PERGAMON

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Control Engineering Practice 11 (2003) 1301–1313

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

Maximum allowable delay bounds of networked control systems

Dong-Sung Kim[a], Young Sam Lee[a], Wook Hyun Kwon[a,*], Hong Seong Park[b]

[a] *School of Electrical and Computer Engineering, Control Information Systems Laboratory, Seoul National University, San 56-1, Shilrim-dong, Kwanak-ku, Seoul 151-742, South Korea*

[b] *Department of Electrical and Computer Engineering, Kangwon National University, 192-1, Hyoza 2-Dong, Chuncheon, Kangwon-Do 200-701, South Korea*

Received 16 July 2001; accepted 22 October 2002

Abstract

This paper proposes a new method to obtain a maximum allowable delay bound for a scheduling of networked control systems. The proposed method is formulated in terms of linear matrix inequalities and can give a much less conservative delay bound than the existing methods. A network scheduling method is presented based on the delay obtained through the proposed method, the bandwidth of a network is allocated to each node and the sampling period of each sensor and controller is determined. The presented method can handle three types of data (periodic data, sporadic data, and message) and guarantees real-time transmission of periodic and sporadic data, and minimum network utilization for non-real time message.
© 2002 Elsevier Ltd. All rights reserved.

Keywords: Networked control systems; Maximum allowable delay bound; Linear matrix inequalities; Scheduling method

1. Introduction

In distributed control systems, a feedback control loop is closed through a network. Distributed control systems with networks are called networked control systems (NCSs). In an NCS, various delays with variable length occur due to sharing a common network medium, which are called network-induced delays. Network-induced delays can vary widely according to the transmission time of messages and the overhead time. The network in the NCS should handle three types of data: periodic data, sporadic data, message. The transmission time through the media is largely dependent on the network protocols, especially data link layer protocols of networks and data length. Hence, it is necessary to present some methods to make these network-induced delays bounded and smaller, which are called network scheduling methods for the NCS.

In feedback control systems, it is important that sampled data should be transmitted within a sampling period and that stability of control systems should be guaranteed. While a shorter sampling period is preferable in most control systems, for some purposes 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). Therefore, it is necessary to find the MADB for stability of the NCS, and then to find an appropriate network scheduling method that limits the network-induced delay to less than the MADB. A network scheduling method is required to reduce network-induced delays within the MADBs, while guaranteeing real-time transmission of sporadic and periodic data, and to minimize network utilization for non-real time message.

An MADB has been obtained from stability conditions of control systems. There have been some results on the stability of NCSs (Asok & Yoram, 1988; Krtolica et al., 1994; Feng-Li Lian, Moyne, & Tilbury, 2002), but these were concerned with obtaining stability conditions of the NCS with a given delay. There have been also some results on the MADB for stability in non-networked control systems (Mori, Fukuma, & Kuwahara, 1981; Su & Huang, 1992). In these papers, the MADB is obtained using the Ricatti equation approach, which yields conservative delay bounds. Less conservative results on the MADB in non-networked control

*Corresponding author. Tel.: +8228806485 Ext. 409; fax: +8228788933.

E-mail address: dskim@cisl.snu.ac.kr (D.-S. Kim).

0967-0661/03/$ - see front matter © 2002 Elsevier Ltd. All rights reserved.
PII: S0967-0661(02)00238-1

Nomenclature

$L^j_{S_i}$	the largest required time to start transmitting sensor data in the basic sampling period
N	the total number of nodes in the NCS
N_E	the number of basic sampling periods in the entire sampling period
N^T_p	the total number of transmissions of periodic data during the interval $N_E T^1$
N_p	the number of sensor and actuator data packets for periodic data in U_L, U^*_S, and U^*_L
N^M_P	the maximum integer of N_P
N_α	the total number of α nodes in the NCS (Hereinafter, α can be C (controller), A (actuator), or S (sensor))
N^i	the total number of nodes in the ith loop
N^1_α	the number of α nodes in the ith loop
N^M_S	the maximum number of sporadic data which is arrived in a basic sampling period. The basic sampling period means the minimum sampling period in all loops
N^b_A	the number of actuators in U_L
N^b_S	the number of sensors in both U_L and U^*_S
P	the total number of loops that use the same medium
T^j	a sampling period of the jth loop
$T^j_{\alpha_i}$	the data transmission time of periodic data in the ith α node in the jth loop
T_β	an interval for transmission of β data or messages. (Hereinafter, β can be P (Periodic Data), S (Sporadic Data), or N (Messages))
T^x_O	Overhead time of transmission
T_O	maximum overhead time in one node
T^x_O	maximum overhead time related to nodes which do not take part in transmission during each basic sampling period
T_{O_β}	the maximum overhead time to transfer β data or a message packet
T^j_D	the MADB in the jth loop
T^i_B	the ith basic sampling period in the largest sampling period
U_L	a set of loops whose all nodes are included in the considered basic sampling period
U^*_L	a set of loops whose all nodes are not included in the considered basic sampling period
U_N	Utilization of messages in a basic sampling period
U^m_N	the minimum utilization for the messages
U^*_S	a set of sensors which are in U^*_L

systems are reported in Li and de Souza (1997a, b) and Park (1999). However, these results still remains to be improved (Moon, Park, Kwon, & Lee, 2001). The MADB thus obtained can be extended as a maximum bound of a sampling period in the NCS. That is, the sampling periods determined by the proposed sampling period decision algorithm can be set to values less than the MADB.

A scheduling method was presented in the NCS with fieldbus networks (Cavalieri, Stefano, & Mirabella 1995; Beauvais & Deplanche, 1995). But those papers did not consider the MADB and the controller delay time, which were important in control applications. There have been some studies on scheduling algorithms that can be applied to the NCS (Beauvais & Deplanche, 1995; Hong, 1995). A dynamic scheduling algorithm modified from the rate monotone scheduling algorithms was presented for periodic and asynchronous data in fieldbus networks. A heuristic algorithm was presented for periodic tasks only (Beauvais & Deplanche, 1995), but it did not support asynchronous data. The several algorithms for dynamically scheduling of networked control systems were proposed (Zuberi & Shin, 1997; Hong Ye, Walsh, & Bushnell, 2001). It had limitations when applied to the NCS because it did not consider some characteristics of the NCS, such as the MADB and sampling periods. A scheduling algorithm that can allocate the bandwidth of a network and determine sensor data sampling periods was presented (Hong, 1995). In Hong (1995), the NCS had only single input and single output (SISO), only periodic data were considered, and the MADB was not obtained analytically.

A network scheduling method considering three types of data based on a multi-input and multi-output (MIMO) system was proposed (Park, Kim, Kim, & Kwon, 2002). In this paper, the estimation of MADB using the Ricatti equation is too conservative, which means the estimated MADB is too small and the network scheduling method discussed in this paper is somewhat heuristic. In Branicky, Phillips, and Zhang (2000), Wei Zhang, Branicky, and Phillips (2001) and

Walsh and Hong Ye (2001), calculation methods of MADBs and stability analysis of NCSs were presented. However, these results were conservative to be of practical use and still remains to be improved. Further research is needed with regard to an estimation of a less conservative MADB for stability of the NCS and systematic scheduling methods for three types of data.

In this paper, a new method to obtain the MADB guaranteeing a stability of the NCS is proposed in terms of linear matrix inequalities (LMI). The proposed method gives a much less conservative delay bound than the existing methods. The paper includes a network scheduling method considering three types of data based on a MIMO system. The network scheduling method is based on the results in Park et al. (2002). It allocates the bandwidth of a network to a node, determines the sensor data sampling periods of each loop using the obtained MADB, guarantees real-time transmission of sporadic data and periodic data within the sampling periods, and minimizes the network utilization for message.

This paper is organized as follows. In the following section, an NCS model is described. In Section 3, an MADB for the stability of the NCS is derived by LMI formulation. In Section 4, a network scheduling method that allocates the bandwidth and determines the sampling period for the NCS is presented. In Section 5, the simulation results are given to show that the method is useful. Finally, the conclusions are presented in Section 6.

2. MADB for stability in control loop

NCSs can be described as Fig. 1. A control loop is composed of a controller, sensors, and actuators. The sensors, the actuators, and a controller share a common communication medium.

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 outputs of the plant could deviate from the desired trajectory, or the controlled system. Hence, it is necessary to derive the MADB from parameters and configurations of the given plant and the controller. Fig. 2 shows a feedback control system with network induced delays.

The timing such an NCS is illustrated in Fig. 3. A drawback with this setup is that system becomes time varying.

In this paper, a stability is checked by single control loop model with each sensor and actuator node. The node which have a multiple control loop can be changed to the sum of nodes have a single control loop. That is to say, the node which have a multiple control loop can be changed to the sum of nodes which have a single control loop.

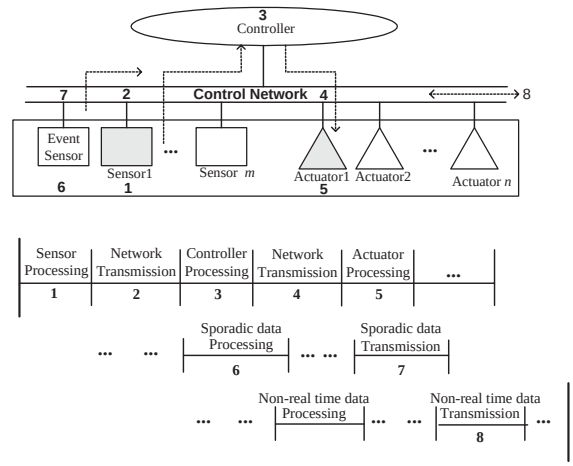

Fig. 1. Diagram of control loops using a network.

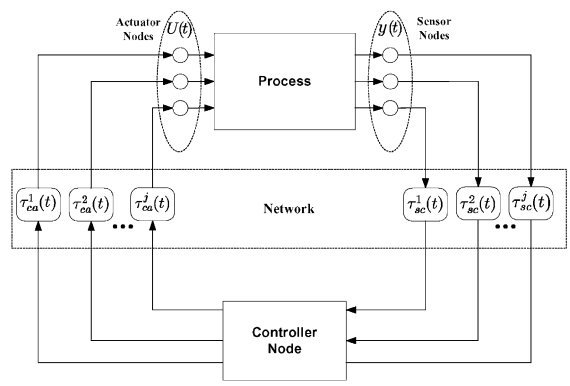

Fig. 2. Networked control loops with sensors and actuators (Nilson, 1998).

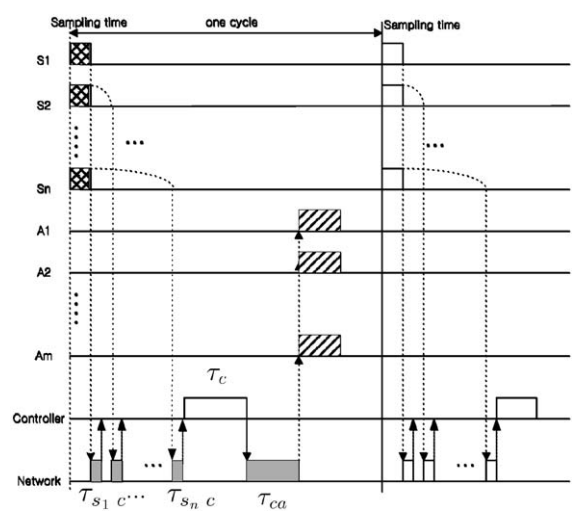

Fig. 3. Timing of signals in the NCS.

In Fig. 3, the timing diagram illustrates the process output and sampling instants, the signal into the controller node, the signal into the actuator node and the network-induced delay.

2.1. MADB of NCS in continuous-time system

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

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

where $u_p^j(t) \in R^{N_A^j}$, $y_p^j(t) \in R^{N_S^j}$, $x_p^j(t) \in R^{N_P^j}$. N_A^j, N_S^j, and N_P^j is the dimension of the sensor, actuator, and 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

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

where $u_c^j(t) \in R^{N_S^j}$, $y_c^j(t) \in R^{N_A^j}$, $x_c^j(t) \in R^{N_P^j}$. $\tilde{\tau}_c^j$ is computation time in the controller j, which satisfies $0 \leqslant \tilde{\tau}_c^j \leqslant \tau_{c,max}^j$, where $\tau_{c,max}^j$ is the maximum computation time in the controller j. For conveniences, 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

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

where $0 \leqslant \tilde{\tau}_{sc}^j \leqslant \tau_{sc,max}^j$, $0 \leqslant \tilde{\tau}_{ca}^j \leqslant \tau_{ca,max}^j$, $\tilde{\tau}_{sc}^j$ 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. In this paper, the lower and upper bounds of $\tau_{sc}(t)$ and $\tau_{ca}(t)$ are only used as constraints.

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

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

where $x^j(t) = [x_p^{j\mathrm{T}}(t) \;\; x_c^{j\mathrm{T}}(t)]^{\mathrm{T}}$.

Then the above equation can be rewritten as

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

where

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

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

$$0 \leqslant \tilde{\tau}_1^j = \tilde{\tau}_{sc}^j \leqslant \tau_{sc,max}^j = \tilde{\tau}_{1,max}^j,$$

$$0 \leqslant \tilde{\tau}_2^j = \tilde{\tau}_{sc}^j + \tilde{\tau}_{ca}^j + \tilde{\tau}_c^j \leqslant \tau_{sc,max}^j + \tau_{ca,max}^j + \tau_{c,max}^j = \tilde{\tau}_{2,max}^j$$

and

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

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

To generalize results to the multiple state-delayed case, consider the following system:

$$\dot{x}(t) = Fx(t) + \sum_{i=1}^{N} F_i x(t - \tau_i),$$
$$x(t) = \phi(t), \quad t \in [-\bar{\tau}, 0], \tag{6}$$

where $x(t) \in \mathbf{R}^n$ is the state, $\tau_i > 0$ is the delay of the system, $\phi(\cdot)$ is the initial condition, F, F_i are real constant matrices with appropriate dimensions, and $\bar{\tau}$ is upper bound of τ_i.

Our aim is to develop a new method to obtain the MADB guaranteeing stability of the NCS. In obtaining the results of this paper, the following upper bound for the inner product of two vectors plays an important role:

$$-2a^{\mathrm{T}}b \leqslant \inf_{X,Y,Z} \begin{bmatrix} a \\ b \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} X & Y - I \\ Y^{\mathrm{T}} - I & Z \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}, \tag{7}$$

where $\begin{bmatrix} X & Y \\ Y^{\mathrm{T}} & Z \end{bmatrix} \geqslant 0$ and I denotes an identity matrix with an appropriate dimension. Extending the idea of Eq. (7), Lemma 1 is derived.

Lemma 1 (Moon et al., 2001). *Assume that $a(\cdot) \in \mathbf{R}^{n_a}$, $b(\cdot) \in \mathbf{R}^{n_b}$, and $\mathcal{N}(\cdot) \in \mathbf{R}^{n_a \times n_b}$ are defined on the interval Ω. Then, for any matrices $X \in \mathbf{R}^{n_a \times n_a}$, $Y \in \mathbf{R}^{n_a \times n_b}$, and $Z \in \mathbf{R}^{n_b \times n_b}$, the following holds:*

$$-2\int_{\Omega} a^{\mathrm{T}}(\alpha)\mathcal{N}b(\alpha)\,\mathrm{d}\alpha \leqslant \int_{\Omega} \begin{bmatrix} a(\alpha) \\ b(\alpha) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} X & Y - \mathcal{N} \\ Y^{\mathrm{T}} - \mathcal{N}^{\mathrm{T}} & Z \end{bmatrix} \begin{bmatrix} a(\alpha) \\ b(\alpha) \end{bmatrix} \mathrm{d}\alpha,$$

where

$$\begin{bmatrix} X & Y \\ Y^{\mathrm{T}} & Z \end{bmatrix} \geqslant 0. \tag{8}$$

Theorem 1. *If there exist* $P>0$, $Q_i>0$, X_i, Y_i *and* Z_i, $i=1,\ldots,N$, *such that*

$$\begin{bmatrix} \mathscr{P}_{11} & \mathscr{F}^{\mathrm{T}}\mathscr{Z} \\ \mathscr{Z}\mathscr{A} & -\Gamma \end{bmatrix} < 0, \quad \begin{bmatrix} X_i & Y_i \\ Y_i^{\mathrm{T}} & Z_i \end{bmatrix} \geqslant 0, \tag{9}$$

where

$$\mathscr{P}_{11} \triangleq \begin{bmatrix} \mathscr{T}_{11} & P\mathscr{F}_1 - \mathscr{Y} \\ \mathscr{F}_1^{\mathrm{T}}P - \mathscr{Y} & -\mathscr{Q} \end{bmatrix},$$

$$\mathscr{F} \triangleq [F \quad F_1 \quad \cdots \quad F_N],$$
$$\mathscr{F}_1 \triangleq [F_1 \quad \cdots \quad F_N],$$
$$\mathscr{Y} \triangleq [Y_1 \quad \cdots \quad Y_N],$$
$$\mathscr{Z} \triangleq \bar{\tau}[Z_1 \quad \cdots \quad Z_N],$$
$$\mathscr{Q} \triangleq \mathrm{diag}\{Q_1, \ldots, Q_N\},$$
$$\Gamma \triangleq \bar{\tau}\,\mathrm{diag}\{Z_1, \ldots, Z_N\},$$
$$\mathscr{T}_{11} \triangleq F^{\mathrm{T}}P + PF + \sum_{i=1}^{N} \{Y_i + Y_i^{\mathrm{T}} + \bar{\tau}X_i + Q_i\},$$

then system (6) *is asymptotically stable for any time-delay* τ_i *satisfying* $0 \leqslant \tau_i \leqslant \bar{\tau}_i$, $i=1,\ldots,N$.

This theorem extends the results in Moon et al. (2001), which deals with an MADB for a single state-delayed systems. The MADB can be obtained efficiently using the MATLAB LMI Toolbox. System (5) can be represented by Eq. (6) with $N=3$ and $\bar{\tau}$ can be interpreted as $\max\{\bar{\tau}_1, \bar{\tau}_2, \bar{\tau}_3\}$.

The delay bound of each control loop is used as a parameter in the determination of the sampling period and allocation of bandwidth.

2.2. MADB of NCS in discrete-time system

As a simple example of extension to the discrete-time case of Theorem 1, the system (10) is considered. In a discrete-time system, the one-time-step delay is considered:

$$x(k+1) = Fx(k) + F_1x(k-\tau),$$

$$x(k) = \phi(k), \quad k \in [-\bar{\tau}, 0], \tag{10}$$

where $\tau > 1$ is the delay of the system, $\phi(\cdot)$ is the initial condition, F, F_1 are real constant matrices with appropriate dimensions, and $\bar{\tau}$ is upper bound of τ. A discrete-time equivalent of Theorem 1 is represented by

Theorem 2. *If there exist* $P>0$, $Q>0$, X, Y *and* Z, *such that*

$$\begin{bmatrix} \mathscr{S}_{11} & -Y & F^{\mathrm{T}}P & \bar{\tau}(F-I)^{\mathrm{T}}Z \\ -Y^{\mathrm{T}} & -Q & F_1^{\mathrm{T}}P & \bar{\tau}F_1^{\mathrm{T}}Z \\ PF & PF_1 & -P & 0 \\ \bar{\tau}Z(F-I) & \bar{\tau}ZF_1 & 0 & -\bar{\tau}Z \end{bmatrix} < 0,$$

$$\mathscr{S}_{11} \triangleq -P + \bar{\tau}X + Y + Y^{\mathrm{T}} + Q, \tag{11}$$

$$\begin{bmatrix} X & Y \\ Y^{\mathrm{T}} & Z \end{bmatrix} \geqslant 0 \tag{12}$$

then system (10) *is asymptotically stable for any time-delay* τ *satisfying* $0 \leqslant \tau \leqslant \bar{\tau}$.

Theorem 2 can be proved via the similar procedure in Appendix A. For the proof of Theorem 2, the conditions of $\Delta V(k) = V(k+1) - V(k) < 0$, $\Delta V(k) = \Delta V_1(k) + \Delta V_2(k) + \Delta V_3(k) < 0$ can be obtained by Lemma 1.

An MADB in a discrete-time system is a maximum sampling time which is obtained from Theorem 2. In general, a faster sampling rate is said to be desirable in sampled-data systems so the discrete-time control design and performance can approximate that of the continuous-time system. But in NCSs, a faster 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 NCS design.

3. Scheduling algorithm in multiple control loops

This section describes a network scheduling method based on the MADB obtained through the proposed method, the bandwidth of a network is allocated to each node and the sampling period of each sensor and controller is determined. 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. 4. In addition to the three periods,

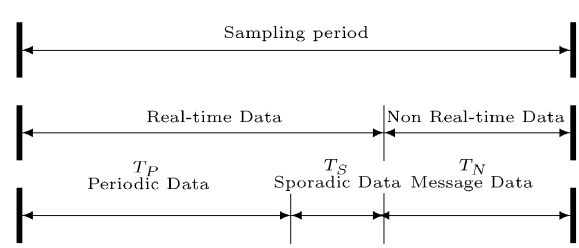

Fig. 4. Configuration of phases in a basic sampling period.

there can be a synchronization period. The synchronization period is included in T_O^x.

The following assumptions are used in this paper:

- In networks, communications are error-free. That is, there are no failures in transferring messages.
- Sampling time of sensors in a loop is synchronized at starting instant of basic sampling periods.
- 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, $2\times T^1$, $4\times T^1$, $8\times T^1$, ...) and should not exceed the MADB in the corresponding loop.
- Controller computational delay can be absorbed into either τ_{ca} or τ_{sc} (Walsh & Hong Ye, 2001).

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 sampling periods of all loops can be used as the largest period.

The sixth assumption was used for absorbing controller delay time to node data transmission time without loss of generality.

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

$$U_N = \frac{T_N}{T^1}. \tag{13}$$

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

$$U_N^m \leqslant U_N \tag{14}$$

should be satisfied. Using Eq. (13), the above equation is converted to

$$U_N^m T^1 \leqslant T_N. \tag{15}$$

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

$$N_S^M \hat{T}_S^M \leqslant T_S \tag{16}$$

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. This means that $N_S^M \hat{T}_S^M$ 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

$$T^1 = L_{S_i}^j + T_P + T_S + T_N + T_O^x + T_{CD}, \tag{17}$$

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^*. $L_{S_i}^j$ is the largest required time to start transmitting sensor data, which can be shortened in network protocols using adequate scheduling method in the basic sampling period. 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

$$N_P = N_S^b + N_A^b, \tag{18}$$

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

$$T_P = \sum_{j\in U_L}\left(\sum_{i=1}^{N_S^j}(T_{S_i}^j + T_{O_P}) + \sum_{i=1}^{N_A^j}(T_{C_i}^j + T_{O_P})\right) + \sum_{j\in U_L^*, i\in U_S^*}(T_{S_i}^j + T_{O_P}), \tag{19}$$

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

$$L_{S_i}^j + T_P + N_S^M \hat{T}_S^M + U_N^m T^1 + T_O^x \leqslant T^1. \tag{20}$$

The above equation can be changed to

$$T_P \leqslant \langle (1 - U_N^m)T^1 - N_S^M \hat{T}_S^M - T_O^x - L_{S_i}^j \rangle, \tag{21}$$

where

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

Note that the NCS 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 data transmission times of sensors are equal as M_S and data transmission times of controllers are equal as M_C, then using Eq. (19), the above equation becomes

$$N_S^b(M_S + T_{O_P}) + N_A^b(M_C + T_{O_P}) \leqslant \langle (1 - U_N^m)T^1 - N_S^M \hat{T}_S^M - T_O^x - L_{S_i}^j \rangle. \tag{22}$$

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

Eq. (22) becomes

$$N_P \leqslant \left\lfloor \frac{\langle (1-U_N^m)T^1 - N_S^M \hat{T}_S^M - T_O^x - L_{S_i}^j \rangle}{M + T_{O_P}} \right\rfloor, \tag{23}$$

where $\lfloor Z \rfloor$ is the largest integer smaller than or equal to the value Z. Let the right part of Eq. (23) be N_P^M.

Let the largest sampling period in the NCS be $N_E T^1$ 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 $N_E T^1$ be N_P^{T}, and then it can be calculated as

$$N_P^{\mathrm{T}} = \sum_{j=1}^{P} Q(N^j) N^j, \tag{24}$$

where $Q(N^j) = N_E T^1 / T^j$ 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 $N_P^M N_E$. The largest sampling period in the NCS depends on the largest MADB (T_D^P). The sampling period decision algorithm based on the bisection method can decide the basic sampling period.

(1) Set the MADB of each control loop using Theorems 1 or 2.
(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 Eq. (24) and the results of the above step.
(4) Let $T^1 = T_D^1$, $T_L = 0$, and $T_U = T^1$, $k = 0$.
(5) Choose T^j such as $T^j \leqslant T_D^j$ and $T^j = max(2^k T^1)$ for $k = 0, 1, 2, \ldots$.
(6) Compute N_P^M using Eq. (23).
(7) If $\lceil N_P^{\mathrm{T}} / N_E \rceil$ is equal to N_P^M or $(N_P^M - \lceil N_P^{\mathrm{T}} / N_E \rceil)$ is within in the given bound, then $T_B^j = T^j$ for $j = 1, \ldots, P$ and go to the next step,
else if $\lceil N_P^{\mathrm{T}} / N_E \rceil$ 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 $\lceil N_P^{\mathrm{T}} / N_E \rceil$ is greater than N_P^M and $k = 0$, then terminate the algorithm (the scheduling is failed),
else if $\lceil N_P^{\mathrm{T}} / N_E \rceil$ is greater than N_P^M and $k \neq 0$, then $T_U = T^1$, take the basic sampling period (T^1) as $(T_L + T_U)/2$, $k++$, and go to step 5 ($\lceil Z \rceil$ is the smallest integer larger than or equal to the value Z).
(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 following bandwidth scheduling algorithm.

In Figs. 5 and 6, the flow chart of determination sampling period algorithm is described.

Using the bandwidth-scheduling algorithm, data packets can be allocated as follows. First, sensor data packets in loop 1 are transmitted to the corresponding controller through the network medium. When all sensor data packets in loop 1 are transmitted, computations of control values in the controller of loop 1 are started. This is the controller delay time in loop 1. During this controller delay, sensor data packets in next loop (loop 2) are transmitted using the network medium. So the controller delay is overlapped with transmission time of sensor data in loop 2. After the controller delay time, a controller of loop 1 transmits its data to actuators. After 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 time for messages is left after 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 is less than or equal to the MADB, is selected as a minimum sampling period of loop 1 according to the sampling period decision algorithm.

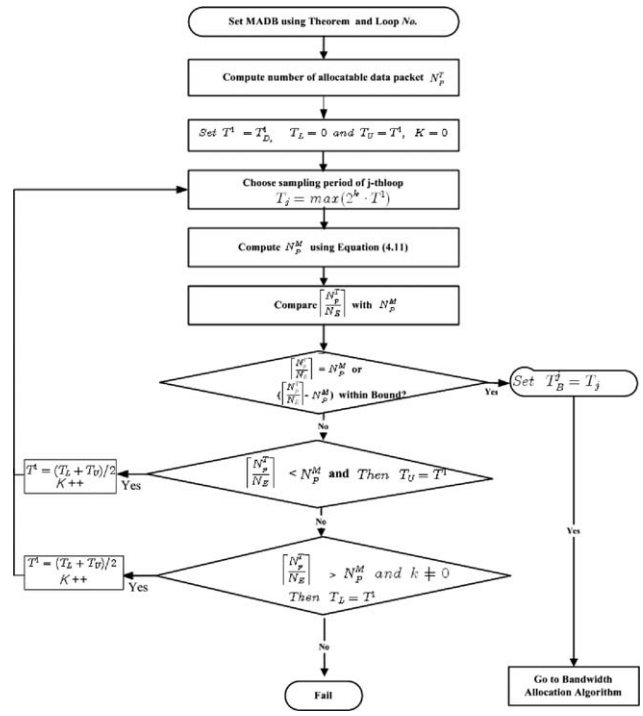

Fig. 5. Flow chart of scheduling algorithm.

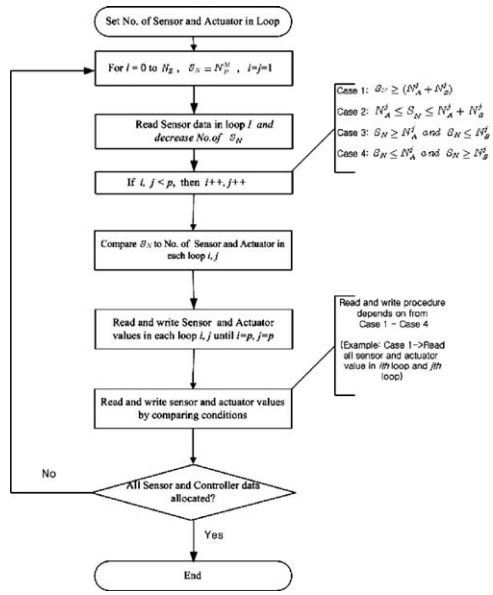

Fig. 6. Flow Chart of bandwidth allocation algorithm.

other high-speed network protocols should be selected or the number of nodes should be reduced. These two algorithms are presented as the scheduling method in this paper.

Set $\bar{N}_S^h = N_S^h$, $\bar{N}_A^h = N_A^h$ for $1 \leqslant h \leqslant P$, and
$\bar{N}_S^h = \bar{N}_A^h = 0$ for $h > P$,
for $l = 1$ to $l = N_E$ do,
  set $S_N = N_P^M$,
  set $i = j = 1$,
  read the sensor values in loop i,
  $S_N = S_N - \bar{N}_S^i$, $\bar{N}_S^i = 0$, and $i++$,
  repeat
    while ($\bar{N}_S^i == 0$ and $i \leqslant P$),
      $i++$,
    end of while,
    while ($\bar{N}_A^j == 0$ and $j \leqslant P$),
      $i++$,
    end of while,
    if $S_N \geqslant (\bar{N}_A^j + \bar{N}_S^i)$,
      read all sensor values in the ith loop,
      write all actuator values in the j-th loop,
      $S_N = S_N - \bar{N}_S^i - \bar{N}_A^j$, $\bar{N}_S^i = \bar{N}_A^j = 0$, $i++$, $j++$,
    else if $\bar{N}_A^j < S_N < (\bar{N}_A^j + \bar{N}_S^i)$,
      if $j < i$,
        $S_N = S_N - \bar{N}_A^j$,
        read S_N sensor values in the ith loop,
        write all actuator values in the jth loop,
        $\bar{N}_S^i = \bar{N}_S^i - S_N$, $S_N = \bar{N}_A^j = 0$, $j++$,
      else,
        read $min(S_N, \bar{N}_S^i)$ sensor values in the ith loop,
        $\bar{N}_S^i = \langle \bar{N}_S^i - S_N \rangle$, $S_N = \langle S_N - \bar{N}_S^i \rangle$,
        if $\bar{N}_S^i = 0$, then $i++$,
      endif,
    else if $S_N \leqslant \bar{N}_A^j$ and $S_N \geqslant \bar{N}_S^i$,
      read all sensor values in the ith loop,
      $S_N = S_N - \bar{N}_S^i$, $\bar{N}_S^i = 0$, $i++$,
      write S_N actuator values in the jth loop,
      $\bar{N}_A^j = \bar{N}_A^j - S_N$, $S_N = 0$,
    else if $S_N \leqslant \bar{N}_A^j$ and $S_N < \bar{N}_S^i$,
      if $(i - j) < 2$,
        read S_N sensor values in the ith loop,
        $\bar{N}_S^i = \bar{N}_S^i - S_N$, $S_N = 0$,
      else,
        write S_N actuator values in the jth loop,
        $\bar{N}_A^j = \bar{N}_A^j - S_N$, $S_N = 0$,
      endif,
    endif,
  until ($S_N = 0$ or $j > P$),
  $m = 1$,
  while $((\lceil \frac{l \cdot T^1}{T^m} \rceil - \lfloor \frac{l \cdot T^1}{T^m} \rfloor) == 0)$ and $(m \leqslant P)$),
    $\bar{N}_S^m = N_S^m$, $\bar{N}_A^m = N_A^m$, $m++$,
  end of while,
end of for loop.

Let us consider some points in applications of the scheduling method of the NCS in the case of token control networks. The worst overhead should be reserved for the case when the token has been passed over to the next station just before a transmission request is made. In token control, overhead time occupies a large part of the whole period and synchronization are difficult to estimate. 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 well adjusted, a node may have to wait while the token is passed over all the other nodes. Because an 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 NCS.

Now consider the application 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 synchronizations. 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 similar to the case of token control network.

4. Simulation results

4.1. Analysis of an MADB in a control loop

As an analysis of the MADB, the following state space of the plant (Branicky et al., 2000; Wei Zhang et al., 2001) is considered.

$$\dot{x}(t) = A_c x(t) + B_c u(t),$$
$$y(t) = C_c x(t), \tag{25}$$

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 obtained as 53.8 ms from Lemma 1 (Park et al., 2002) and 26.3×10^3 ms from Theorem 1.

The MADBs were obtained from methods in Branicky et al. (2000) and Walsh et al. (1999) are 0.27 and 0.45 ms, respectively. In Wei Zhang et al. (2001), the MADB was obtained as 1.7×10^3 ms by stability region technique. These results are summarized in Table 1.

4.2. Examples of NCS model

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 a part of a six-axis robot. If the armature inductance (L_a) and viscous frictional coefficient (B_m) are negligible, the motor dynamics can be modelled by

$$\dot{x}_p = F_p x_p + G_p u_p$$
$$= \begin{bmatrix} -K_i K_b / R_a J & 0 \\ 1 & 0 \end{bmatrix} x_p + \begin{bmatrix} K_i / R_a J \\ 0 \end{bmatrix} u_p, \tag{26}$$

$$y_p = x_p, \tag{27}$$

where $x_p = [\omega \;\; \theta]^{\mathrm{T}}$, u_p is 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, respectively, the armature resistance, torque constant, back emf constant, and inertia of rotor and load. If a constant gain (K) is used as a state feedback controller, system (5) is changed to

$$\dot{x}_p(t) = F_p x_p(t) + G_p * K x_p(t - \tau) \tag{28}$$

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

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

Using Lemma 1 (Park et al., 2002) 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 2. Using Lemma 3 (Park et al., 2002) and the given parameters of the motors, the MADBs are calculated as 0.16, 0.21, 0.22, 0.30, 0.33 and 0.39 ms. Using Theorem 1, the MADBs are calculated as 1.812×10^3, 1.802×10^3, 1.799×10^3, 1.785×10^3, 1.780×10^3, and 1.771×10^3 ms. Therefore the final MADB is obtained as 1.771×10^3 ms by Theorem 1.

4.3. Application of a scheduling method using MADB

For the test of a scheduling method using MADBs, three motors have the value of $R_a = 14$, 19, and 21 Ω, respectively. Using Lemma 1 from Park et al. (2002) and the given parameters of the motors, the MADBs are calculated as 3.1, 6.1, and 7.5 ms. Using Theorem 1, the MADBs are calculated as 1.799×10^3, 1.785×10^3 and 1.780×10^3 ms, respectively.

From now, 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 4 bytes.

Using the notation in the nomenclature, 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 $j = 1, 2, 3$,
$N^* = 1$, $P = 3$,
$N = \sum_{j=1}^{P} (N_C^j + N_S^j) + N^* = 10$,
$L_{S_i}^j = 0.1$ ms for $i = 1, 2$, $j = 1, 2, 3$,

Table 1
Simulation results of MADB (ms) by model (4.1)

Theorem 1	In Wei Zhang et al. (2001)	In Park et al. (2002)	In Branicky et al. (2000)
26.3×10^3	1.7×10^3	53.8 (Lemma 1) 17.5 (Lemma 3)	0.45

Table 2
Simulation results of MADB (ms) in the each loop

Loop	R_a (Ω)	By Lemma 1 (Park et al., 2002)	By Lemma 3 (Park et al., 2002)	By Theorem 1 (or 2)
Loop 1	10	1.4	0.16	1.812×10^3
Loop 2	13	2.6	0.21	1.802×10^3
Loop 3	14	3.1	0.22	1.799×10^3
Loop 4	19	6.1	0.30	1.785×10^3
Loop 5	21	7.5	0.33	1.780×10^3
Loop 6	25	10	0.39	1.771×10^3

$T_D^1 = 3$ 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 NCS. $L_{S_i}^j$ is the ith sensor delay in the jth loop.

Because actuator nodes are assumed not to send any data in normal operations, the transmission of all actuator nodes are not considered. The transmission speeds are different 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 PROFIBUS. If the universal asynchronous receiver and transmitter (UART) character which consists of 11 bits/byte is used in the token control, then parameters can be given as follows:

$T_{S_i}^j = T_{C_1}^j = M = 4$ bytes $\times$ 11 bits/byte $\times$ 1 μs/bit $=$ 44 μs, for $i = 1, 2,\ j = 1, 2, 3$.

$T_S^M = 2$ bytes $\times$ 11 bits/byte $\times$ 1 μs/bit $= 22$ μs.

Message overhead for periodic and sporadic data: $T_{O_P}^M = T_{O_S}^M = 9$ bytes $\times$ 11 bits/byte $\times$ 1 μs/bit $= 99$ μs.

Protocol overhead for periodic data can be bounded by one token rotation time and given by $T_{O_P}^P = 10(N) \times 3$ bytes (token) $\times$ 11 bits/byte $\times$ 1 μs/bit $= 330$ μs.

Protocol overhead for sporadic data is calculated as $T_{O_S}^P = 10(N) \times 3$ bytes (token) $\times$ 11 bits/byte $\times$ 1 μs/bit $= 330$ μs,

$T_S = 2(N_S^M) \times (T_S^M + T_{O_S}^M + T_{O_S}^P) = 902$ μs,

$T_O^x = 330$ μs, $T_N = 0.16 \times 3$ ms $= 0.48$ ms.

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

$T_{S_i}^j = T_{C_1}^j = M = 4$ bytes $\times$ 8 bits/byte $\times$ 1 μs/bit $=$ 32 μs, for $i = 1, 2,\ j = 1, 2, 3$,

$T_{O_P}^M = 45$ bits $(RP_DAT) \times 1$ μs/bit $= 45$ μs,

$T_{O_P}^P = 61$ bits $(ID_DAT) \times 1$ μs/bit $= 60$ μs,

$T_S^M = 2$ bytes $\times$ 8 bits/byte $\times$ 1 μs/bit $= 16$ μs,

$T_{O_S}^M = 45$ bits $(RP_DAT) \times 1$ μs/bit $= 45$ μs,

$T_{O_S}^P = \{61(ID_RQ)+(45+16)(RP_RQ)+61(ID_DAT)\}$ bits $\times$ 1 μs/bit $= 183$ μs,

$T_S = 2(N_S^M) \times (T_S^M + T_{O_S}^M + T_{O_S}^P) = 488$ μs,

$T_N = 0.16 \times 3 = 0.48$ ms.

Applying steps 1–6 of the sampling period decision algorithm to the example, N_P^T is calculated as 12 and N_E is 2, while 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, 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 the case of the polling control network.

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

In the polling control case, as the overhead time in each node is bounded by a constant value, calculation results in the last section are similar to the real values.

An MADB can be determined from 1.77 to 1.81 s by Theorem 1 or 2. Hence, the basic sampling period can be reduced by about 1.4 ms in this case. Applying steps 1–6 of the sampling period decision algorithm to the example, N_P^T is calculated as 804 and N_E is 1214, while N_P^M is 8498 in the polling control network.

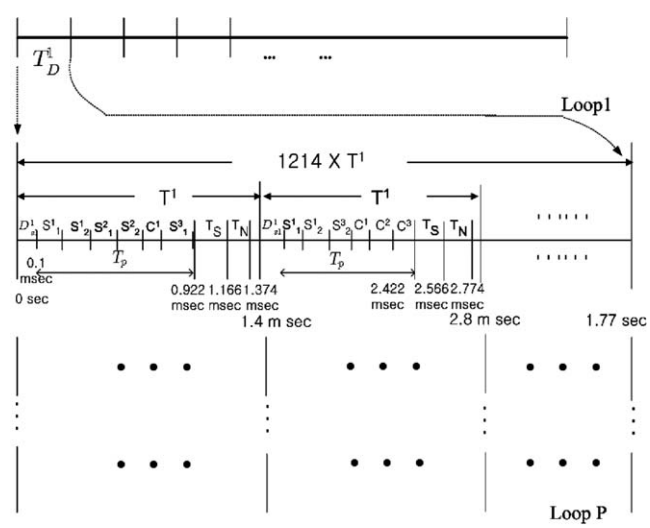

Fig. 7. Bandwidth allocation result using polling control. (S_i^j and C_i^j means sensor data and control data of the ith node in the jth loop respectively.)

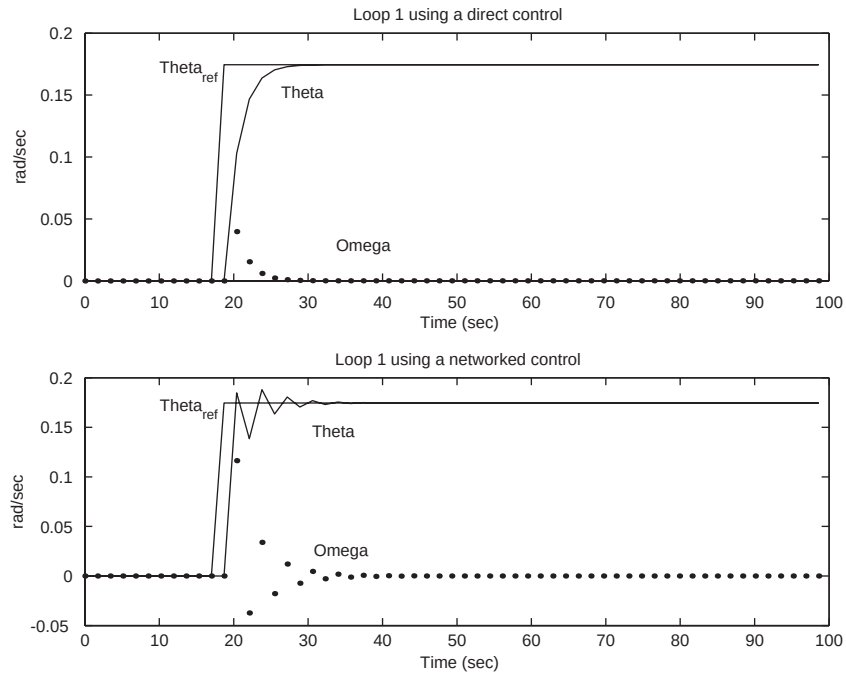

Fig. 8. Outputs of motor 1 position control ($R_a = 10$ (Ω), $\bar{\tau} = 1.812 \times 10^3$ ms).

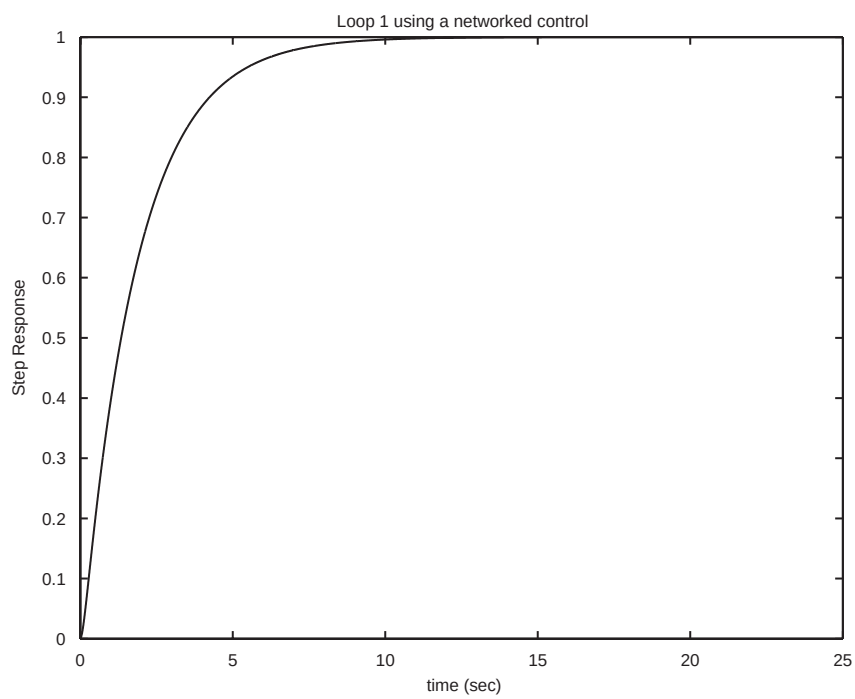

Fig. 9. Step response of networked control of loop 1.

Results from calculations are matched to those of the allocations. If 1.77 s MADB is selected, the bandwidth can be allocated as shown in Fig. 7. In Fig. 7, S_i^j and C_i^j means sensor data and control data of the ith node in the jth loop respectively.

From these results, it can be shown that the large-scaled NCSs with many nodes be scheduled using token control or polling control.

The simulation results in case of polling control case are shown in Fig. 8. In Fig. 8, 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 NCS is similar to that in the directly connected systems. The simulation results of the step response of networked control system in Fig. 9.

5. Conclusions

In this paper, the MADBs are obtained for the stability of the NCS using LMI formulation, and are used as the basic parameter for a scheduling method for the NCS. Further, the scheduling method for the NCS 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 to those, the presented method can guarantee real-time transmission of sporadic and periodic data, and minimum utilization for nonreal-time messages.

In an NCS, the presented method is useful, as it provides a solution to determine the sampling periods 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 NCS.

As the sampling periods used in the proposed method are multiples each other in the order 2, the simplified algorithm based on multiples of the smallest sampling period is necessary to be studied. As future works, the jitters will be investigated for the analysis of NCSs.

Acknowledgements

The authors acknowledge the financial supports from Brain Korea 21, Korea. They also thank the editor and anonymous reviewers for their time and efforts.

Appendix A

Proof of Theorem 1. Choose a Lyapunov functional as follows:

$$V(x(t-\alpha), \alpha \in [0, \bar{\tau}]) = V_1 + V_2 + V_3, \tag{29}$$

where

$$V_1 \triangleq x^{\mathrm{T}}(t)Px(t), \tag{30}$$

$$V_2 \triangleq \sum_{i=1}^{N} \left\{ \int_{-\tau_i}^{0} \int_{t+\beta}^{t} \dot{x}^{\mathrm{T}}(\alpha) Z_i \dot{x}(\alpha)\, \mathrm{d}\alpha\, \mathrm{d}\beta \right\}, \tag{31}$$

$$V_3 \triangleq \sum_{i=1}^{N} \left\{ \int_{t-\tau_i}^{t} x^{\mathrm{T}}(\alpha) Q x(\alpha)\, \mathrm{d}\alpha \right\}. \tag{32}$$

Since $x(t-\tau_i) = x(t) - \int_{t-\tau_i}^{t} \dot{x}(\sigma)\,\mathrm{d}\sigma = x(t) - \int_{t-\tau_i}^{t} [Fx(\sigma) + \sum_{i=1}^{N} \{F_i x(\sigma - \tau_i)\}]\,\mathrm{d}\sigma$, the system (6) can be written as (Hale & Lunel, 1993)

$$\dot{x}(t) = \left(F + \sum_{i=1}^{N} F_i \right) x(t) - \sum_{i=1}^{N} \left\{ F_i \int_{t-\tau_i}^{t} \dot{x}(\alpha)\,\mathrm{d}\alpha \right\}$$

and thus the derivative of V_1 satisfies the relation $\dot{V}_1 = 2x^{\mathrm{T}}(t)[P(F + \sum_{i=1}^{N} F_i)]x(t) - 2\sum_{i=1}^{N} \{x^{\mathrm{T}}(t)PF_i \int_{t-\tau_i}^{t} \dot{x}(\alpha)\,\mathrm{d}\alpha\}$. Defining $a(\cdot)$, $b(\cdot)$, and $\mathcal{N}$ in Eq. (8) as $a(\alpha) \triangleq x(t)$, $b(\alpha) \triangleq \dot{x}(\alpha)$, and $\mathcal{N} \triangleq PF_i$ for all $\alpha \in [t-\tau_i, t]$ and applying Lemma 1 will supply $\begin{bmatrix} X_i & Y_i \\ Y_i^{\mathrm{T}} & Z_i \end{bmatrix} \geqslant 0$, $i = 1, \ldots, N$, and

$$\begin{aligned}
\dot{V}_1 \leqslant{}& 2x^{\mathrm{T}}(t) \left[P \left(F + \sum_{i=1}^{N} F_i \right) \right] x(t) \\
&+ \sum_{i=1}^{N} \tau_i x^{\mathrm{T}}(t) X_i x(t) \\
&+ 2 \sum_{i=1}^{N} \left\{ x^{\mathrm{T}}(t)(Y_i - PF_i) \int_{t-\tau_i}^{t} \dot{x}(\alpha)\,\mathrm{d}\alpha \right\} \\
&+ \sum_{i=1}^{N} \left\{ \int_{t-\tau_i}^{t} \dot{x}^{\mathrm{T}}(\alpha) Z_i \dot{x}(\alpha)\,\mathrm{d}\alpha \right\} \\
\leqslant{}& x^{\mathrm{T}}(t) \\
&\times \left[F^{\mathrm{T}}P + PF + \sum_{i=1}^{N} \{\bar{\tau} X_i + Y_i + Y_i^{\mathrm{T}}\} \right] x(t) \\
&+ 2 \sum_{i=1}^{N} \{x^{\mathrm{T}}(t)(PF_i - Y_i)x(t-\tau_i)\} \\
&+ \sum_{i=1}^{N} \left\{ \int_{t-\tau_i}^{t} \dot{x}^{\mathrm{T}}(\alpha) Z_i \dot{x}(\alpha)\,\mathrm{d}\alpha \right\}.
\end{aligned}$$

Since $\dot{V}_2$ and $\dot{V}_3$ yield the relation

$$\begin{aligned}
\dot{V}_2 = \sum_{i=1}^{N} \Bigg\{ & \tau_i \left[Fx(t) + \sum_{i=1}^{N} \{F_i x(t-\tau_i)\} \right]_i^{\mathrm{T}} \\
&\times \left[Fx(t) + \sum_{i=1}^{N} \{F_i x(t-\tau_i)\} \right] \\
&- \sum_{i=1}^{N} \left\{ \int_{t-\tau}^{t} \dot{x}^{\mathrm{T}}(\alpha) Z \dot{x}(\alpha)\,\mathrm{d}\alpha \right\},
\end{aligned}$$

$$\dot{V}_3 = \sum_{i=1}^{N} \{x^{\mathrm{T}}(t)Q_i x(t) - x^{\mathrm{T}}(t-\tau_i)Q_i x(t-\tau_i)\},$$

we have the derivative of V as follows:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 + \dot{V}_3 \leqslant \bar{x}^{\mathrm{T}} \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^{\mathrm{T}} & X_{22} \end{bmatrix} \bar{x},$$

where

$$\bar{x} \triangleq [x^{\mathrm{T}}(t) \quad x^{\mathrm{T}}(t-\tau_1) \quad \cdots \quad x^{\mathrm{T}}(t-\tau_N)]^{\mathrm{T}},$$

$$\begin{aligned}
X_{11} \triangleq{}& F^{\mathrm{T}}P + PF \\
&+ \sum_{i=1}^{N} \{Y_i + Y_i^{\mathrm{T}} + \bar{\tau} X_i + Q_i + \bar{\tau} F^{\mathrm{T}} Z_i F\},
\end{aligned}$$

$$X_{12} \triangleq \begin{bmatrix} F_1 - Y_1 + \sum_{i=1}^{3} \tau F^{\mathrm{T}} Z_i F_1 \\ \cdots PF_N - Y_N + \sum_{i=1}^{3} \tau F^{\mathrm{T}} Z_i F_N \end{bmatrix},$$

$$X_{22} \triangleq -\operatorname{diag}\{Q_1, \ldots, Q_N\} + \begin{bmatrix} \sum_{i=1}^{N} \tau F_1^{\mathrm{T}} Z_i F_1 & \cdots & \sum_{i=1}^{N} \tau F_1^{\mathrm{T}} Z_i F_N \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} \tau F_N^{\mathrm{T}} Z_i F_1 & \cdots & \sum_{i=1}^{N} \tau F_N^{\mathrm{T}} Z_i F_N \end{bmatrix}.$$

Therefore, if

$$\begin{bmatrix} X_{11} & X_{12} \\ X_{12}^{\mathrm{T}} & X_{22} \end{bmatrix} < 0, \tag{33}$$

system (6) is asymptotically stable according to the Lyapunov–Krasovskii stability theorem (Hale & Lunel, 1993). Eq. (33) can be rewritten into

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^{\mathrm{T}} & Y_{22} \end{bmatrix} + \mathscr{F}^{\mathrm{T}} \mathscr{Z} \Gamma^{-1} \mathscr{Z} \mathscr{F} < 0, \tag{34}$$

where

$$Y_{11} \triangleq F^{\mathrm{T}} P + PF + \sum_{i=1}^{N} \{Y_i + Y_i^{\mathrm{T}} + \tau X_i + Q_i\},$$

$$Y_{12} \triangleq [PF_1 - Y_1 \quad \cdots \quad PF_N - Y_N],$$

$$Y_{22} \triangleq -\operatorname{diag}\{Q_1, \ldots, Q_N\},$$

By the Schur complement (Boyd, Ghaouli, Feron, & Balakrishnan, 1994), Eq. (34) is equivalent to the first inequality in Eq. (9). This completes the proof. □

References

Asok, R., & Yoram, H. (1988). Integrated communication and control systems: Part ii—Design consideration. *Journal of Dynamic Systems, Measurement, and Control*, *110*, 374–381.

Beauvais, J. P., & Deplanche, A.-M. (1995). Heuristics for scheduling periodic complex real-time tasks in a distributed system. *Proceedings of the IECON'95* (pp. 55–60).

Boyd, S., Ghaoui, L. E., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. Philadelphia, PA: SIAM.

Branicky, M. S., Phillips, S. R., & Zhang, W. (2000). Stability of networked control systems: Explicit analysis of delay. *Proceedings of the American control conference*, vol. 4 (pp. 2352–2357).

Cavalieri, S., Stefano, A., & Mirabella, O. (1995). Pre-run-time scheduling to reduce schedule length in the fieldbus environment. *IEEE Transactions on Software Engineering*, *21*, 865–880.

Feng-Li Lian, Moyne, J., & Tilbury, D. (2002). Network design consideration for distributed control systems. *IEEE Transactions on Control Systems Technology*, *10*, 297–307.

Hale, J. K., & Lunel, S. M. V. (1993). *Introduction to functional differential equations*. New York: Springer.

Hong, S. H. (1995). Scheduling algorithm of data sampling times in the integrated communication and control systems. *IEEE Transactions on Control System Technology*, *3*, 225–231.

Hong Ye, Walsh, G. C., & Bushnell, L. G. (2001). Real-time mixed-traffic wireless networks. *IEEE Transactions on Industrial Electronics*, *48*, 883–890.

Krtolica, R., Özgüner, Ü., Chan, H., Göktas, H., Winkelman, J., & Liubakka, M. (1994). Stability of linear feedback systems with random communication delays. *International Journal of Control*, *59*, 925–953.

Li, X., & de Souza, C. E. (1997a). Criteria for robust stability and stabilization of uncertain linear systems with state delays. *Automatica*, *33*, 1657–1662.

Li, X., & de Souza, C. E. (1997b). Delay-dependent robust stability and stabilization of uncertain linear delay systems: A linear matrix inequality approach. *IEEE Transactions on Automatic Control*, *42*, 1144–1148.

Moon, Y. S., Park, P. G., Kwon, W. H., & Lee, Y. S. (2001). Delay-dependent robust stabilization of uncertain state-delayed systems. *International Journal of Control*, *74*, 1447–1455.

Mori, T., Fukuma, N., & Kuwahara, M. (1981). Simple stability criteria for single and composite linear systems with time delays. *International Journal of Control*, *34*, 1175–1184.

Nilson, J. (1998). *Real time control systems with delays*. Ph.D. dissertation, Department of Automatic control, Lund Institute of Technology, Lund, Sweden.

Park, P. G. (1999). A delay-dependent stability criterion for systems with uncertain time-invariant delays. *IEEE Transactions on Automatic Control*, *44*, 876–877.

Park, H. S., Kim, Y. H., Kim, D. S., & Kwon, W. H. (2002). A scheduling method for network based control systems. *IEEE Transactions on Control System Technology*, *10*, 318–330.

Su, T.-J., & Huang, C.-G. (1992). Robust stability of delay dependence for linear uncertain systems. *IEEE Transactions on Automatic Control*, *37*, 1656–1659.

Walsh, G. C., & Hong Ye. (2001). Scheduling of networked control systems. *IEEE Control Systems Magazine*, *21*, 57–65.

Wei Zhang, Branicky, M. S., & Phillips, S. M. (2001). Stability of networked control systems. *IEEE Control Systems Magazine*, *21*, 84–99.

Zuberi, K. M., & Shin, K. G. (1997). Scheduling messages on controller area network for real-time cim applications. *IEEE Transactions on Robotics and Automation*, *13*, 310–316.