Systematic Scheduling Method for Messages and Tasks in Distributed Control Systems

Hong Seong Park, Hyoung Yuk Kim, Weon Joon Kang
hspark@cc.kangwon.ac.kr, (petrus, tailhook)@control.kangwon.ac.kr
BK21 Dept. of Electrical and Computer Eng, Kangwon National University

Abstract **- This paper presents a systematic scheduling method to guarantee the end-to-end constraints including precedence constraints of tasks, messages and both. The presented systematic method is the integrated or joint one of both tasks executed in each node and messages transmitted via the network. The presented method is designed to apply to the general distributed control system that has multiple loops and in a single loop has sensor nodes with multiple sensors, actuator nodes with multiple actuators, controller nodes with multiple tasks, and several constraints. The worst-case response time is analysed using the proposed method.**

I. INTRODUCTION

Nowadays, the distributed control via fieldbus systems becomes one of very important requirements in the field area but has a following restriction: a designer of the distributed control system should consider some constraints such as timing constraints, execution times of tasks, allocation of tasks in each node and their priorities, and precedence relationship of either tasks or messages. These constraints make the system design difficult.

Considering a current distributed control system, the processing power and the transmission speed via a network are increasing. This means that one processor or node should execute several kinds of tasks and a certain task exchanges several types of messages with other tasks. For examples, sensor nodes (actuator nodes) execute several tasks such as sampling (output operation), diagnostics, and communication. Controller executes more complex tasks than sensor/actuator nodes, examples of which are computing of control values, diagnostics, sending and receiving of data, monitoring of data, processing of events, and logging of data. For an example, if a task for sending (or receiving) message to (or from) actuators (or sensors) has lower priority than task for diagnostics, the former is completed after the latter is finished. In other words, the worst-case response time is changed according to the priorities of the tasks. Note that an example of the worst-case response time is the time elapsed from sampling time from sensor to the time to output the control value to a plant via actuator. That is, the execution time of these tasks as well as transmission time of messages should be considered because they affect the end-to-end timing or the worst-case response time.

Researches on the real-time scheduling can be divided into three categories: scheduling for tasks in a processor [1-4], scheduling for messages via a network [5-7] and scheduling for an integrated model of both tasks and messages [8-10]. There have been a few researches on joint scheduling method considering both task execution time and message transmission time at the same time, via which the worst-case response time was analysed [8-10]. In [8-9], the systematic scheduling method was suggested under assumptions that a controller node has only one computation task and a sensor(actuator) node has one sensor(actuator) and the CAN protocol is used. In the suggested method, the tasks running in a single processor(node), multiple inputs and multiple outputs in a single loop weren't considered. In addition, the control value can output to a plant after new data is sampled from the plant, which makes the performance of the control system deteriorate. In [10], the heuristic scheduling method was proposed under assumptions that no task can receive more than one message and TDMA protocol is used. Since one task receives only one message, the proposed method is not suitable in a general control system, which receives multiple messages from multiple analog/digital sensors, computes control values from several tasks, and transmits one or more values to one or more actuators at the same time. In addition, the proposed method can be difficult to be used in a case that several constraints in the control system such as precedence relation are applied.

This paper suggests a systematic joint scheduling method of messages and tasks used in general distributed control systems, in which multiple control loops exists and multiple sensor/actuator nodes and the controller in a single control loop have several computation tasks, which affect the worst-case response time, and all the tasks in these nodes can receive one or more messages. Also this paper analyses the worst-case response time. The proposed method is based on [8] but is the extended method, which solves the sampling problem discussed above and considers multiple tasks, executed in all the nodes and their priorities.

In Section 2, problem statements are discussed. In Section 3 intermediate constraints are derived and finally, we give some conclusions.

II. PROBLEM STATEMENTS

To meet the end-to-end constraints, we should consider all the relationship from sensors to actuators so that some relationships such as precedence of either tasks or messages should be derived. If the output task of the actuator node is executed over the next sampling time of the sampling task, it may make the performance of system worse. That is, it means that the data is sampled before the control value has an effect on the plant.

To solve this problem, the additional constraint for the systematic scheduling is necessary.

Constraint: The sum of initial phase time and deadline of the actuator task at a control loop has to be less than or equal to the sensor task.

$$\phi_{actuator} + d_{actuator} \le T_{sensor}$$

where T_i : Period of the task i,

ϕ_i : Initial phase time of the task i, and

d_i : Deadline of the task i.

If the above constraint is not satisfied on solving intermediate constraints, the parameters of tasks have to be recalculated repeatedly with larger periods. But larger period makes a later initial phase time because of an assumption that the deadline is equal to period. It is well known that the performance of system is better as the period is smaller and the worst-case response time has a limited value. Note that the worst-case response time is smaller than or equal to the period. So, it is necessary to find more an optimal period to minimize initial phase time.

In the system based on pre-emptive, independent tasks and fixed task priorities, the worst-case execution time, $R^T{}_i$ of the task i was presented in [1] as

$$R_i^T = B_i^T + \sum_{\forall j \in hp(\tau_i)} \left\lceil \frac{R_i^T}{T_j^T} \right\rceil C_j^T + C_i^T \qquad (1)$$

where B_i^T : Longest time that the task i can be blocked by a lower priority task to complete its use of protected data,

T_j^T : Period of the task j,

C_i^T : Execution time of the task i,

$hp(\tau_i)$: Set of tasks having higher priority than $\tau_i \in \{\tau_1, \tau_2, ..., \tau_n\}$, and

τ_i : the task i.

And the worst-case transmission times of CAN, R_i^m of the message i[10] is

$$R_i^m = B_i^m + \sum_{\forall j \in hp(m_i)} \left\lceil \frac{R_i^m}{T_j^m} \right\rceil C_j^m + C_i^m \qquad (2)$$

where B_i^m : Longest time that m_i can be delayed by a lower priority messages,

T_j^m : Period of the message j,

C_i^m : Transmission time of the message i,

$hp(m_i)$: Set of messages having higher priority than $m_i \in \{m_1, m_2, ..., m_n\}$, and

m_i : the message i.

Because R_i^τ satisfies the following condition (3), the deadline of the task i can be shorter.

$$T_i^T \ge R_i^T \ , \ d_i^T \ge R_i^T \qquad (3)$$

There is a virtual communication task that executes data transmission between two ports or nodes as shown in Fig.1. A virtual communication task is not a real task but represents the state that data is being transmitted.

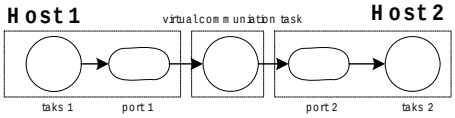

Fig.1 Virtual Communication Task

III. TASK-BASED SCHEDULING METHOD

A. System Model and Task Model

An example of the target system considered in this paper consists of multiple control loop, where each control loop has multiple sensors and multiple actuators connected via CAN, as shown in Fig.2.

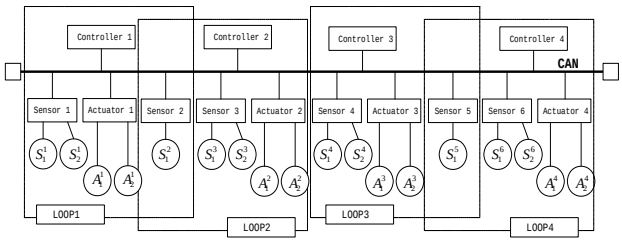

Fig. 2 Target System with Multiple loops

Totally 4 control loops are considered, one of which consists of two sensor nodes, a controller node and an actuator node. Some sensor value in nodes such as Sensor2 and Sensor5, are transmitted to multiple controller nodes at same time. For an example, the 1st control loop in Fig.2 consists of Sensor1, Sensor2,

Controller1 and Actuator1. And the 2nd control loop consists of Sensor2, Sensor3, Controller2, and Actuator2.

Generally sensor nodes have the same number of sampling task as the number of sensors. But in this case, it's not easy to synchronize data input from several sensors. In a case that several sensors can have effects on an actuator, input data synchronization might be the important constraint because sensor data could be valid when they were sampled in the specified input data synchronization time. Therefore, as shown in Fig.3(b), a task, called sampling server task τ_0, is added to Fig.3(a) to guarantee input data synchronization. It samples all sensors within the input data synchronization time. And then, as shown in Fig.3(c), we integrated τ_0 with τ_1, τ_2 into one task named τ_s.

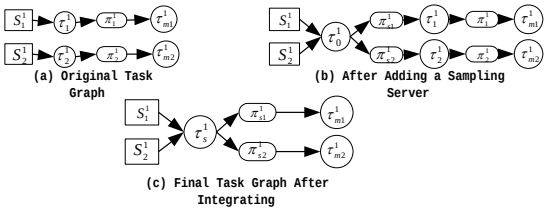

Fig.3 Modification of task graph in Sensor node

In Fig.3, τ_s denotes as a sampling server, π_{si} as port for transmitting to network transmitting to network. In controller nodes, it is assumed there are several tasks whose only one task is the control task. The task graph of the whole system is shown in Fig.4.

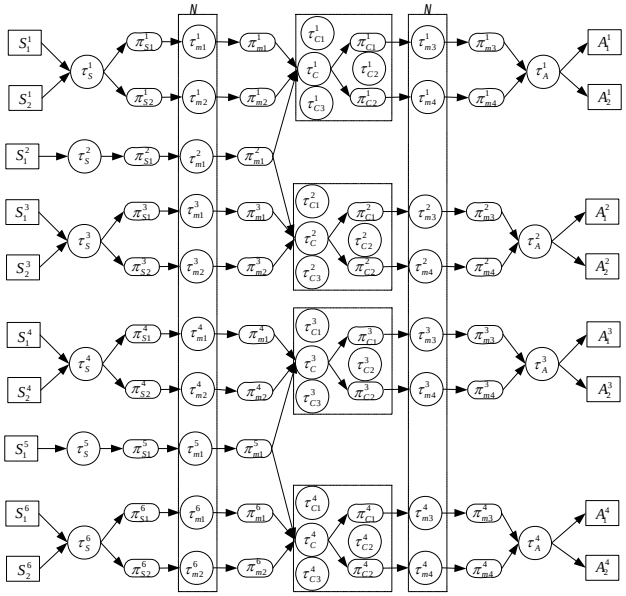

Fig.4 The Task Graph of Whole System

B. Deriving Intermediate Constraints

In this subsection, intermediate constraints will be derived, which represents equations or inequalities for the period, initial phase time and deadline of tasks. Those intermediate constraints consist of two constraints sets: set on period and set on phase and deadline. Constraints on period are as follows:

$T_S^1 \mid T_{m1}^1,\ T_S^1 \mid T_{m2}^1,\ T_S^2 \mid T_{m1}^2,\ T_{m1}^1 \mid T_C^1,\ T_{m2}^1 \mid T_C^1,\ T_{m1}^2 \mid T_C^1$
$T_C^1 \mid T_{m3}^1,\ T_C^1 \mid T_{m4}^1,\ T_{m3}^1 \mid T_A^1,\ T_{m4}^1 \mid T_A^1$

where T_j^i is the period of task τ_j^i

S_j^i	The j-th sensor in the i-th sensor node (i=1 ~ 6, j=1,2)
τ_s^i	The sampling server task in the i-th sensor node (i=1 ~ 6)
π_{sj}^i	Port for transmitting the j-th sensor value in i-th sensor node (i=1 ~ 6, j=1,2)
τ_{mj}^i	Virtual communication tasks for π_{sj}^i
π_{mj}^i	Port for the j-th message from i-th sensor node (j=1,2, i=1~6) Port for the j-th message from i-th control node (j=3,4, i=1~4)
τ_c^i	The control task in the i-th control node (i=1~4)
τ_{c1}^i	Sporadic task in the i-th control node (i=1~4)
τ_{cj}^i	Periodic task in the i-th control node (i=1~4, j=2,3)
π_{cj}^i	Port for transmitting the j-th control value in i-th control node (i=1~4, j=1,2)
τ_A^i	The actuator task in the i-th actuator node (i = 1~4)
A_j^i	The j-th actuator in the i-th actuator node (I, j =1~4)

Table.1 The notation for Fig. 4

and constraints on phase and deadline are

$$\phi_A^1 - \phi_S^1 + d_A^1 \le MAVT^1 \ ,\ \phi_A^1 - \phi_S^2 + d_A^1 \le MAVT^1$$
$$\phi_A^1 - \min(\phi_S^1, \phi_S^2) + d_A^1 \le MAVT^1$$
$$\phi_A^1 + d_A^1 \le T_S^1 \ ,\ \phi_A^1 + d_A^1 \le T_S^2$$
$$\phi_A^1 + d_A^1 \le \min(T_S^1, T_S^2)$$
$$\phi_C^1 \ge \phi_S^1 + d_S^1 + \max(d_{m1}^1, d_{m2}^1) \ ,\ \phi_C^1 \ge \phi_S^2 + d_S^2 + d_{m1}^2$$
$$\phi_C^1 = \max(\phi_S^1 + d_S^1, \phi_S^2 + d_S^2) + \max(d_{m1}^1, d_{m2}^1, d_{m1}^2)$$
$$\phi_A^1 = \phi_C^1 + d_C^1 + \max(d_{m3}^1, d_{m4}^1)$$

where $MAVT^1$ is Maximum Allowable Validity Time or deadline of the loop 1 and ϕ_j^i is the initial phase time of task τ_j^i.

Constraints for the 2nd, 3rd and 4th loops are derived using the same way.

C. Solving Intermediate Constraints

After deriving intermediate constraints, they can be solved by 3 pruning steps[8,9]. The applicable period sets of each task is calculated by time granularity, utilization and harmonicity pruning with a time granularity of 5 and utilization factor of 0.8. After selecting a period for each task in the applicable period sets, deadline of each control task and each message task can be calculated using (1),

(2) and (3) and finally starting time, priority, period and deadline are obtained as shown in Table 2.

It is shown in Fig.5 that the end-to-end worst-case response time of each control loop increases as the number of the tasks with higher priority than control task increases in the control node. That is, Fig.5 shows the effect of the number of tasks in the single processor, it can be known from Fig.5 that the priorities of tasks related to loop are very important.

The process listed above is summarized in Fig. 6.

Loop 1 (MAVT: 60)								Loop 2 (MAVT: 80)								Loop 3 (MAVT: 100)								Loop 4 (MAVT: 120)							
Task	Priority		e	T	R	D	ϕ	Task	Priority		e	T	R	D	ϕ	Task	Priority		e	T	R	D	ϕ	Task	Priority		e	T	R	D	ϕ
	T	M							T	M							T	M							T	M					
τ_s^1	1		2	40	2	2	0	τ_s^3	1		3	80	3	3	0	τ_s^4	1		3	60	3	3	0	τ_s^5	1		3	120	3	3	0
τ_{m1}^1		1	1	40	0.2	1	2	τ_{m1}^3		4	1	80	0.5	1	3	τ_{m1}^4		10	1	60	1	1	3	τ_{m1}^5		13	1	120	1.2	1	3
τ_{m2}^1		2	1	40	0.3	1	2	τ_{m2}^3		5	1	80	0.6	1	3	τ_{m2}^4		11	1	60	1.1	2	3	τ_{m2}^5		14	1	120	1.3	1	3
τ_s^2	1		1	40	1	1	0	τ_s^3	1		1	40	1	1	0	τ_s^5	1		1	60	1	1	0	τ_s^5	1		1	60	1	1	0
τ_{m1}^2		3	1	40	0.4	1	1	τ_{m1}^3		3	1	40	0.4	1	1	τ_{m1}^5		12	1	60	1.2	2	1	τ_{m1}^5		12	1	60	1.2	2	1
τ_{c1}^1	1		4	20	4	4	0	τ_{c1}^3	1		4	20	4	4	0	τ_{c1}^4	1		4	20	4	4	0	τ_{c1}^4	1		4	20	4	4	0
τ_c^1	2		11	40	15	15	4	τ_c^3	2		13	80	17	17	4	τ_c^3	2		15	60	19	19	5	τ_c^4	2		14	120	18	18	4
τ_{c2}^1	3		6	50	25	25	19	τ_{c2}^3	3		6	50	27	27	21	τ_{c2}^3	3		6	50	29	29	24	τ_{c2}^4	3		6	50	28	28	22
τ_{c3}^1	4		8	60	33	33	52	τ_{c3}^3	4		8	60	35	35	48	τ_{c3}^3	4		8	60	37	37	58	τ_{c3}^4	4		8	60	36	36	50
τ_{m3}^1		6	1	40	0.6	1	19	τ_{m3}^3		8	1	80	0.8	1	21	τ_{m3}^3		15	1	60	1.4	2	24	τ_{m3}^4		17	1	120	1.6	2	22
τ_{m4}^1		7	1	40	0.7	1	19	τ_{m4}^3		9	1	80	0.9	1	21	τ_{m4}^3		16	1	60	1.5	2	24	τ_{m4}^4		18	1	120	1.6	2	22
τ_A^1	1		2	40	2	2	20	τ_A^2	1		4	80	4	4	22	τ_A^2	1		4	60	4	4	26	τ_A^4	1		4	120	4	4	24

Table 2. The results of scheduling tasks

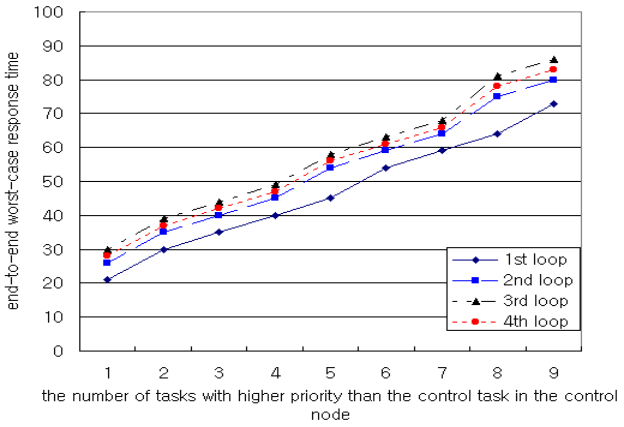

Fig.5 The effect of tasks in the control node

IV. CONCLUSIONS

This paper suggested the systematic scheduling methodology for the general fieldbus-based distributed control system, which has multiple loops, all the nodes in the loop are connected to CAN and in a single loop has several sensor nodes with multiple sensors, several actuator nodes with multiple actuators, a controller node with multiple tasks such as control task, event processing task.

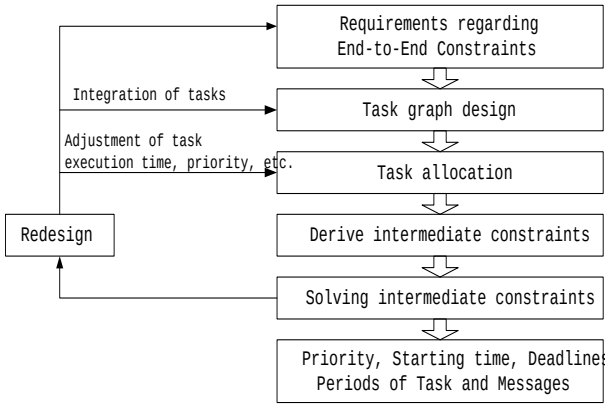

Fig.6 The process for solving the end-to-end constraints

This paper suggested the systematic joint scheduling method of tasks and messages to guarantee the end-to-end constraints including several types of precedence constraints for the distributed control system, via which the initial phase, priority, and period of each task and of each message can be derived easily and the effects of priorities of tasks can be known.

REFERENCES

[1] A. Burns, "Preemptive priority based scheduling: An appropriate engineering approach," in Principles of Real-Time Systems(Ed. S. Son), Prentice Hall, 1994.

[2] K. Ramamritham and J.A. Stankovic, "Scheduling algorithm and operating systems support for real-time systems," Proceedings of IEEE, pp. 55-67, Jan. 1994.

[3] N.C. Audsley, A. Burns, and A.J. Wellings, "Deadline monotone scheduling theory and application," IFAC J. Control Engr. Practice, Vol. 1, No. 1, pp.71-78, 1993.

[4] J. Xu and D. Parnas, "Scheduling processes with release times, deadlines, precedence and exclusion relations," IEEE Tr. on Software Engineering, pp.360-369, Mar., 1990.

[5] P. Lorenz and Z. Mammeri, " Real-time Software Architecture: Application to FIP filedbus," Proc of 1995 AARTC, pp.415-423, 1995.

[6] P. Raja and G. Ulloa, "Priority Polling and Dynamic Time-Window Mechanisms in a Multicycle Fieldbus," Proc of 1993 COMPEURO, pp.452-460, 1993.

[7] K. Tindell, H. Hansson, and A. Wellings, "Analyzing Real-time Communications: Controller Area Network," IEEE Real-time Systems Symposium, 1994.

[8] J.W. Park, Y.S. Kim, et al,"Network conscious of Distributed Real-Time Systems," Journal of System Architecture, pp. 131-156, 1998.

[9] Y.S.Kim, H.S.Park, and W.H.Kwon, "An Architecture for a Network Based Robot Control System", ETFA'99, Vol2, pp.875-880, Oct., 1999.

[10] K. Tindell,"Holistic Schedulability Analysis for Distributed Hard Real-time Systems," Report, Dept. of Computer Science Report, Univ. of York.