrix} \tag{13}$$

$$\alpha(k) = 1 - gT_s(1-\tau(k)), \quad \beta(k) = gT_s\tau(k). \tag{14}$$

Let $a := gT_s$ then

$$A_0 = H(0) = \begin{bmatrix} 1-a & 0 \\ 1 & 0 \end{bmatrix}, \quad A_1 = H(1) = \begin{bmatrix} 1 & -a \\ 1 & 0 \end{bmatrix}$$

$$A_c = A_1 - A_0 = \begin{bmatrix} a & -a \\ 0 & 0 \end{bmatrix}.$$

Solving the Lyapunov equation $A_0^T P A_0 - P = -I$, where P is a symmetric positive-definite matrix, we shall obtain $F = A_0^T P A_c + A_c^T P A_0 + A_c^T P A_c$. The following Lemma 2 and Lemma 3 are derived from the analysis of the discrete-time delayed systems in [3].

Lemma 2 [3]: The equilibrium state $z_e = 0$ of the closed system

$$z(k+1) = (A_0 + \tau(k)A_c)z(k) \qquad (k = 0, 1, 2, \ldots) \tag{15}$$

is exponentially stable for all possible sequences $(\tau(k); k = 0, 1, 2, \ldots)$ of one-time-step delay occurrence, if the control gain K is such that

$$\lambda_{\max}(Q^{-1}G) < 1 \tag{16}$$

where $\lambda_{\max}$ denotes the maximal eigenvalues of the indicated matrix, Q is a symmetric positive–definite matrix, and $G := F(P)$.

Let us assume that $\tau_k := 0, 1, 2, \ldots$ is Markov chain with two states S_1 and S_2 corresponding to the realization of the binary random variable τ_k, and the transition probability matrix at time k given by

$$\Pi_k = \begin{bmatrix} p_{11}(k) & p_{12}(k) \\ p_{21}(k) & p_{22}(k) \end{bmatrix} \tag{17}$$

where $p_{ij}(k)(i, j = 1, 2; k = 0, 1, 2, \ldots)$ are the conditional probabilities that the chain being in the state S_j at time $k-1$ will jump to the state S_i at time k. Hence, the $p_{ij}(k)$ satisfies the constraints

$$0 \le p_{ij}(k) \le 1, \quad p_{1j}(k) + p_{2j}(k) = 1$$
$$(i, j = 1, 2; k = 0, 1, 2, \ldots). \tag{18}$$

The probabilities that the chain will find itself in the states S_1 and S_2 at time $k = 0, 1, 2, \ldots$ are denoted as $p_1(k)$ and $p_2(k) = 1 - p_2(k)$ $(k = 0, 1, 2, \ldots)$ and are determined by

$$p(k+1) = \Pi(k)p(k) \tag{19}$$

where $p(k) := [p_1(k), p_2(k)]^T$. We assume that $p_0 := [1, 0]^T$. The states $z(k)$ in (12) given by

$$z(k+1) = H(\tau(k))z(k) \tag{20}$$

where

$$H(\tau(k)) = A_0 + \tau(k)A_c \tag{21}$$

are in this case a discrete-time random process, too. It is assumed that the initial z_0 is nonrandom.

Lemma 3 [3]: When the Markov chain is homogeneous in time, i.e., when $p(k) = p = \text{constant}$, the zero state of the jump system and is exponentially stable in the mean-square sense if and only if the spectral radius of the matrix

$$L = \begin{bmatrix} p_{11}(A_0 \otimes A_0) & p_{12}(A_0 \otimes A_0) \\ p_{21}(A_1 \otimes A_1) & p_{22}(A_1 \otimes A_1) \end{bmatrix} \tag{22}$$

is less than one, where $\otimes$ is the Kronecker product operator.

The MADB in a discrete-time system is a maximum sampling time obtained from Lemma 2. The MADB obtained from Lemmas 1 and 2 can be conservative because it is derived from their sufficient conditions. The MADB obtained from Lemma 3 can be less conservative, but Lemma 3 is more complex than Lemma 1 and Lemma 2. So if the MADB obtained from Lemma 1 or Lemma 2 is not enough to guarantee transmissions of all types of data, a new MADB should be obtained from Lemma 3. If the new MADB is not enough to do it, the amount of transmission of data should be reduced not much influencing the system performance. This delay bound or MADB of each control loop will be used as a major parameter or a reference value in the sampling period decision algorithm and the bandwidth scheduling algorithm.

C. Examples for an MADB in a Control Loop

As an analysis of the MADB, the following state space of the plant [15] is considered:

$$\begin{aligned}\dot{x}(t) &= A_c x(t) + B_c u(t)\\ y(t) &= C_c x(t)\end{aligned} \tag{23}$$

where

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & -0.1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \quad \text{and} \quad C_c = [1 \quad 0].$$

The MADB is 53.8 ms from Lemma 1. Let us apply Lemma 2 for the MADB. Sampling time $T_s = 4$ ms and $K = 10$ are used for the stability test. A discrete equivalent of the plant represented by the above equation is

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k)\\ y(k) &= Cx(k)\end{aligned} \tag{24}$$

where

$$A = \begin{bmatrix} 1 & 0.004 \\ 0 & 0.9960 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.0004 \end{bmatrix} \quad \text{and} \quad C = [1 \quad 0].$$

Without delay, the closed-loop system is

$$\begin{aligned}H(\tau(k)) = A_0 &= \begin{bmatrix} A & -BK \\ CA & -CBK \end{bmatrix}\\ &= \begin{bmatrix} 1.00 & 0.004 & 0 \\ 0 & 0.9960 & -0.004 \\ 1 & 0.004 & 0 \end{bmatrix}\end{aligned} \tag{25}$$

where $H(\tau(k))$ is from [3]. On the other hand, with one-step delay we have

$$\begin{aligned}H(\tau(k))_{delay} = A_1 &= \begin{bmatrix} A & -BK \\ C & 0 \end{bmatrix}\\ &= \begin{bmatrix} 1.00 & 0.004 & 0 \\ 0 & 0.9960 & -0.004 \\ 1 & 0 & 0 \end{bmatrix}\end{aligned} \tag{26}$$

where $H(\tau(k))_{\text{delay}}$ is from [3].

In the case of the sampling times: $T_s = 4$ ms, 5 ms, 6 ms, 6.5 ms, and $K = 10$, $\lambda(A_0)_{\max} < 1$, $\lambda(A_1)_{\max} < 1$, and $\lambda(A_0A_1)_{\max} < 1$. Therefore, for all sequences, the system is stable. The sampling time $T_s = 6.7$ ms and $K = 10$. $\lambda(A_0)_{\max} < 1$, $\lambda(A_1)_{\max} > 1$, and $\lambda(A_0A_1)_{\max} < 1$. These results are summarized in Table I. The MADB can be obtained as the smaller value than $T_s = 6.7$ ms in the discrete-time analysis by Lemma 2. The MADB is obtained as 53.8 ms from Lemma 1 in the continuous-time analysis. The MADBs obtained from methods in [13] and [14] are 0.27 ms and 0.45 ms, respectively. In this example, the MADB from the continuous-time analysis is less conservative than one from the discrete-time analysis.

III. Sampling Period Decision Algorithm and Bandwidth Scheduling Algorithm in Multiple Control Loops

For simplicity, let the loop number with the smallest MADB be 1, and let us renumber all loops according to the magnitude of the MADB. That is, the smaller the MADB of a loop is, the lower its loop number is. Note that this minimum sampling period T^1 is considered as a basic sampling period. A basic sampling period consists of T_P, T_S, and T_N as shown in Fig. 3. In addition to the three periods (T_P, T_S, and T_N), there can be a synchronization period. The synchronization period is not mainly considered in this paper, but it is included in T_O^x.

TABLE I
STABILITY OF SEQUENCES BY LEMMA 2

Sampling(t)	A_0	A_1	A_0A_1	*Other*
$4ms$	Stable	Stable	Stable	Stable
$5ms$	Stable	Stable	Stable	Stable
$6ms$	Stable	Stable	Stable	Stable
$6.5ms$	Stable	Stable	Stable	Stable
$6.7ms$	Stable	*Unstable*	Stable	*Unstable*

The following assumptions are used in this paper.

- Sampling time of sensors in a loop is synchronized at the starting instant of basic sampling periods.
- In networks, communications are error-free. That is, there are no failures in transferring messages.
- Packets transferred from sensors to controllers or controllers to actuators have the same length.
- Control actions of one control loop do not affect other control loops.
- Sampling periods of each loop are adjusted as multiples of the smallest sampling period (T^1) in the order 2 (e.g., T^1, $2 \times T^1$, $4 \times T^1$, $8 \times T^1 \cdots$) and should not exceed the MADB in the corresponding loop.
- Controller delay time such as the computation time for control values can be obtained and is less than or equal to the transmission time of a control value or a sensor value.
- Deadlines of all sporadic data are the smallest sampling period (T^1).

The fifth assumption is introduced to simplify the algorithm. Under this assumption multiples of the smallest sampling period can be used as the sampling periods of loops and the least common multiple (LCM) of the sampling periods of all loops can be used as the largest period.

The sixth assumption is used for the simple analysis. In a real environment, the following sequence for control of plants is used: reading of sensor, transmission of the sensor data, computation of the control value using the received sensor data, and its transmission to the actuator. If the controller delay time or the computation time in the controller is overlapped with the sensor (or actuator) data transmission time, the analysis becomes simple since the controller delay time is not considered. Otherwise, the controller delay time should be considered. This means that an idle time in the usage of the network is necessary before the control value is transmitted after receipt of the sensor data. Strictly speaking, the idle time, an example of which is the computation delay time minus the sensor data transmission time, should be calculated. That makes the analysis difficult.

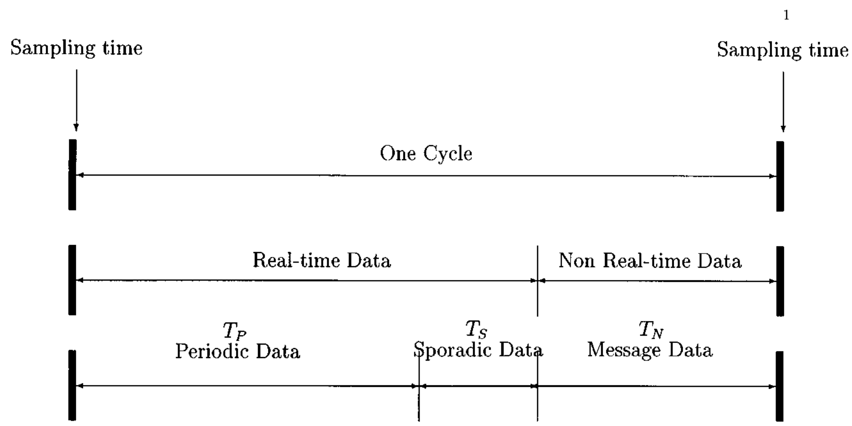

Fig. 3. Configuration of phases.

The last assumption is introduced to simplify the schedulability condition. Deadlines of sporadic data can be larger than the smallest sampling period (T^1). If it is transmitted within T^1, it can be said that the deadline is satisfied. It is noted that this assumption is very strict.

In the NBCS, if the controller delay time is considered, it is difficult to allocate the bandwidth of nodes since the controller delay time is required to compute control values in a loop only after all sensor values in the loop are received. Due to this problem the following rule is used.

Sensor nodes use the medium prior to a controller node in the loop. After transmitting all sensor values in the loop, sensor values in the next higher loop are transmitted by overlapping of controller delay time and sensor data transmission time. Then the control values are transmitted immediately without any idle time after the transmission of the sensor data in the higher loop.

In the worst case or the practical case, one controller delay time cannot overlap the sensor data transmission time though the above rule is used.

Now, let's calculate the time needed for a certain basic sampling period. Utilization of messages in a basic sampling period denoted by U_N can be represented as

$$U_N = \frac{T_N}{T^1}. \tag{27}$$

To guarantee the minimum messages, which is denoted by U_N^m, the following inequality:

$$U_N^m \leq U_N \tag{28}$$

should be satisfied. Using Equation (27), the above equation is converted to

$$U_N^m \cdot T^1 \leq T_N. \tag{29}$$

This period for messages (T_N) includes the overhead time ($T_{O_N}^M$ and $T_{O_N}^P$).

To transmit all sporadic data which arrived during the previous cycle, the following condition:

$$N_S^M \cdot \hat{T}_S^M \leq T_S \tag{30}$$

should be satisfied, where $\hat{T}_S^M = T_S^M + T_{O_S}$, T_S^M is the maximum value of data transmission time of sporadic data in the basic sampling period, and $\lceil Z \rceil$ is the smallest integer larger than or equal to the value Z.This means that $N_S^M \cdot \hat{T}_S^M$ is the maximum value of T_S in the basic sampling period during which all the sporadic data are transmitted.

A basic sampling period consists of sampling delay, transmission time of periodic data, transmission time of sporadic data, and transmission time of messages. Considering one specific basic sampling period, it can be written as

$$\begin{aligned} T^1 = &(D_{S_i}^j) + \sum_{j \in U_L} \sum_{i=1}^{N_S^j} (T_{S_i}^j + T_{O_P}) \\ &+ \sum_{i \in U_L} \sum_{k=1}^{N_A^i} (T_{C_k}^i + T_{O_P}) + \sum_{j \in U_L^*, i \in U_S^*} (T_{S_i}^j + T_{O_P}) \\ &+ T_S + T_N + T_O^x + T_{CD}. \end{aligned} \tag{31}$$

where T_{CD} denotes controller delay time, U_L denotes a set of loops whose all nodes are included in the considered basic sampling period, U_L^* denotes a set of loops whose all nodes are not included in the considered basic sampling period but some of the nodes in those loops are partly in the considered basic sampling period, and U_S^* denotes a set of sensors which are in U_L^*. $(D_{S_i}^j)$ is the largest required time to start transmitting sensor data, which can be shortened in some network protocols using an adequate scheduling method. Let N_P be the number of sensor and actuator data packets for periodic data in U_L, U_S^*, and U_L^*. Then N_P is given by

$$N_P = N_S^b + N_A^b \tag{32}$$

where N_A^b denotes the number of actuators in U_L and N_S^b denotes the number of sensors in both U_L and U_S^*. Let

$$\begin{aligned} T_P = &\sum_{j \in U_L} \sum_{i=1}^{N_S^j} (T_{S_i}^j + T_{O_P}) \\ &+ \sum_{j \in U_L} \sum_{i=1}^{N_A^j} (T_{C_i}^j + T_{O_P}) \\ &+ \sum_{j \in U_L^*, i \in U_S^*} (T_{S_i}^j + T_{O_P}). \end{aligned} \tag{33}$$

Then the basic sampling period is bounded as the following equation:

$$T_P + N_S^M \cdot \hat{T}_S^M + U_N^m \cdot T^1 + T_O^x + T_{CD} \le T^1. \quad (34)$$

The above equation can be changed to

$$T_P \le \left\langle (1 - U_N^m) \cdot T^1 - N_S^M \cdot \hat{T}_S^M - T_O^x - T_{CD} \right\rangle \quad (35)$$

where

$$\langle x \rangle = \begin{cases} x & \text{for } x > 0 \\ 0 & \text{for } x \le 0. \end{cases}$$

Note that the NBCS cannot be scheduled if T_P is less than or equal to zero. In this case, other high-speed network protocols should be selected or the number of nodes should be reduced. If the data transmission times of sensors are equal as M_S and the data transmission times of controllers are equal as M_C, using (34). The above equation becomes

$$\begin{aligned} &N_S^b(M_S + T_{O_P}) + N_A^b(M_C + T_{O_P}) \\ &\quad \le \left\langle (1 - U_N^m) \cdot T^1 - N_S^M \cdot \hat{T}_S^M - T_O^x - T_{CD} \right\rangle. \end{aligned} \quad (36)$$

The left-hand side is the period for the periodic data (T_P) and is bounded by the right-hand side.

Now consider the schedulability. If data transmission times of sensors and controllers are equal as M, then (36) becomes

$$N_P \le \left\lfloor \frac{\left\langle (1 - U_N^m) \cdot T^1 - N_S^M \cdot \hat{T}_S^M - T_O^x - T_{CD} \right\rangle}{M + T_{O_P}} \right\rfloor \quad (37)$$

where $\lfloor Z \rfloor$ is the largest integer smaller than or equal to the value Z. Let the right part of (37) be N_P^M which indicates the maximum integer of N_P.

Let the largest sampling period in the NBCS be $N_E \cdot T^1$ (i.e., $N_E = T^P/T^1$) and the ith basic sampling period in the largest sampling period be T_B^i. Let the number of sensor and actuator data packets for periodic data during $N_E \cdot T^1$ be N_P^T (that is, total number of transmissions of periodic data during the interval $N_E \cdot T^1$). Then it can be calculated as

$$N_P^T = \sum_{j=1}^{P} Q(N^j) \cdot N^j \quad (38)$$

where $Q(N^j) = (N_E \cdot T^1)/T^j$ for $j = 1, \ldots, P$. Since T^j for $j = 1, \ldots, P$ are adjusted as multiples of T^1 in the order 2, $Q(N^j)$ for $j = 1, \ldots, P$ have integer values. The schedulability can be checked by comparing N_P^T with $N_P^M \cdot N_E$.

The largest sampling period in the NBCS depends on the largest MADB (T_D^P).

The following sampling period decision algorithm based on the bisection method can decide the basic sampling period.

1. Set the MADB of each control loop using Lemma 1 or Lemma 2 (or Lemma 3).
2. Reorder control loops according to the MADBs such that the smaller the MADB of a loop is, the lower its loop number is.
3. Compute N_P^T using (38) and the results of the above step.
4. Let $T^1 = T_D^1$, $T_L = 0$, and $T_L = T^1$. $k = 0$.
5. Choose T^j such as $T^j \le T_D^j$ and $T^j = \max(2^k \cdot T^1)$ for $k = 0, 1, 2, \ldots$.
6. Compute N_P^M using (37).
7. If $\lceil N_P^T/N_E \rceil$ is equal to N_P^M or ($N_P^M - \lceil N_P^T/N_E \rceil$) is within a given bound, then $T_B^j = T^j$ for $j = 1, \ldots, P$ and go to the next step,
   else if $\lceil N_P^T/N_E \rceil$ is less than N_P^M, then $T_U = T^1$, take the basic sampling period (T^1) as $(T_L + T_U)/2$, $k{+}{+}$, and go to step 5),
   else if $\lceil N_P^T/N_E \rceil$ is greater than N_P^M and $k = 0$, then terminate the algorithm (the scheduling is failed),
   else if $\lceil N_P^T/N_E \rceil$ is greater than N_P^M and $k \ne 0$, then $T_L = T^1$, take the basic sampling period (T^1) as $(T_L + T_U)/2$, $k{+}{+}$, and go to step 5.
8. For each basic sampling period, $T_B^j (j = 1, \ldots, N_E)$, allocate the bandwidth for sensor nodes and actuator nodes using the bandwidth scheduling algorithm.

The following is the bandwidth-scheduling algorithm.

Set $N_S^h = N_S^h$, $N_A^h = N_A^h$ for $1 \le h \le P$, and $N_S^h = N_A^h = 0$ for $h > P$,
for $l = 1$ to $l = N_E$ do,
  set $S_N = N_P^M$ (number of allocatable data packets),
  set $i = j = 1$,
  read the sensor values in loop i,
  $S_N = S_N - N_S^i$, $N_S^i = 0$, and $i{+}{+}$,
  repeat
  while ($N_S^i == 0$ and $i \le P$),
    $i{+}{+}$,
  end of while,
  while ($N_A^j == 0$ and $j \le P$),
  $j{+}{+}$,
  end of while,
  if $S_N \ge (N_A^j + N_S^i)$,
read all sensor values in the ith loop,
write all actuator values in the jth loop,
$S_N = S_N - N_S^i - N_A^j$, $N_S^i = N_A^j = 0$, $i{+}{+}$, $j{+}{+}$,
  else if $N_A^j < S_N < (N_A^j + N_S^i)$,
  if $j < i$,
$S_N = S_N - N_A^j$,
read S_N sensor values in the ith loop,
write all actuator values in the jth loop,
$N_S^i = N_S^i - S_N$, $S_N = N_A^j = 0$, $j{+}{+}$,
  else,
read $\min(S_N, N_S^i)$ sensor values in the ith loop,

$N_S^i = \langle N_S^i - S_N \rangle$, $S_N = \langle S_N - N_S^i \rangle$,
if $N_S^i = 0$, then $i++$,
endif,
else if $S_N \le N_A^j$ and $S_N \ge N_S^i$,
read all sensor values in the ith loop,
$S_N = S_N - N_S^i$, $N_S^i = 0$, $i++$,
write S_N actuator values in the jth loop,
$N_A^j = N_A^j - S_N$, $S_N = 0$,
else if $S_N \le N_A^j$ and $S_N < N_S^i$,
if $(i - j) < 2$,
read S_N sensor values in the ith loop,
$N_S^i = N_S^i - S_N$, $S_N = 0$,
else,
write S_N actuator values in the jth loop,
$N_A^j = N_A^j - S_N$, $S_N = 0$,
endif,
endif,
until ($S_N = 0$ or $j > P$),
$m = 1$,
while $(((\lceil (l \cdot T^1)/T^m \rceil - \lfloor (l \cdot T^1)/T^m \rfloor) == 0)$
and $(m \le P))$,
$N_S^m = N_S^m$, $N_A^m = N_A^m$, $m++$,
end of while,
end of for loop.

Using the bandwidth-scheduling algorithm, data packets are allocated as follows. First, sensor data packets in the loop 1 are transmitted to the corresponding controller through the network medium. When all sensor data packets in the loop 1 are transmitted, computations of control values in the controller of the loop 1 are started. This is the controller delay time in the loop 1. During this controller delay, sensor data packets in the next loop (the loop 2) are transmitted using the network medium.

So the controller delay is overlapped with the transmission time of sensor data in the loop 2. After the controller delay time, a controller of the loop 1 transmits its data to actuators. After the transmission of the controller, data packets of other nodes are scheduled in the same method as above during the specified period for periodic data. If the period for periodic data in the basic sampling period is ended, data packets for sporadic data are scheduled. If the time for messages is left after the transmissions of all sporadic data, then data packets for messages are scheduled. Before the basic sampling period is ended, an interval for synchronization could exist according to applications. If there are unallocated nodes in other loops after the first basic sampling period, the unallocated nodes in other loops are scheduled in the next basic sampling period in the same method as above. The smallest period, which contains the period for periodic, sporadic data, and messages (if possible) less than or equal to MADB. It is selected as a minimum sampling period of the loop 1 according to the sampling period decision algorithm.

If the bandwidth being able to transmit all data packets in all MADBs of loops cannot be allocated, other high-speed network protocols should be selected or the number of nodes should be reduced. These two algorithms (the sampling period decision algorithm and the bandwidth scheduling algorithm) are presented as the scheduling method in this paper. This scheduling method is based on the earliest deadline first (EDF) algorithm [17], [18]. However, it is modified for applications to control the loops containing real-time and nonreal-time data.

Let us consider some points in applications of the scheduling method for the NBCS in the case of token control networks such as PROFIBUS. The worst overhead (for example, the overhead time T_O^x) should be reserved in case the token has been passed over to the next station just before a transmission request is made. In token control, the overhead time (which includes token passing time) occupies a large part of the whole period and synchronizations are very difficult. The overhead time can be varied according to the order in which the token is passed. If the order of passing the token is not adjusted appropriately, a node may have to wait while the token is passed over all the other nodes. Because the address of each node is related to the order in which the token is passed, an address of each node has to be adjusted. This can be done using the previous scheduling algorithm for the NBCS. For example, if a node α is scheduled before other node β, then the node α should have the token prior to the node β and have smaller number of address than that of the node β. By an appropriate selection of the order of nodes, T_O^x in the period for periodic data can be bounded by one token rotation time. To prevent unsent data or message accumulations in the node and give the right for the transmissions of data or message to all nodes in one basic sampling period, the target token rotation time should be much less than the basic sampling period.

Now let's consider applications of the scheduling method in case of the polling control network such as field instrumentation protocol (FIP). As there is no need to wait for the token, the data or messages are transmitted after the sensor delay time. T_O^x is needed only for synchronization. Hence if the synchronization period is not considered, T_O^x can be zero. However, in polling control such as FIP, considerable overhead time (T_{O_S}) is required for sporadic data, since two or more transfers of data packets such as the sporadic data request frame and its corresponding frame from the bus arbitrator are required. The rest of the procedure is very similar to the case of the token control network.

IV. SIMULATIONS

A. Examples for MADB

As an example for verification of the proposed method, a plant with six dc motors is considered. Each motor has an armature position controller with two sensors and one actuator, which are linked via the network. This configuration of six motors can be assumed to be part of a robot. If the armature inductance (L_a) and viscous frictional coefficient (B_m) are negligible, the motor dynamics can be modeled by

$$\dot{x}_p = F_p x_p + G_p u_p = \begin{bmatrix} -K_i K_b / R_a J & 0 \\ 1 & 0 \end{bmatrix} x_p + \begin{bmatrix} K_i / R_a J \\ 0 \end{bmatrix} u_p \quad (39)$$

$$y_p = x_p \quad (40)$$

where $x_p = [\omega \quad \theta]^T$, u_p is the applied voltage (V), and ω and θ are, respectively, the rotor angular velocity (rad/s) and displacement (rad). R_a, K_i, K_b, and J represent the armature

TABLE II
MADB IN THE EACH LOOP

Control Loop	$R_a(\Omega)$	Lemma 1	Lemma 2
LoopA	10	1.4	0.2
LoopB	13	2.6	0.3
LoopC	14	3.1	0.3
LoopD	19	6.1	0.4
LoopE	21	7.5	0.4
LoopF	25	10	0.4

resistance, the torque constant, the back emf constant, the inertia of rotor and load, respectively. If a constant gain (K) is used as a state feedback controller, the equation (5) is changed to

$$\dot{x}_p(t) = F_p x_p(t) + G_p * K x_p(t-\tau) \tag{41}$$

as a single control loop in the NBCS, where $\tau = \tilde{\tau}_c + \tilde{\tau}_{sc} + \tilde{\tau}_{ca}$.

For simulations, the motor in each loop has the nominal values such that $R_a = 10$ (Ω), $K_i = 10$ (oz-in/A), $K_b = 0.075$ (V/rad/s), and $J = 0.006$ (oz-in-s^2). The tested motors in each loop have the same nominal values as the previous one except R_a. Other motors have the values of $R_a = 13$ (Ω), $R_a = 14$ (Ω), $R_a = 19$ (Ω), $R_a = 21$ (Ω) and $R_a = 25$ (Ω), respectively.

Using Lemma 1 and the given parameters of the motors, the MADBs are calculated as 1.4, 2.6, 3.1, 6.1, 7.5, and 10 ms in Table II. Using Lemma 2 and the given parameters of the motors, the MADBs are calculated as 0.2, 0.3, 0.3, 0.4, 0.4, and 0.4 ms in Table II. Hence the final MADB is given as 1.4 ms. From these two examples, it can be known that MADB from the continuous-time analysis is less conservative than one from the discrete-time analysis.

B. Application of a Scheduling Method

For the test of a scheduling method, one motor has the nominal values such that $R_a = 14$ (Ω), $K_i = 10$ (oz-in/A), $K_b = 0.075$ (V/rad/sec), and $J = 0.006$ (oz-in-s^2). Other motors have the same nominal values as the previous one except R_a. Other two motors have the values of $R_a = 19$ (Ω) and $R_a = 21$ (Ω), respectively. Using Lemma 1 and the given parameters of the motors, the MADBs are calculated as 3.1, 6.1, and 7.5 ms. The MADBs of each loop can be set as 3, 6, and 7 ms for convenience of calculations. It is assumed that data for the sensor and the actuator have four bytes. Using the notations in the last section, the followings are given:

$N_C^j = N_A^j = 1$ for $j = 1, 2, 3$;
$N_S^j = 2$ for $j = 1, 2, 3$, $N^j = 3$ for $j = 1, 2, 3$;
$N^* = 1$, $P = 3$, $N = \sum_{j=1}^{P}(N_C^j + N_S^j) + N^* = 10$;
$D_{S_i}^j = 0.1$ ms for $i = 1, 2$, $j = 1, 2, 3$;
$D_{C_1}^j = 0.1$ ms for $j = 1, 2, 3$;
$T_D^1 = 3$ ms, $T_D^2 = 6$ ms, $T_D^3 = 7$ ms;
$N_S^M = 2$;
$U_N^m = 0.16$;

where N^* is total number of extra nodes which do not participate in control loops in the NBCS. $D_{S_i}^j$ or $D_{C_i}^j$ is the ith sensor or controller delay in the jth loop, respectively.

Because the actuator nodes are assumed not to send any data in normal operations, the transmission of all actuator nodes are not considered. The transmission speeds varies according to the given network protocols, but in this example the transmission speed is assumed to be 1 Mbps, regardless of the given network protocols, for an equal comparison between the polling control and the token control network.

The data length of sensors and controllers is assumed to be four bytes and that of sporadic data is assumed to be two bytes. For simplicity of an analysis, it is assumed that buffering delays and packetizing delays are neglected.

First, let us consider process field bus (PROFIBUS). If the universal asynchronous receiver and transmitter (UART) character which consists of 11 bits/byte is used in the token control, the parameters can be given as follows:

$$\begin{aligned} T_{S_i}^j &= T_{C_1}^j = M = 4 \text{ bytes} \times 11 \text{ bits/byte} \times 1\,\mu\text{s/bit} \\ &= 44\,\mu\text{s}, \text{ for } i = 1, 2,\ j = 1, 2, 3 \\ T_S^M &= 2 \text{ bytes} \times 11 \text{ bits/byte} \times 1\,\mu\text{s/bit} = 22\,\mu\text{s}. \end{aligned}$$

Message overhead for periodic and sporadic data

$$T_{O_P}^M = T_{O_S}^M = 9 \text{ bytes} \times 11 \text{ bits/byte} \times 1\,\mu\text{s/bit} = 99\,\mu\text{s}.$$

Protocol overhead for periodic data can be bounded by one token rotation time and given by

$$\begin{aligned} T_{O_P}^P &= 10(N) \times 3 \text{ bytes(token)} \times 11 \text{ bits/byte} \times 1\,\mu\text{s/bit} \\ &= 330\,\mu\text{s}. \end{aligned}$$

Protocol overhead for sporadic data is calculated as

$$\begin{aligned} T_{O_S}^P &= 10(N) \times 3 \text{ bytes(token)} \times 11 \text{ bits/byte} \times 1\,\mu\text{s/bit} \\ &= 330\,\mu\text{s} \\ T_S &= 2(N_S^M) \times (T_S^M + T_{O_S}^M + T_{O_S}^P) = 902\,\mu\text{s} \\ T_O^x &= 330\,\mu\text{s},\ T_N = 0.16 \times 3 \text{ ms} = 0.48 \text{ ms}. \end{aligned}$$

Next, let us consider FIP. The parameters can be given as follows:

$$\begin{aligned} T_{S_i}^j &= T_{C_1}^j = M = 4 \text{ bytes} \times 8 \text{ bits/byte} \times 1\,\mu\text{s/bit} \\ &= 32\,\mu\text{s}, \text{ for } i = 1, 2,\ j = 1, 2, 3 \\ T_{O_P}^M &= 45 \text{ bits}(RP\_DAT) \times 1\,\mu\text{s/bit} = 45\,\mu\text{s} \\ T_{O_P}^P &= 61 \text{ bits}(ID\_DAT) \times 1\,\mu\text{s/bit} = 61\,\mu\text{s} \\ T_S^M &= 2 \text{ bytes} \times 8 \text{ bits/byte} \times 1\,\mu\text{s/bit} = 16\,\mu\text{s} \\ T_{O_S}^M &= 45 \text{ bits}(RP\_DAT) \times 1\,\mu\text{s/bit} = 45\,\mu\text{s} \\ T_{O_S}^P &= \{61(ID\_RQ) + (45+16)(RP\_RQ) \\ &\quad + 61(ID\_DAT)\} \text{ bits} \times 1\,\mu\text{s/bit} = 183\,\mu\text{s} \\ T_S &= 2(N_S^M) \times \left(T_S^M + T_{O_S}^M + T_{O_S}^P\right) = 488\,\mu\text{s} \\ T_N &= 0.16 \times 3 = 0.48 \text{ ms}. \end{aligned}$$

Applying steps 1 to 6 of the sampling period decision algorithm to the example, N_P^T are given as 12 and 2, respectively. N_P^M is 2 in the token control network and 14 in the polling control network. Hence, 2 nodes and 14 nodes can be scheduled in the token control and the polling control network, respectively,

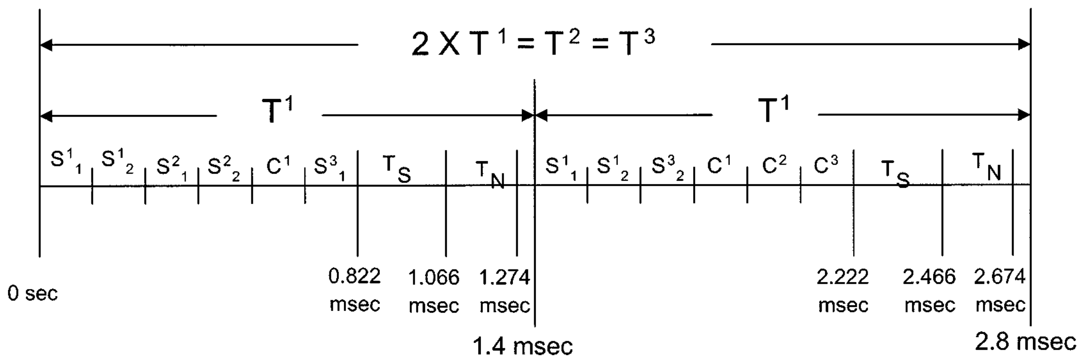

Fig. 4. Bandwidth allocation result using polling control.

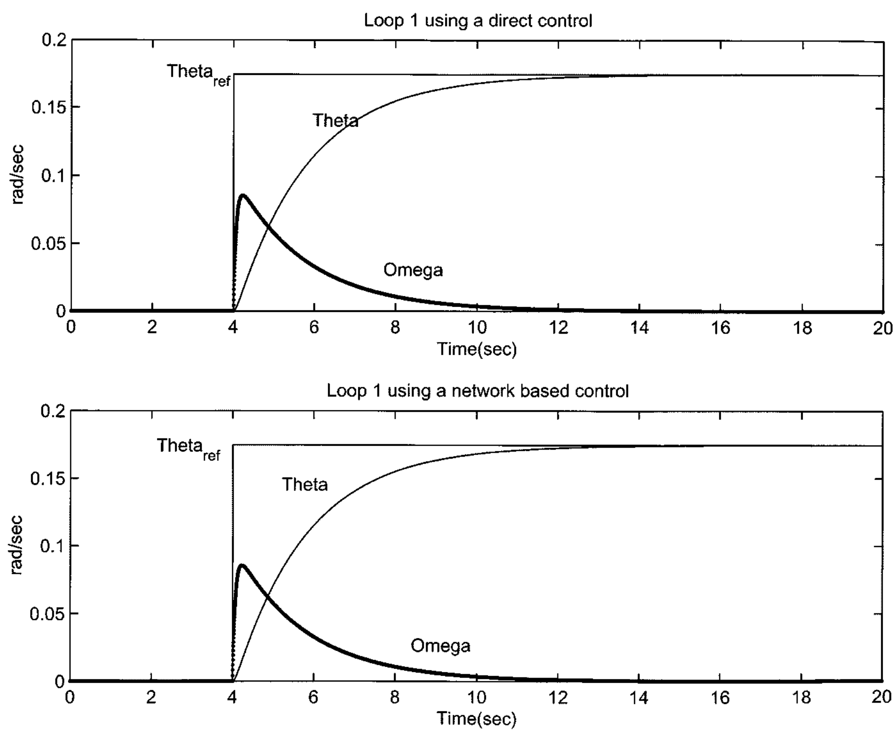

Fig. 5. Outputs of motor 1 position control.

within a basic sampling period. From this calculation, it can be shown that all nodes cannot be scheduled using token control, but can be, using polling control. Then, following the repetition steps of the sampling period decision algorithm, the sampling period can be reduced in case of the polling control network.

As a final step of the scheduling method for the NBCS, bandwidth is allocated using the bandwidth scheduling algorithm. In case of token control, the end time of the transmission in loop 3 exceeds $2 \times T_D^1$ which is the sampling period of loop 3 (from the bandwidth allocation algorithm in the last section). Therefore, scheduling is impossible in this case.

In case of polling control, as the overhead time in each node is bounded by a constant value, calculation results in the last section are very similar to the real values. A basic sampling period ranges from 1.4 ms to 3.0 ms. Hence, the basic sampling period can be reduced to about 1.6 ms in this case. The result from calculations are matched to those of the allocations. If the basic sampling period of 1.4 ms is selected, the bandwidth can be allocated as shown in Fig. 4. The simulation results in case of polling control case are shown in Figs. 5–7. In Figs. 5–7, we show the outputs (ω, θ) of the motor position control system in which a controller, sensors, and an actuator are connected directly or connected by a network. The behavior of the outputs in the NBCS is similar to that in the directly connected systems from Figs. 5–7.

V. Conclusion

The NBCS should satisfy two different sets of characteristics. One is the characteristics of control systems such as stability and sampling period, and the other is the characteristics of network systems such as real-time transmission of sporadic and periodic data, and the minimum network utilization for non-real-time messages. Generally it is difficult for the NBCS to be designed to satisfy all of the above characteristics. In this paper, the MADBs are obtained for the stability of the NBCS, and are used as the basic parameter for a scheduling method for the NBCS. Furthermore, the presented scheduling method for the

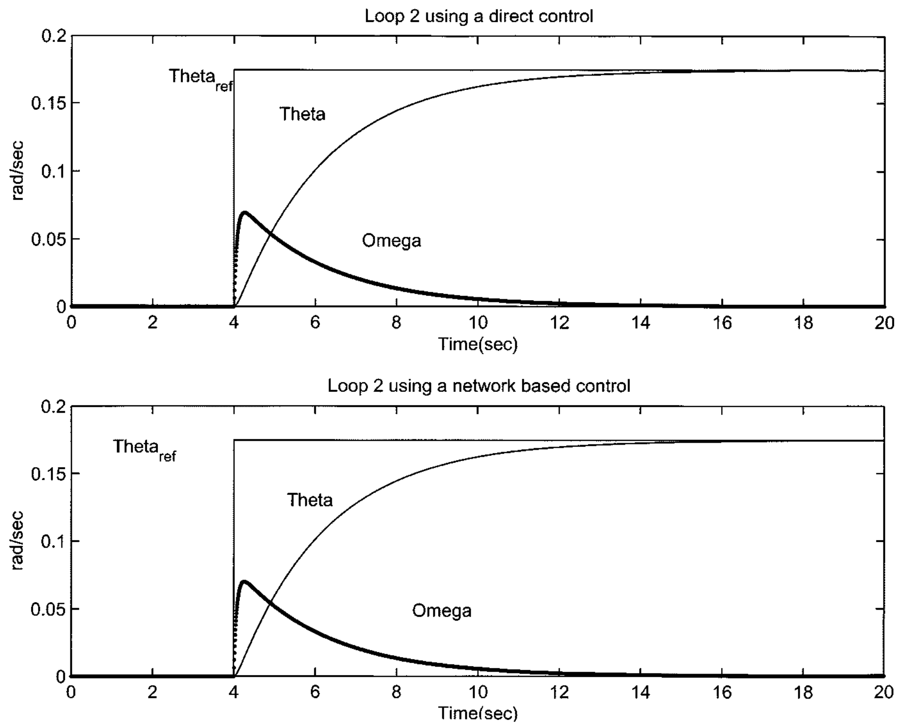

Fig. 6. Outputs of motor 2 position control.

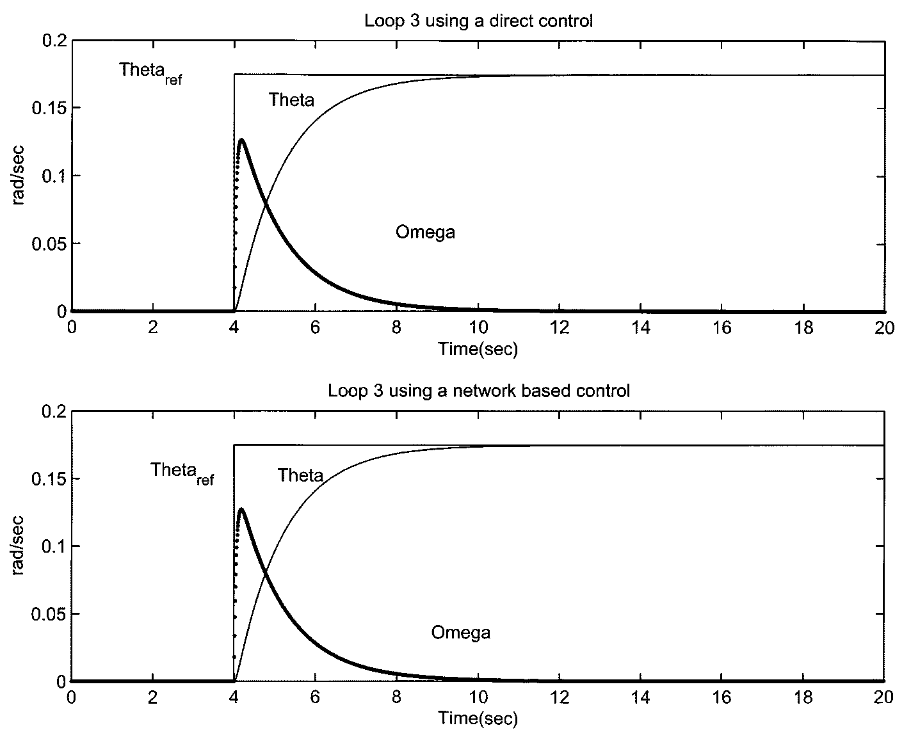

Fig. 7. Outputs of motor 3 position control.

NBCS can adjust the sampling period as small as possible, allocate the bandwidth of the network for three types of data, and exchange the transmission orders of data for sensors and actuators. In addition, the presented method can guarantee real-time transmission of sporadic and periodic data, and minimum utilization for nonreal-time messages.

In the NBCS, the presented method is very useful, as it provides a solution to determine the sampling period of each control loop and it can indicate whether the predetermined network protocol is possible for the given control system or not. An example is presented to show the usefulness of the proposed method for the NBCS.

As the sampling periods used in the proposed method are multiples of each other in the order 2, the simplified algorithm based on multiples of the smallest sampling period is necessary to be studied. As the future works, delays due to packet losses, buffering, and packetizing will be investigated for the NBCS.

Appendix

Proof: This proof is based on [5]. If $x^j(t)$ is the solution of (5), then one can obtain

$$\begin{aligned} x^j(t-\tilde{\tau}_i^j) &= x^j(t) - \int_{-\tilde{\tau}_i^j}^{0} \dot{x}^j(t+\theta)\,d\theta \\ &= x^j(t) - \int_{-\tilde{\tau}_i^j}^{0} \Bigg[F^j x^j(t+\theta) \\ &\qquad + \sum_{i=1}^{3} F_i^j x^j(t-\tilde{\tau}_i^j+\theta)\Bigg]\,d\theta \end{aligned} \tag{42}$$

where $i = 1, 2, 3$. Inserting the above equation into the equation (5) yields

$$\begin{aligned} \dot{x}^j(t) &= F^j x^j(t) + \sum_{i=1}^{3}\Bigg\{F_i^j x^j(t) - F_i^j \int_{-\tilde{\tau}_i^j}^{0} \\ &\quad \cdot \Bigg\{F^j x^j(t+\theta) + \sum_{i=1}^{3} F_i^j x^j(t-\tilde{\tau}_i^j+\theta)\Bigg\}\,d\theta\Bigg\} \\ &= \left(F^j + \sum_{i=1}^{3} F_i^j\right) x^j(t) - \sum_{i=1}^{3}\Bigg\{F_i^j \int_{-\tilde{\tau}_i^j}^{0} \\ &\quad \cdot \Bigg[F^j x^j(t+\theta) + \sum_{i=1}^{3} F_i^j x^j(t-\tilde{\tau}_i^j+\theta)\Bigg]\,d\theta\Bigg\}. \end{aligned} \tag{43}$$

Let

$$\sigma = \frac{\lambda_{\min}(Q)}{2\lambda_{\max}(P)}, \quad \delta = \left[\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}\right]^{1/2}$$

and P, Q be the positive-definite symmetric matrices which satisfy the following Lyapunov equation:

$$\left(F^j + \sum_{i=1}^{3} F_i^j\right)^T P + P\left(F^j + \sum_{i=1}^{3} F_i^j\right) = -Q.$$

$\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the minimum and the maximum eigenvalues of the matrix, respectively. Then take

$$V(x) = \tfrac{1}{2}\, {x^j}^T(t) P x^j(t) \tag{44}$$

as our Lyapunov function and its differentiation becomes

$$\begin{aligned} \dot{V} &= \tfrac{1}{2}\,{\dot{x}^j}^T(t) P x^j(t) + \tfrac{1}{2}\,{x^j}^T(t) P \dot{x}^j(t) \\ &\le -\tfrac{1}{2}\,{x^j}^T Q x^j + \Bigg\| {x^j}^T P \sum_{i=1}^{3} F_i^j \int_{-\tilde{\tau}_i^j}^{0} \Bigg[F^j x^j(t+\theta) \\ &\quad + \sum_{i=1}^{3} F_i^j x^j(t-\tau_i^j+\theta)\Bigg]\,d\theta\Bigg\|. \end{aligned} \tag{45}$$

Let $\tau = \max \tilde{\tau}^j_{i,\max}$. And using the Razumikhin-type theorem [5], (5) is asymptotically stable if

$$\tau < \frac{\sigma}{\delta \sum_{i=1}^{3} \left\| F_i^j \left(F^j + \sum_{i=1}^{3} F_i^j\right)\right\|}. \qquad \blacksquare$$

References

[1] Y. Halevi and A. Ray, "Integrated communication and control systems: Part I—Analysis," *J. Dyn. Syst., Measurement, Contr.*, vol. 110, pp. 367–373, Dec. 1988.
[2] A. Ray and Y. Halevi, "Integrated communication and control systems: Part II—Design consideration," *J. Dyn. Syst., Measurement, Contr.*, vol. 110, pp. 374–381, Dec. 1988.
[3] R. Krtolica, Ü. Özgüner, H. Chan, H. Göktas, J. Winkelman, and M. Liubakka, "Stability of linear feedback systems with random communication delays," *Int. J. Contr.*, vol. 59, no. 4, pp. 925–953, 1994.
[4] T. Mori, N. Fukuma, and M. Kuwahara, "Simple stability criteria for single and composite linear systems with time delays," *Int. J. Contr.*, vol. 34, no. 6, pp. 1175–1184, 1981.
[5] T.-J. Su and C.-G. Huang, "Robust stability of delay dependence for linear uncertain systems," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1656–1659, 1992.
[6] P.-L. Liu and T.-J. Su, "On a sufficient condition for robust stability of delay systems," in *Proc. ACC*, 1995, pp. 328–329.
[7] P. Raja and G. Ulloa, "Priority polling and dynamic time-window mechanisms in a multicycle fieldbus," in *Proc. Compeuro*, 1993, pp. 452–460.
[8] J. P. Beauvais and A.-M. Deplanche, "Heuristics for scheduling periodic complex real-time tasks in a distributed system," in *Proc. IECON'95*, 1995, pp. 55–60.
[9] S. Cavalieri, A. Stefano, and O. Mirabella, "Pre-run-time scheduling to reduce schedule length in the fieldbus environment," *IEEE Trans. Software Eng.*, vol. 21, pp. 865–880, 1995.
[10] K. Tindell, H. Hansson, and A. Wellings, "Analysing real-time communications: Controller area network," in *Proc. IEEE Real-Time Syst. Symp.*, 1994.
[11] P. Lorenz and Z. Mammeri, "Real-time software architecture: Application to FIP fieldbus," in *Proc. AARTC*, 1995, pp. 415–423.
[12] S. H. Hong, "Scheduling algorithm of data sampling times in the integrated communication and control systems," *IEEE Trans. Contr. Syst. Technol.*, vol. 3, pp. 225–231, June 1995.
[13] M. S. Branicky, S. R. Phillips, and W. Zhang, "Stability of networked control systems: Explicit analysis of delay," in *Proc. Amer. Contr. Conf.*, vol. 4, 2000, pp. 2352–2357.
[14] G. C. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," in *Proc. Amer. Contr. Conf.*, vol. 4, pp. 2876–2880.
[15] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Contr. Syst. Mag.*, vol. 21, no. 1, pp. 84–99, Feb. 2001.
[16] J. K. Hale and S. M. Verduyn Lunel, *Introduction to Functional Differential Equations*. New York: Springer-Verlag, 1993, pp. 151–161.
[17] S.-T. Levi and A. K. Agrawala, *Real Time System Design*. New York: McGraw-Hill, 1990.
[18] N. C. Audsley, A. Burns, and A. J. Wellings, "Deadline monotone scheduling theory and application," *Proc. IFAC J. Contr. Eng. Practice*, vol. 1, no. 1, pp. 71–78, 1993.

Hong Seong Park was born in Korea in 1961. He received the B.S., M.S., and Ph.D. degrees from Seoul National University, Seoul, Korea, in 1983, 1986, and 1992, respectively.

Since 1992, he has been with the Department of Electrical and Computer Engineering, Kangwon National University, Korea. His research interests include design and analysis of communication networks and mobile/wireless communication, discrete-event systems, and network-based control systems.

Yong Ho Kim received the B.S., M.S., and Ph.D. degree in control engineering from Seoul National University, Seoul, Korea, in 1990, 1992, and 1997, respectively.

From 1997 to 2000, he was a Senior Engineer in the Daewoo Electronics company, Seoul, Korea, where he worked in the digital TV group. Since 2000, he has been with the Teleman Multimedia, Inc., USA and Korea, where he is currently a Senior Manager at the MPEG-4 video codec chip design group. His research interests are multivariable robust and predictive control, computer-aided control system design, factory network application, fuzzy control, digital multimedia system, and MPEG-4 video codec chips.

Dong-Sung Kim was born in Korea on January 25, 1969. He received the B.S. and M.S. degrees in electronic engineering from Hanyang University, Seoul, Korea, in 1992 and 1994, respectively.

Since 1998, he pursues the Ph.D. degree in EECS, Seoul National University, Seoul, Korea. From 1994 to 1998, he worked as a full-time Researcher in ERC-ACI at Seoul National University. From September 2000 to December 2001, he worked as a part-time Lecturer in department of information and communication at Dong-Guk University. His main research interests include home automation system, networked control system, and industrial networks.

Wook Hyun Kwon was born in Korea on January 19, 1943. He received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 1966 and 1972, respectively. He received the Ph.D. degree from Brown University, Providence, RI, in 1975.

From 1976 to 1977, he was an Adjunct Assistant Professor at University of Iowa, Iowa City. Since 1977, he has been with the School of Electrical Engineering, Seoul National University. From 1981 to 1982, he was a Visiting Assistant Professor at Stanford University, Stanford, CA. Since 1991, he has been the Director of the Engineering Research Center for Advanced Control and Instrumentation. His main research interests are currently multivariable robust and predictive controls, statistical signal processing, discrete-event systems, and industrial networks.