International Journal of Automotive Technology, Vol. 6, No. 1, pp. 71−77 (2005)

OPTIMAL PERIOD SELECTION TO MINIMIZE THE END-TO-END RESPONSE TIME

M. SHIN[1)], W. LEE[2)] and M. SUNWOO[1)*]

[1)]Department of Automotive Engineering, Hanyang University, Seoul 133-791, Korea
[2)]Department of Control and Instrumentation Engineering, Changwon National University, Gyeongnam 641-773, Korea

(Received 30 July 2003; Revised 12 February 2004)

ABSTRACT−This paper presents a systematic approach which determines the optimal period to minimize performance measure subject to the schedulability constraints of a real-time control system by formulating the scheduling problem as an optimal problem. The performance measure is derived from the summation of end-to-end response times of processed I/Os scheduled by the static cyclic method. The schedulability constraint is specified in terms of allowable resource utilization. At first, a uniprocessor case is considered and then it is extended to a distributed system connected through a communication link, local-inter network, LIN. This approach is applied to the design of an automotive body control system in order to validate the feasibility through a real example. By using the approach, a set of optimal periods can easily be obtained without complex and advanced methods such as branch and bound (B&B) or simulated annealing.

KEY WORDS : End-to-end response time, LIN, Static cyclic schedule, Optimization

NOMENCLATURE

B	: blocking time
C	: worst-case execution time
I	: interference time
J	: jitter or cost function
R	: worst-case response time
T	: period
U	: utilization
τ	: time delay
w	: weighting factor

SUBSCRIPTS

b	: communication bus
d	: delay
n	: node
t	: task
a	: actuation
i, j, k	: the index of node or task or message
s	: sampling

1. INTRODUCTION

Distributed real-time computing systems are becoming more and more important in the automotive industry in order to optimize system performance and costs. Due to the distributed structure that controllers, sensors and actuators are connected by a communication system, they should be designed more carefully considering the characteristics of real-time and distributed system.

In this paper, a system which is scheduled by a static cyclic scheduler is considered. The static cyclic scheduler is one of the simplest forms of real-time schedulers. The scheduler is still widely applied to safety critical applications such as brake-by-wire system and the flight safety system despite its simplicity and several limitations, because it guarantees the predictability of a system in time and relieves operating load for task scheduling. In the static cyclic scheduling method, a set of tasks is ordered in a cycle which is repeated over and over again and the schedulability of the task set is completely determined by given periods and task execution times. Therefore, the choice of task period is one of the most important processes in the design phase.

However, in the automotive industry, a task period has usually been determined by control engineers who aren't concerned with task schedulability requirements in the implementation phase. From a control point of view, the choice of sampling period in the design phase is often determined by system dynamics without any consideration of schedulability. Control engineers assume that the software engineers, who implement a system designed by

**Corresponding author.* e-mail: msunwoo@hanyang.ac.kr

control engineers, will make a complete implementation that can satisfy all requirements within the given period. But from a scheduling point of view, the allowable resource utilization sets a lower bound on the period. Thus, viewpoints on the task period are different between control and software engineers. However, in order to achieve a better system design, the task schedulability as well as system dynamics should be taken into account in selecting the period at design time.

Recently, many researchers have investigated several approaches to reduce the gap between two different viewpoints by considering system performance and real-time characteristics at the same time in the design phase.

Gerber, Hong, and Saksena (1994) investigated the interaction between task schedulability and system performance from a real-time system design point of view. In their study, they proposed a systematic method to derive the timing attributes of tasks from timing constraints which are imposed on external inputs and outputs.

An optimal solution to a combined task assignment and scheduling problem for communicating periodic tasks in distributed real-time systems was developed by Peng, Shin and Abdelzaher (1997). The maximum normalized task response time called the system hazard, is minimized by utilizing a B&B algorithm, subject to the intercommunication constraints among the tasks to be allocated and scheduled.

From a scheduling point of view, Seto, Lehoczky, Sha and Shin (1996) developed an algorithm to determine optimal task periods, where all tasks were scheduled by the earliest deadline first scheduling algorithm. They carried out a trade-off analysis between system performance and task schedulability by formulating an optimization problem. Based on the algorithm, Seto, Lehoczky and Sha (1998) proposed a search algorithm for the optimal periods when the tasks are scheduled with rate monotonic algorithm.

Rehbinder and Sanfridson (2000) have proposed a method to generate static schedules and to calculate the loss function, which is a measure of control performance, associated with each schedule. They used an integrated approach to the scheduling of a set of optimal control tasks on a uniprocessor via periodic LQ-theory that is formulated as a combinatorial optimization problem.

In previously proposed optimization frameworks, researchers firstly found all the sets of feasible periods to guarantee schedulability by using several heuristic searching algorithms. And then, they chose a set of periods in order to make the cost function have minimum value among the sets as the optimal periods. But, in the proposed approach, a set of optimal periods is obtained by a numerical method without heuristic searching algorithms by formulating the problem as a general static optimization problem. This approach is possible because all periods are selected as design variables instead of task priorities that are regarded as design variables by most previous researchers. Moreover, the performance measure is defined with the sum of end-to-end response times of all processed I/Os that is considered as a constraint by previous studies.

This paper is outlined as follows: In section 2, the timing model for a uniprocessor is derived and the optimal period selection problem is formulated as an optimal design problem. The approach proposed in section 2 is extended to a distributed system using LIN as a communication link in section 3. Section 4 gives two examples, where the theory proposed in the paper is applied to an automotive body control system example connected by LIN. Finally, conclusions and some directions for future work are discussed in section 5.

2. PROBLEM FORMULATION FOR A UNIPROCESSOR

In this section, a timing model for a target system is described. The problem of finding all the periods $[T_1 \dots T_n]$ for a set of tasks $\{t_1 \dots t_n\}$ that are scheduled on a uniprocessor with the static cyclic method will be considered. In order to solve the problem, the scheduling problem is formulated as an optimal design problem where performance measure and constraint are specified by response time and resource utilization respectively.

2.1. Task Model on a Unipocessor

The period selection problem of a set of independent control tasks is considered on a uniprocessor with following assumptions.

(1) All tasks are scheduled based on a static cyclic scheduling algorithm and any tasks cannot overlap in time.
(2) All tasks execute periodically according to a predefined interval called a period.
(3) Each task is independent of other tasks, i.e. tasks cannot have precedence relations and cannot share resources in an exclusive mode.
(4) Each I/O is independently processed by a task (see

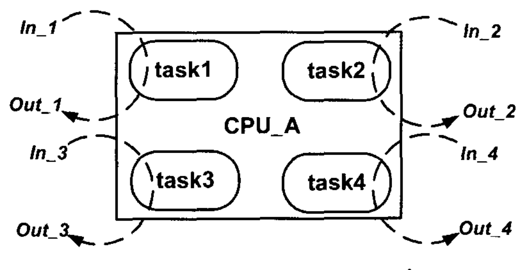

Figure 1. Independent I/O process on a uniprocessor.

Figure 1).

(5) Every periodic task has constant worst-case execution time (WCET) and utilization upper bound of the uniprocessor which are given.

(6) The WCET of task i is less than or equals to its period, i.e. $C^i \le T^i$.

In the case of priority based real-time system worst-case response time of a task i, R^i, is given by the equation (1).

$$R^i + C^i + B^i + I^i + J^i \tag{1}$$

where, C^i is the computational requirement for one input, I^i is the sum of the computational requirements for all inputs from higher levels occurring in the interval $[0, R^i)$, B^i is the blocking time that a task can be delayed by the execution of lower priority tasks and J^i is sampling jitter.

In this task model, I^i and B^i equal to zero due to the assumption 1 and 3. The sampling jitter of the periodic task is less than the period of the task. Therefore, the maximum response time of task i can be rewritten as follows:

$$R^i=(J^i + C^i),\ R^i_{\max}=(T^i + C^i) \tag{2}$$

By assumption 4, the worst-case response time of a task, R^i, equals to the end-to-end response time of I/O processed by the task, $R^{IO}_{end-to-end}$.

$$R^i=R^{IO}_{end-to-end} \tag{3}$$

2.2. Period Optimization on a Unipocessor

In order to choose the optimal periods of tasks on a uniprocessor, a cost function reflecting the characteristics of a target system has to be defined as a performance measure.

In a real-time system, the utilization of resources is limited and the performance of a system is often inversely related to the end-to-end response time, i.e. the shorter end-to-end response time generally means the better performance. In other words, it is obvious that the end-to-end response time should be as small as possible in order to satisfy the condition of limited utilization for better performance. Due to these features, finding the optimal periods can be led to the optimization problem with several constraints.

In the case of uniprocessor, the sum of end-to-end response time of each task is taken as a performance measure, J. The available utilization of a uniprocessor is used as an inequality constraint. As a result, the optimaization problem is formulated as follows:

$$\min\ J=\sum_{i=1}^{N_t} w^i R^i \tag{4}$$

$$\text{subject to } \sum_{i=1}^{N_t} \frac{C^i}{T^i} \le A,\quad 0 < A \le 1$$

where, w^i is a weighting factor for task i representing the task importance, A is allowable utilization of CPU and N_t is the number of tasks.

This is a kind of general static optimization problem and there are many algorithms to solve this problem.

3. PROBLEM FORMULATION FOR A DISTRIBUTED SYSTEM

The approach proposed in the previous section will be extended to manage a distributed system that all nodes are connected through a communication link. In this study, LIN-bus is used for the communication link performed in single-master/multiple-slaves method.

3.1. Network Model for a Distributed System

The period selection problem is extended to distributed system with following assumptions.

(1) LIN-bus is used for a communication link and LIN master sends messages based on the static cyclic method.

(2) Reception (Rx) and transmission (Tx) are processed by interrupt service routine (ISR) for the interface with application program and communication data.

(3) Tasks that take part in communication have the same characteristics as they do in the uniprocessor.

(4) Only one message is required to process one distributed independent I/O. Sampling task and actuation

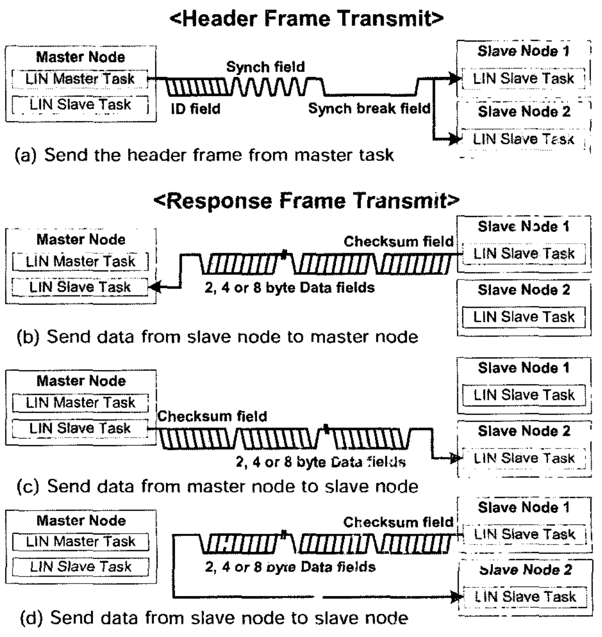

Figure 2. LIN communication method.

task that handle the I/O in two nodes are connected by the message.

Figure 2 shows the communication model for several distributed tasks connected through LIN bus. A header frame, which has synchronization field and identification (ID) field, is transmitted by a master task in a master node. Then, a response frame is sent by one slave task in the master or slave nodes according to the received ID field. It consists of three to nine byte fields, which are composed of two, four or eight bytes data fields determined by ID field and one byte checksum field for error checking.

Figure 3 shows the timing behavior of a distributed system. A sampling task which is activated periodically reads an external input value and stores the processed input value into a transmission data register for C^s as its output. ISR routine for message transmission may send the data immediately when a received message interrupt by an appropriate header frame occurs. The transmission mechanism of a message i, is as follows: at first, LIN master task in the master node sends a header frame every message transmission period, T_b^i, in order to request a message transmission. Next, Rx ISR in the sampling node immediately sends a response frame to an actuation node according to the request of the master task. When the response frame is completely sent to the actuation node, Rx ISR in the node immediately stores the received value to an application buffer so that the application routine may freely access the received data. The time duration from the beginning transmission of a header frame in the master node to the end of Rx ISR in the actuation node is represented as C_b^i.

The actuation task reads periodically the value stored in the application buffer as its input and makes an output based on the received value for C^a.

In the network model for a distributed system, several time delays exist which are caused by cyclic task activation and message transmission. In the sampling node, the time difference between an external input change and the start of sampling task is the sampling delay, τ_d^s. An updated message that is stored in the transmission data register by the sampling task should

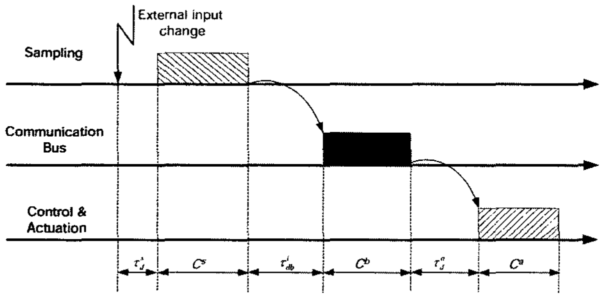

Figure 3. Timing model for a distributed system.

wait for the start of a header frame during τ_{db}^i in order to send the data. Similar to the sampling task, there is a time delay, τ_d^a in the actuation node between the end of message reception and the start of the activation task.

The end-to-end response time of a distributed I/O connected by message i, $R_{end-to-end}^i$, can be represented as equation (5).

$$R_{end-to-end}^i = \tau_d^s + C^s + \tau_{db}^i + C_b^i + \tau_d^a + C^a \tag{5}$$

where $\tau_d^s \le T^s$, $\tau_{db}^i \le T_b^i$, $\tau_d^a \le T^a$

In equation (5), all delays are less than or equal to the period of task activation or message transmission. Therefore, the equation can be rewritten as follows:

$$R_{end-to-end,\max}^i = (T^s + C^s) + (T_b^i + C_b^i) + (T^a + C^a) \tag{6}$$

3.2. Period Optimization on a Distributed System

In order to choose the optimal periods of tasks and messages on a distributed system, a cost function reflecting the characteristics of a target system has to be defined as a performance measure.

In the case of a distributed system, the sum of end-to-end response time of all processed I/Os on a uniprocessor and distributed nodes is taken as a performance measure, J, which is similar to a uniprocessor case. Available utilization of all processors and a communication bus is used as inequality constraints. Then, the formulated optimaization problem is

$$\min\ J = \sum_{k=1}^{N_m} w_b^k R_b^k + \sum_{i=1}^{N_n} \sum_{j=1}^{N_{it}} w^{ij} R^{ij} \tag{7}$$

subject to

$$\sum_{j=1}^{N_{it}} \frac{C^{ij}}{T^{ij}} \le A^i,\ 0 < A^i \le 1 \quad i=1,\ 2,\ \cdots\ N_n$$

$$\sum_{k=1}^{N_m} \frac{C_b^k}{T_b^k} \le A_b,\ 0 < A_b \le 1$$

where R_b^k is the response time of message k, R^{ij} is the response time of task j in node i, N_m is the number of messages, N_n is the number of nodes, N_{it} is the number of task in node i, w_b^k is a weighting factor for message k, w^{ij} is a weighting factor for task j in node i, A_b is allowable utilization of LIN bus, and A_i is allowable utilization of node i.

Additional constraints can be defined according to the application, as equality or inequality constraints to solve the optimization problem, such as $T^1 \le T^2 \le \ldots \le T^n$, $T^i = T^j$ or $T^i = 2T^k$.

4. APPLICATION EXAMPLES

In order to demonstrate the proposed algorithms, an

Table 1. Time attributes of a uniprocessor.

Name	WCET (ms)	Period (ms)	Weighting factor
Allowable utilization = 0.8			
DOOR_T	1.0	15	3
WIN_T	1.5	10	2
MIR_T	2.5	10	1

automotive body control system is considered. The body control system is composed of several modules for the control of doors, mirrors and windows. The static cyclic scheduling method is adopted.

4.1. Example on a Uniprocessor

In this example, all inputs related to a door, a mirror and a window are processed on a uniprocessor. The uniprocessor is implemented as Table 1. Table 1 shows the WCET, period, and weighting factor of each task and allowable utilization of CPU. Using the given periods, CPU utilization, U equals to 0.466 and performance measure, J is 83.5.

In this example, the optimization problem is formulated in equation (8) in order to optimize the given periods to minimize the weighted summation of all end-to-end response times on a given uniprocessor.

$$\begin{aligned} \min\ J &= 3 \times (1.0+T^{DOOR\_T}) \\ &\quad +2 \times (1.5+T^{WIN\_T}) + 1 \times (2.5+T^{MIR\_T}) \\ s.t.\ & \frac{1.0}{T^{DOOR\_T}} + \frac{1.5}{T^{WIN\_T}} + \frac{2.5}{T^{MIR\_T}} \le 0.8 \end{aligned} \tag{8}$$

By solving this optimization problem, the optimal values for T^{DOOR_T}, T^{WIN_T}, T^{MIR_T} and the corresponding minimum performance measure J are obtained as in Table 2. Table 2 shows that the CPU utilization is increased up to the allowable range by reducing task periods in order to minimize the summation of end-to-end response times. By referring the optimal values, a set of periods is selected for real implementation with a harmonic relationship as in Table 2. In this case, U equals to 0.75 and J is 43.5. Figure 4 shows the static cyclic schedule of the uniprocessor with three tasks.

Table 2. Results for the uniprocessor.

Name	WCET (ms)	Optimal period (ms)	Implemented period (ms)	Weighting factor
$U_{optimal} = 0.8$, $J_{optimal} = 40.3181$				
DOOR_T	1.0	3.6411	5	3
WIN_T	1.5	5.4616	5	2
MIR_T	2.5	9.9717	10	1

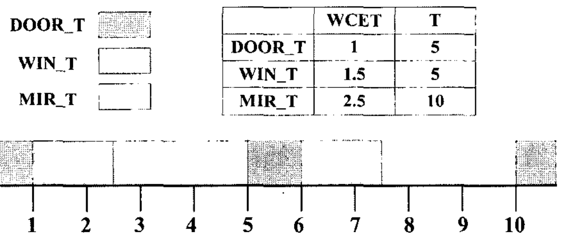

Figure 4. Static cyclic schedule of a uniprocessor with three tasks.

4.2. Example on a Distributed System

In this section, an example that only the function for the mirror in the previous example is distributed by using LIN network is considered. Other tasks irrelevant to the mirror are the same as the previous example. The distributed system is implemented as in Table 3.

Table 3 shows the WCET, period, and weighting factor of each task and message and allowable utilization of all CPUs and LIN bus. LIN_M that is LIN master task, sends a header frame cyclically according to a predefined period and processes all jobs connected with communication such as mode change and communication error handling. MIR_SAM is a task for sampling in a sampling node and MIR_ACT is an actuation task in an actuation node. MIR_MSG presents a LIN message for containing the mirror input. In this example, it is assumed that communication is performed according to the communication model of Figure 2(d). Using the given periods, U_{Master} equals to 0.2833, $U_{Sampling}$ is 0.1, $U_{Actuation}$ is 0.15, $U_{LIN\ bus}$ is 0.125 and J is 177.

Since the period of LIN_M task should be less than or equal to the transmission period of MIR_MSG, additional constraints are required to get more realistic results as follows: $C_b^{MIR\_MSG} \le T^{LIN\_M} \le T_b^{MIR\_MSG}$ and T^{LIN_M} =

Table 3. Time attributes of a distributed system.

Node	Name	WCET (ms)	Period (ms)	Weighting factor
Allowable utilization of all CPUs = 0.8				
Allowable utilization of LIN bus = 0.7				
Master	DOOR_T	1.0	15	3
Master	WIN_T	1.5	10	2
Master	LIN_M	1.0	15	1
Sampling	MIR_SAM	1.0	10	1
Actuation	MIR_ACT	1.5	10	1
LIN bus	MIR_MSG	7.5	60	1

Table 4. Results for the distributed system.

Node	Name	WCET (ms)	Optimal period (ms)	Imple-mented period (ms)	Weight-ing fac-tor
Utilization of all CPUs = 0.8 Utilization of LIN bus = 0.7					
Minimum performance measure = 58.5347					
Master	DOOR_T	1.0	2.8302	5	3
Master	WIN_T	1.5	4.2453	5	2
Master	LIN_M	1.0	10.7143	15	1
Sampling	MIR_SAM	1.0	1.25	5	1
Actuation	MIR_ACT	1.5	1.875	5	1
LIN bus	MIR_MSG	7.5	10.7143	15	1

$k \times T_b^{MIR\_MSG}$ where k is a natural number.

Therefore, the optimization problem is formulated in equation (9) in order to optimize the given periods to minimize the weighted summation of all end-to-end response times on a distributed system.

$$\min\ J = 1 \times (1.0 + T^{LIN\_M}) + 3 \times (1.0 + T^{DOOR\_T}) + 2 \times (1.5 + T^{WIN\_T}) + 1 \times (1.0 + T^{MIR\_SAM}) + 1 \times (1.5 + T^{MIR\_ACT}) + 1 \times (7.5 + T_b^{MIR\_MSG})$$
$$s.t. \begin{cases} \dfrac{1.0}{T^{LIN\_M}} + \dfrac{1.0}{T^{DOOR\_T}} + \dfrac{1.5}{T^{WIN\_T}} \le 0.8 \\ \dfrac{1.0}{T^{MIR\_SAM}} \le 0.8, \quad \dfrac{1.5}{T^{MIR\_ACT}} \le 0.8 \\ \dfrac{7.5}{T_b^{MIR\_MSG}} \le 0.7 \\ C_b^{MIR\_MSG} \le T^{LIN\_M} \le T_b^{MIR\_MSG} \\ T^{LIN\_M} = k \times T_b^{MIR\_MSG} \end{cases} \qquad (9)$$

Table 4 shows the analyzed results. Similar to the previous example, the utilization of CPUs and the LIN bus is increased up to allowable ranges. Also, a set of periods for real implementation with harmonic relationship is redefined in Table 4 by referring to the optimal results. In this case, U_{Master} equals to 0.5667, $U_{Sampling}$ is 0.2, $U_{Actuation}$ is 0.3, $U_{LIN\ bus}$ is 0.5 and J is 81.5.

As explained above, the derived periods which are obtained by the optimization problem can be used as a guideline to determine the minimum bound of period or the relative size of task and message periods.

5. CONCLUSION

In this study, the issue of selecting the periods for a given set of real-time periodic tasks and messages has been studied. The performance of the system is optimized in the sense that the sum of end-to-end response times of processed I/Os is minimized and all the tasks are schedulable with the static cyclic scheduling method. To solve the optimal period selection problem, timing models for task and network are described and the problem is formulated as an optimal design problem. In this formulation, the performance measure and basic constraints are specified by response time and resource utilization respectively using the models. By using the proposed approach, a set of optimal periods can easily be obtained without complex and advanced methods such as B&B or simulated annealing. As confirmed in the previous example, if more specific information about timing attributes of the target system such as maximum period bound, the harmonic relationship between task periods and relative size in period is given, more feasible value for implementation can be acquired by defining additional constraints.

While this study has just concentrated on a system scheduled by the static cyclic method, many real-time systems are operated by priority based scheduling policy. Therefore, the consideration about priority based systems and different protocols that work based on the event is worthwhile to pursue. Also, it is necessary to take into account dependent tasks with the precedence and exclusive relationship.

ACKNOWLEDGEMENT–This research is supported in part by MOST (Ministry of Science and Technology) under the National Research Laboratory program.

REFERENCES

Bettati, R. and Liu, Jane W.-S. (1992). End-to-end scheduling to meet deadlines in distributed systems. *Proceedings of the 12th International Conference on Distributed Computing Systems,* June 1992.

Choi, J. B., Shin, M. and Sunwoo, M. (2004). Development of timing analysis tool for distributed real-time control system. *Int. J. Automotive Technology* **5, 4,** 269–276.

Gerber, R., Hong, S. and Saksena, M. (1994). Guaranteeing end-to-end timing constraint by calibrating intermediate processes. *Proceedings of the IEEE Real-Time Systems Symposium,* December, 1994.

LIN Consortium. (2003) LIN Specification Package 2.0

Liu, C. L. and Layland, J. W. (1973). Scheduling algorithms for multiprogramming in a hard-teal-time environment. *JACM* **20, 1,** 46–61, 1997.

Peng, D. P., Shin, K. G. and Abdelzaher, T. (1997). Assignment and scheduling of communicating periodic task in distributed real-time systems. *IEEE Transaction* *on Software Engineering* **23, 12,** December, 1997.

Rehbinder, H. and Sanfridson, M. (2000). Integration of off-line scheduling and optimal control. *Proceedings of the 12th European Conference on Real-Time Systems*, Stockholm, Sweden, 137–143.

Ryu, M., Hong, S. and Saksena, M. (1997). Streamlining real-time controller design: From performance specifications to end-to-end timing constraints. *Proceedings of Real-Time Applications and Technology Symposium*, June 192–203.

Seto, D., Lehoczky, J. P. and Sha, L. (1998). Task period selection and schedulability in real-time systems. *Proceedings of the 19th IEEE Real-Time Systems Symposium,* 188–198.

Seto, D., Lehoczky, J. P., Sha, L. and Shin, K. G. (1996). On task schedulability in in real-time systems. *Proceedings RTSS* 96, 13–21.

Shin, K. G. and Meissner, C. L. (1999). Adaptation and graceful degradation of control system Performance by task reallocation and period adjustment. *Proceedings of the 11th Euromicro conference on Real-Time Systems,* June.

Shin, M., Lee, W. and Sunwoo, M. (2002). Holistic scheduling analysis of a CAN based body network system. *Transactions of KSAE* **10, 5,** 114–120.

Shin, M., Lee, W., Sunwoo, M. and Han, S. (2002). Development of a body network system with OSEK/VDX standards and CAN protocol. *Transactions of KSAE* **10, 4,** 175–180.

Tindell, K. and Clark, J. (1994). Holistic schedulability analysis for distributed hard-real-time systems. *Microprocessing & Microprogramming – Euromicro Journal* **50, 2-3**, 117–134.