Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100411

Robust Co-Design of Control and Real-Time Scheduling in Cyber-Physical Systems

Kyung-Joon Park[1,*], *Man-Ki Yoon*[2] *and Chang-Gun Lee*[3]

[1] Department of Information and Communication Engineering, DGIST, Daegu, South Korea
[2] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA
[3] School of Computer Science and Engineering, Seoul National University, Seoul, South Korea

Received: 24 Mar. 2016, Revised: 10 May 2016, Accepted: 11 May 2016
Published online: 1 Jul. 2016

Abstract: Cyber-physical systems (CPS) have emerged as a promising research paradigm, which is the convergence of control, communication, and computation. In CPS, real-time transactions visit multiple resources such as sensors, actuators, networks, and microprocessors. One fundamental issue, which is called control and real-time scheduling co-design, is how to maximize control performance of the physical systems while satisfying the real-time constraints imposed by limited computational resources. Although there have been extensive studies on the co-design problem in a single-resource system, multi-resource cases have not been fully studied. In this paper, we propose an optimization framework for robust control design with end-to-end response time constraints in a multi-resource system. We introduce a rigorous robust performance metric from the control theoretic viewpoint. Then, we investigate the impact of end-to-end response time analysis techniques on the control performance. We show that the traditional per-job response time analysis significantly degrades the control performance when real-time tasks visit a resource multiple times. We demonstrate that we can improve the control performance by adopting the per-resource response time analysis. Our simulation results verify the effectiveness of the proposed co-design framework.

Keywords: Cyber-physical systems, control and real-time scheduling co-design, robust system design, optimization

1 Introduction

Recently, a cyber-physical system (CPS) has emerged as a promising research paradigm, which is a convergence of control, communication, and computation [1,2,3]. A key feature of CPS is a tight integration of, and coordination between, the computational and physical elements. In fact, CPS encompasses most of man-made complex systems. In these CPS applications, it is of critical importance how to resolve the complex interactions between various computational and physical components.

One fundamental issue in CPS is how to balance the tradeoff between control performance and real-time constraints. In general, in order to improve control performance, more processor time should be devoted to control tasks, which will obviously reduce the processor usage for meeting the deadline of real-time tasks. Consequently, it is crucial how to maximize control performance while satisfying all the deadlines of real-time tasks. An illustration of a CPS application is shown in Fig. 1, where sensors/actuators, controllers, and other nodes communicate through a network.

There have been quite extensive studies on real-time scheduling and control co-design in a single-resource system, where the utilization bound has been typically used to check the schedulability of tasks. What has not been fully investigated is how to co-design scheduling and control in a multi-resource system, where real-time transactions visit multiple resources.

In this paper, we investigate the problem of real-time scheduling and control co-design in a multi-resource system. Our contributions can be summarized as follows:

(i) We formulate scheduling and control co-design in a multi-resource system as an optimization problem with an objective of maximizing a robust performance of physical control systems.
(ii) By adopting the recently-developed per-resource end-to-end response time analysis rather than the conventional per-job analysis, we show that we can enlarge the feasible region of the co-design optimization

* Corresponding author e-mail: kjp@dgist.ac.kr

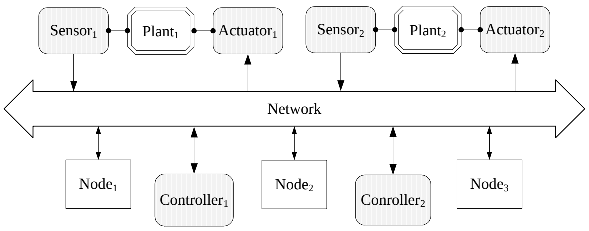


Fig. 1: An illustration of a CPS application.

problem.

(iii) By combining the control objective for robust performance and the per-resource response time analysis, we demonstrate that we can significantly improve the robustness of the overall system.

The rest of the paper is organized as follows. We provide a summary of related work in Section 2. In Section 3, we formulate control and real-time scheduling co-design as a constrained optimization problem with end-to-end response time constraints. Then, in Section 4, we first investigate a metric for control performance of the system as the objective function of the optimization problem in Section 3. We further introduce the per-resource analysis in order to derive a tight bound for the end-to-end response time, which enables us to obtain a larger feasible region compared to the conventional per-job analysis. Our simulation results are given in Section 5. Finally, our conclusion follows in Section 6.

2 Related Work

An early work on integration of real-time scheduling and control design was carried out by Seto *et al.* [4], where an optimal sampling period selection algorithm was proposed under the assumption that control performance monotonically increases as the periods decrease. In [5], RMA schedulability are formulated as an integer programming to obtain all the feasible periods of a task set, and then the optimal periods are derived by evaluating a given cost function. Overviews on scheduling and control co-design can be found in [6] and [7].

Palopoli *et al.* [8] presented a rigorous optimization approach for scheduling and control co-design in a single-resource system under the utilization bound constraint. More recently, by adopting a performance metric from the robust control theory, an optimization approach has been proposed for determining the periods of control tasks in a single-resource system [9]. Our control design follows the approach in [8] by adopting the notion of the stability radius. In the meantime, it should be noted that our co-design formulation differs from these previous studies in that we study the scheduling and control co-design in multi-resource systems with the end-to-end response time constraints.

A scheduling problem in multi-resource systems has been investigated in an optimization framework [10], where the multi-resource scheduling has been formulated as minimization of the aggregate response times of transactions under the traditional end-to-end response time constraints. A period assignment problem has been also tackled in [11], where an optimization approach has been proposed by taking into account the control delay.

From the perspective of real-time schedulability theory, the classic work of Joseph *et al.* [12] presented the worst-case response time analysis for multiple tasks on a single processor fixed-priority scheduling system. This analysis was extended by Tindell *et al.* for arbitrary deadlines [13] and distributed systems with multiple resources [14]. These studies have been further extended in many ways; reducing or eliminating the jitters [15,16], or considering precedence and timing relations among jobs [17,18,19]. However, all these studies are based on Tindell's per-job analysis in [14] and hence have a common fundamental issue of the multiple visit problem.

The delay composition theorem of [20] and [21] respectively considered the overlapped executions in pipelined distributed systems and in distributed acyclic systems to reduce the overestimation of the per-job end-to-end delay analysis. However, these approaches are not applicable to our cases where transactions visit resources multiple times in arbitrary manners.

So far, there has been little research on robust co-design of control and real-time scheduling except some preliminary work in [22], which we significantly extends here by including more thorough analysis as well as in-depth simulation results.

3 Problem Formulation

In this section, we present the notation used in our analysis and the formulation of the co-design problem as optimization with end-to-end response time constraints.

3.1 Mathematical Notation

We consider a real-time control system that consists of M resources denoted by $\mathbf{R} := \{R_1, R_2, \cdots, R_M\}$, which are either processors or communication links. Without loss of generality, we do not distinguish the type of resources under the assumption that every resource schedules its jobs based on the fixed-priority preemptive scheduling. Note that non-preemptive tasks on communication links can be dealt with by considering one message length as a blocking factor [23].

With this M-resource real-time system, we assume N periodic control *transactions* denoted by $\{\Gamma_1, \Gamma_2, \cdots, \Gamma_N\}$, where Γ_i has a higher priority than Γ_j if $i < j$. Each transaction Γ_i is composed of $|\Gamma_i|$ *tasks*, denoted by $\{\tau_{i,1}, \tau_{i,2}, \cdots, \tau_{i,|\Gamma_i|}\}$. Each task $\tau_{i,j}$, $j = 1, \cdots, |\Gamma_i|$ of Γ_i is executed on resource $r_{i,j} \in \mathbf{R}$ with the worst-case execution time of $e_{i,j}$.

The first task $\tau_{i,1}$ of transaction Γ_i is released with a period of p_i and the subsequent tasks are released at the completion times of their immediate precedent tasks. Consequently, we can represent Γ_i as follows.

$$\Gamma_i = (p_i, \{\tau_{i,1} = (r_{i,1}, e_{i,1}), \tau_{i,2} = (r_{i,2}, e_{i,2}), \cdots, \\ \tau_{i,|\Gamma_i|} = (r_{i,|\Gamma_i|}, e_{i,|\Gamma_i|})\}).$$

Here, we call one occurrence of the sequence $\tau_{i,1}, \tau_{i,2}, \cdots, \tau_{i,|\Gamma_i|}$ an *instance* of transaction Γ_i. Then, we assume that each instance of Γ_i should be completed in a period, i.e., the end-to-end deadline is equal to the period p_i. However, it should be noted that our analysis can also be applied in a straightforward manner to the case when the end-to-end deadline is shorter than the period [24].

3.2 Co-Design Problem Formulation

For a given set of N periodic transactions $\{\Gamma_1, \Gamma_2, \cdots, \Gamma_N\}$ over M resources $\{R_1, R_2, \cdots, R_M\}$, we consider the problem of how to maximize a certain control performance metric while satisfying the end-to-end schedulability constraints. In general, the problem of real-time scheduling and control co-design can be formulated as a constrained optimization problem, where the periods of control transactions are the decision variables and a control performance metric is the objective function under the end-to-end schedulability constraints as follows.

$$\begin{aligned} &\text{maximize } U(\mathbf{p}) \\ &\text{subject to } e2eRspTime_i(\mathbf{p}) \le p_i, i = 1, \cdots, N, \end{aligned} \tag{1}$$

where $\mathbf{p} = (p_1, \cdots, p_N)$, i.e., the periods of all the transactions, U is a certain metric for control performance, and $e2eRspTime_i$ is the end-to-end response time of transaction Γ_i.

In our formulation of (1), for the schedulability constraints in multi-resource systems, we introduce the end-to-end response time instead of the utilization bound typically used for the single-resource case. Though the utilization bound condition is easy to deal with in analysis because of its simplicity, it is rather a sufficient condition even in a single-resource system, and may not be efficient enough in multi-resource cases.

The control performance of each transaction will typically degrade as its period p_i increases. In addition, the overall objective function U in (1) is generally a certain increasing function of the control performance of the individual transactions. Hence, in order to maximize the objective function U, the periods of transactions, p_i's, should be decreased as much as possible. However, decreased p_i's will result in increase of the end-to-end response times of all lower-priority transactions $\Gamma_j, i < j$, because smaller p_i's will consume more processor time. Consequently, it is of critical importance how to balance this tradeoff between the control performance and the processor usage of control transactions, which is a fundamental issue in CPS.

With the optimization formulation of (1), there are two remaining issues. First, we need to determine an effective metric for control performance. Since there are various possible approaches for controller design, it is important to choose a reasonable metric that can guarantee a certain control performance. Second, it will be crucial how to calculate the end-to-end response time in an efficient manner. Since there exist many schemes for obtaining the end-to-end response time, it is required for better control performance to use a method that gives a tight bound for the end-to-end response time. In the subsequent section, we investigate these two issues in detail.

4 Systematic Approach for Solving the Co-Design Optimization Problem

In this section, we investigate the following two issues: design of the control problem and the derivation of a tight bound for the end-to-end response time.

4.1 Control Problem Formulation and Performance Metric

Our first task is how to formulate the control problem with a proper performance metric. Here, we aim to design a controller that gives *robust* performance against limitations in implementation such as imprecise actuation and truncation errors. In particular, we adopt the controller design approach in [8]. It should be noted that our overall co-design problem is quite different from that in [8] in the sense that we consider a multi-resource system with end-to-end response time constraints while the work in [8] deals with a single-resource case with the utilization bound.

For control problem formulation, consider that each control transaction Γ_i, $i = 1, \cdots, N$ controls a single input completely reachable system described by n_i linear differential equations, where n_i is called the dimension of the system. The continuous-time system dynamics with the state vector $x^{(i)} = [x_1^{(i)} \ \cdots \ x_{n_i}^{(i)}]^T$ and the control input u_i, where A^T denotes the transpose of A, can be represented in a matrix form as

$$\dot{x}^{(i)} = A_i x^{(i)} + B_i u_i, \tag{2}$$

where $A_i \in \mathbb{R}^{n_i \times n_i}, B_i \in \mathbb{R}^{n_i \times 1}$, and $i = 1, \cdots, N$. For notational simplicity, we will use the superscript (i) and subscript i only when they are strictly required.

Since there is a delay of p in each control loop, the delayed input of $u((k-1)p)$ is applied to the control system during the k-th sampling period. Hence, for the sampled discrete-time system, we introduce an additional state variable of $z(kp) = u((k-1)p)$ in order to account for the delayed input. Then, the augmented system equations sampled with the period p is given as follows from [25]:

$$\begin{bmatrix} x((k+1)p) \\ z((k+1)p) \end{bmatrix} = \Phi \begin{bmatrix} x(kp) \\ z(kp) \end{bmatrix} + \Upsilon u(kp), \tag{3}$$

where

$$\Phi = \begin{bmatrix} e^{Ap} & b\int_0^p e^{A\xi} d\xi \\ 0 & 0 \end{bmatrix}, \Upsilon = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

In the meantime, a state feedback control law for the augmented state vector $[x(kp)^T z(kp)]^T$ is given as follows.

$$u(kp) = k_x x(kp) + k_u z(kp), \tag{4}$$

where k_x and k_u are feedback gain vectors, of which the sizes are $1 \times n_i$ and 1×1, respectively. By plugging (4) into (3), the closed-loop dynamics can be derived as

$$\begin{bmatrix} x((k+1)p) \\ z((k+1)p) \end{bmatrix} = (\Phi + \Upsilon K) \begin{bmatrix} x(kp) \\ z(kp) \end{bmatrix}, \tag{5}$$

where $K = [k_x\ k_u]$. With the discrete-time equations in (5), each control transaction has a vector of feedback gains K as control parameters.

As a performance metric for control design, same as in [8], we define the stability region for control parameters K_i of transaction Γ_i as follows: Let Λ_i denote a set such that the system in (5) is asymptotically stable if and only if $K_i \in \Lambda_i$. Here, we call Λ_i the *stability region* of transaction Γ_i. Obviously, a small area of Λ_i requires a more accurate controller design because the control parameters K_i should remain in the region despite the imprecision in implementation. Hence, with a large area of Λ_i, the control system will become more robust to implementation errors.

The stability region Λ_i is generally a complex region in a multidimensional space. Hence, in order to quantify the stability region Λ_i with a single scalar value, we need an effective measure that properly represents the area of Λ_i. Here, we introduce the stability center θ_i and the stability radius μ_i as the Chebyshev center and the Chebyshev radius of Λ_i, respectively [8]. Briefly speaking, the Chebyshev center of a bounded set is defined as the center of the largest inscribed ball of the set, and the corresponding radius is called the Chebyshev radius. With the above definitions, the *stability radius* μ_i, which is actually the Chebyshev radius of Λ_i, is an effective measure of the stability region Λ_i.

With the stability radius μ_i, we can now define the performance metric $U(\mathbf{p}) = \min_{i=1,\cdots,N} \mu_i$, which is the smallest stability radius among those of the N stability regions.

Now, the overall co-design problem in (1) becomes

$$\begin{aligned} &\text{maximize} \min_{i=1,\cdots,N} \mu_i(p_i) \\ &\text{subject to } e2eRspTime_i(\mathbf{p}) \le p_i, i = 1,\cdots,N, \end{aligned} \tag{6}$$

where we explicitly show the dependencies of μ_i on p_i because the stability radius μ_i of transaction Γ_i is a function of its own period p_i.

In a qualitative sense, the objective of the co-design optimization formulation in (6) can be described as *to maximize the worst control performance* among those of N transactions. In this manner, we can improve the overall robustness of the entire system. Otherwise, if we introduce a different objective function, we could improve the control performance of some transactions at the expense of degraded performance of others. In this case, those degraded control loops will be vulnerable components from the overall system perspective. Consequently, the formulation in (6) gives a robust system performance in a holistic manner. [1]

4.2 Computation of the Stability Radius

In the case of first-order systems where $n_i = 1$, the stability region Λ can be analytically obtained as a triangle by using the Jury criterion [25]. First, the characteristic polynomial of the matrix $\Phi + \Upsilon K$ in (5) is given as

$$z^2 - \left(e^{\lambda_1 p} + k_u\right) z + e^{\lambda_1 p} k_u - b k_x I(p),$$

where λ_1 is the eigenvalue of the continuous-time system in (2) and $I(p) = \int_0^p e^{\lambda_1 \xi} d\xi = (e^{\lambda_1 p} - 1)/\lambda_1$. Then, the Jury criterion [25] gives the following inequalities for $K = [k_x\ k_u]$:

$$\begin{bmatrix} e^{\lambda_1 p} & -bI(p) \\ -(e^{\lambda_1 p} - 1) & bI(p) \\ -(e^{\lambda_1 p} + 1) & bI(p) \end{bmatrix} \begin{bmatrix} k_x \\ k_u \end{bmatrix} < \begin{bmatrix} 1 \\ 1 - e^{\lambda_1 p} \\ 1 + e^{\lambda_1 p} \end{bmatrix}. \tag{7}$$

Consequently, the stability region Λ for K can be obtained as a triangle, which is formed by three lines given in (7). In addition, from Proposition 1 in [8], the stability radius μ is given as

$$\mu = \begin{cases} \frac{\lambda_1}{e^{\lambda_1 p}(\lambda_1 + |B|) - |B|}, & \text{if } \lambda_1 > 0; \\ \frac{2\lambda_1}{e^{\lambda_1 p}(\lambda_1 + 2|B|) + \lambda_1 - 2|B|}, & \text{if } \lambda_1 < 0; \\ \frac{1}{1 + p|B|}, & \text{if } \lambda_1 = 0. \end{cases} \tag{8}$$

Note that it can be easily shown from (8) that the stability radius μ monotonically decreases with p.

[1] It is still possible to introduce a different objective function for improving the aggregate control performance rather than robustness. One possible candidate is $U(\mathbf{p}) = \sum_{i=1}^{N} \mu_i(p_i)$. A detailed treatment on the formulation with different objectives will be a subject of future work.

In general, it is formidable to compute the stability radius in higher-order systems. However, it is still possible to derive the empirical probability for the system in (5) being stable by using the randomized algorithms [26].

Let $P(\theta,\mu)$ denote the empirical probability for (5) being stable in the set of $B_\mu(\theta) = \{K \mid ||K-\theta|| \leq \mu\}$. Once μ is given, we can numerically find $\theta^*(\mu)$ that maximizes $P(\theta,\mu)$. Hence, for any given tolerance of ε, the stability radius and the stability center can be estimated as the minimum μ and the corresponding θ^* such that $P(\theta^*(\mu),\mu) \leq 1-\varepsilon$.

Here, we give a brief introduction on how to apply the randomized algorithms to the calculation of the empirical probability $P(\theta,\mu)$ for a given μ. First, draw m random samples for θ, denoted by $\theta_1,\cdots,\theta_m$. Then, for each θ_i, by drawing n samples of K in $B_\mu(\theta)$, calculate the empirical probability of $P(\theta_i,\mu)$ denoted by $P_n(\theta_i,\mu)$. Finally, we can obtain the estimate of the stability center as $\theta_{m,n} = \arg\max_{i=1,\cdots,m} P_n(\theta_i,\mu)$. Note that a detailed explanation including the selection rule for m and n with a given ε can be found in [26].

4.3 Per-Job End-to-End Response Time Analysis and Multiple Visit Problem

With the control problem formulation in the preceding sections, the remaining issue is how to derive a tight bound for the end-to-end response time. For calculation of the end-to-end response time, we may use the conventional per-job end-to-end response time analysis [14], of which a brief overview is as follows.

For task $\tau_{i,k}$ in transaction Γ_i, its per-job worst-case response time, denoted by $w_{i,k}$, is calculated by using the following recursive equation.

$$w_{i,k} = e_{i,k} + \sum_{\forall j<i} \sum_{\{a|r_{j,a}=r_{i,k}\}} \left\lceil \frac{J_{j,a}+w_{i,k}}{p_j} \right\rceil e_{j,a}, \qquad (9)$$

where $J_{j,a}$ is the worst-case release jitter of a-th task $\tau_{j,a}$ of a higher priority transaction Γ_j. Equation (9) implies that the per-job worst-case response time of task $\tau_{i,k}$ can be calculated by adding the following two terms; *(i)* its own execution time $e_{i,k}$ and *(ii)* the largest possible delay due to higher priority jobs on the same resource. Consequently, by applying (9) to all the tasks in transaction Γ_i, the worst-case end-to-end response time, $e2eRspTime_i$, can be calculated by summing up all the per-job response times as follows.

$$e2eRspTime_i = \sum_{k=1}^{|\Gamma_i|} w_{i,k}.$$

However, this per-job analysis can severely overrate the end-to-end response time when transaction Γ_i visits the same resource multiple times, which is termed the *multiple visit problem* [24]. This overvaluation of the end-to-end response time will result in a severe underestimation of the maximum schedulable region, which can significantly degrade both the scheduling and control performance.

As an illustrative example for the multiple visit problem of the per-job analysis, we consider the case in Fig. 2(a), where three Electronic Control Units (ECUs) are connected through a Controller Area Network (CAN) bus. We assume two transactions in the system as follows: A high priority transaction consists of five tasks (❶,❷,❸,❹,❺) that utilizes ECU_2, CAN, ECU_3, CAN, and ECU_1, respectively. A low priority transaction has five tasks (⑥,⑦,⑧,⑨,⑩) that utilizes ECU_1, CAN, ECU_2, CAN, and ECU_3, respectively. In this system, the conventional per-job analysis is illustrated in Fig. 2(b).

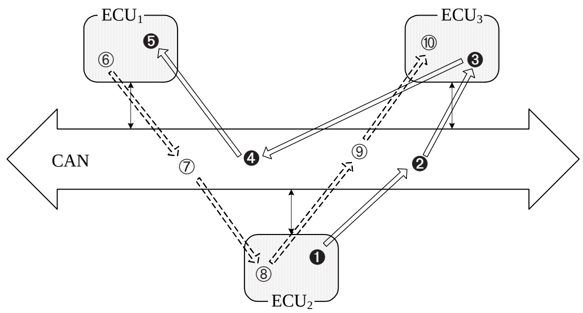
In the figure, the low priority transaction visits CAN two times with task ⑦ and task ⑨. For each visit, the per-job analysis assumes that the worst-case delay by the high-priority tasks is attributed by task ❷ and task ❹. Hence, as shown in Fig. 2(b), the execution times of tasks ❷ and ❹ in the high-priority transaction may be *double-counted* in calculation of the end-to-end response time of the low-priority transaction. Obviously, this redundant counts in the per-job analysis will result in an overestimation of the end-to-end response time, which becomes more severe as the number of the multiple visit increases. Accordingly, the conventional per-job response time analysis may conclude that the overall system is unschedulable even when the computational resources are severely underutilized.

4.4 Per-Resource End-to-End Response Time Analysis

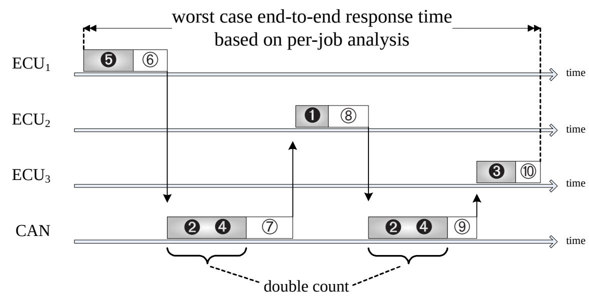
To resolve the multiple visit problem of the traditional per-job response time analysis explained in the previous section, we introduce the recently developed per-resource end-to-end response time analysis [24]. In a nutshell, the per-resource analysis calculates the total delay at each resource. Then, by summing up the total delays at every resource, the worst-case bound for the end-to-end response time can be obtained. By completely changing the viewpoint from a job to a resource, the per-resource analysis can significantly reduce the redundant counting in the per-job analysis caused by the multiple visit problem.

Figure 3 gives an illustration that compares the per-job analysis and the per-resource analysis. As shown in the figure, the per-resource analysis has no redundant counting for multiple visits, and consequently provides a much tighter bound on the end-to-end response time.

In our per-resource response time analysis, the end-to-end response time of transaction Γ_i can be calculated by summing up the times spent at all the visiting resources as



(a) Two transactions on an ECU network



(b) Per-job based analysis

Fig. 2: Illustrative example of the multiple visit problem.

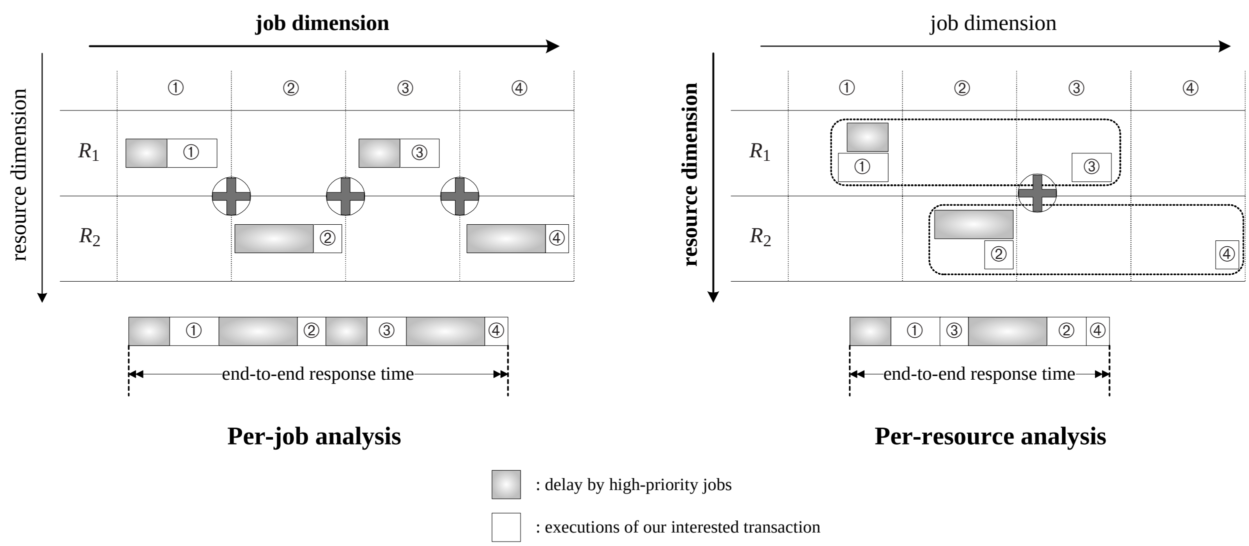


Fig. 3: Conceptual comparison of per-job analysis and per-resource analysis.

follows:

$$e2eRspTime_i = \sum_{\forall R_l \in \mathbf{R}} \left(\sum_{\{(i,k)|r_{i,k}=R_l\}} e_{i,k} + \sum_{j=1}^{i-1} TD_i^j(R_l) \right), \qquad (10)$$

where $e_{i,k}$ is the execution time of task $\tau_{i,k}$ and $TD_i^j(R_l)$ denotes the per-resource total delay, which is defined as the worst-case total delay that one instance of Γ_i experiences due to higher priority transactions Γ_j, $j = 1, \cdots, i-1$ at resource R_l.

In order to further derive the total delay $TD_i^j(R)$ in (10), we introduce a notion of the per-resource total window, denoted by $TW_i(R)$, which is defined as the time duration during which an instance of transaction Γ_i has unfinished tasks on resource R. Then, to find $TD_i^j(R)$, we introduce an iterative convergence approach, similarly as in the traditional recursive response time equation [13, 12].

Initially, we set $TD_i^j(R) = 0$ for all the transactions Γ_j, $j = 1, \cdots, i-1$ and for all the resources $R \in \{R_1, R_2, \cdots, R_M\}$. Then, we have the following iterative equation between $TW_i(R)$ and $TD_i^j(R)$:

$$TW_i(R) = \sum_{v_1 \le k \le v_m} e_{i,k} + \sum_{\forall R_l \in \mathbf{R}} \sum_{j=1}^{i-1} TD_i^j(R_l) X_{v_1}^{v_m}(R_l), \qquad (11)$$

where

$$X_{v_1}^{v_m}(R_l) = \begin{cases} 1, \text{ if any of } \{\tau_{i,v_1}, \cdots, \tau_{i,v_m}\} \text{ visits } R_l; \\ 0, \text{ otherwise.} \end{cases}$$

Once $TW_i(R_l)$ for resource R_l is given, $TD_i^j(R_l)$ can be obtained by

$$TD_i^j(R_l) = \sum_{\{a|r_{j,a}=R_l\}} \left(C_i^{j,a}(TW_i(R_l)) \times e_{j,a} \right), \quad (12)$$

where $C_i^{j,a}(TW_i(R_l))$ denotes the worst-case total number of instances of $e_{j,a}$ attributing to $TD_i^j(R_l)$ in $TW_i(R_l)$. We can calculate $C_i^{j,a}(TW_i(R_l))$ by

$$C_i^{j,a}(TW_i(R_l)) = \min\left[Z_{j,a}(TW_i(R_l)), \sum_{\{k|r_{i,k}=R_l\}} I_{j,a}(i,k) \right], \quad (13)$$

where $Z_{j,a}(TW_i(R_l)) = \lceil J_{j,a} + TW_i(R_l)/p_j \rceil$ and $I_{j,a}(i,k)$ is the largest possible number of release of task $\tau_{j,a}$ during the busy period of task $\tau_{i,k}$ which can be obtained from (9).

Consequently, by applying (11), (12), and (13) altogether in an iterative manner, we can calculate the total delay $TD_i^j(R)$, which in turn gives the end-to-end response time by (10). [2]

5 Numerical Study

In this section, we numerically study the performance of the proposed approach for scheduling and control co-design.

5.1 Simulation Setup

We consider a multi-resource system in Fig. 1. Assume that there are four transactions, denoted by $\{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\}$. Our goal is to determine the periods of transactions Γ_1 and Γ_2 while Γ_3 and Γ_4 have fixed periods. [3] In addition, we assume that the shared network in Fig. 1 is a Controller Area Network (CAN) bus, which is considered as one of the resources as already explained in Section 3.1. The visit sequences of transactions are given as follows:

$$\Gamma 1 : \{S_1, \text{CAN}, C_1, \text{CAN}, A_1\},$$
$$\Gamma 2 : \{S_2, \text{CAN}, C_2, \text{CAN}, A_2\},$$
$$\Gamma 3 : \{N_1, \text{CAN}, C_1, \text{CAN}, N_2, \text{CAN}, C_1, \text{CAN}, N_1\},$$
$$\Gamma 4 : \{N_3, \text{CAN}, C_2, \text{CAN}, N_2, \text{CAN}, C_2, \text{CAN}, N_3\},$$

[2] Detailed derivations and proofs on the per-resource end-to-end response time analysis can be found in [24].

[3] Though our co-design approach can be applied to the case of N transactions, we consider two transactions in order to effectively show the results in a geometrical manner. Note that we add two more transactions with fixed periods to make the situation complicated.

Table 1: The periods of transactions and the execution times of resources in ms.

	Period	S_i, C_i, A_i, Node$_i$	CAN
Γ_1	p_1	10	50
Γ_2	p_2	20	60
Γ_3	1500	30	70
Γ_4	3000	40	80

where S_i, C_i, A_i, $i = 1,2$ and N_i, $i = 1,2,3$ denote Sensor$_i$, Controller$_i$, Actuator$_i$, and Node$_i$ in Fig. 1, respectively. The periods of each transaction and the execution times at each resource are summarized in Table 1. The priority of a transaction is given with the rate-monotonic priority assignment. For the dynamics of Plant$_1$ and Plant$_2$, we use $\lambda_1 = 1$ and $\lambda_2 = 3$, respectively, and $B = 1$ for both plants.

Note that we can solve the optimization problem in (6) by numerically finding the maximum value of μ such that $\mathbf{p}(\mu) = (p_1(\mu), \cdots, p_N(\mu))$ satisfies the end-to-end response time constraints in (1). Here, $p_i(\mu)$, $i = 1, \cdots, N$ is the corresponding period for a given μ, which can be analytically obtained from the one-to-one relation in (8) for first-order systems. In cases of higher-order systems, we can use the bounds derived in [8].

5.2 Feasible Region of the Periods

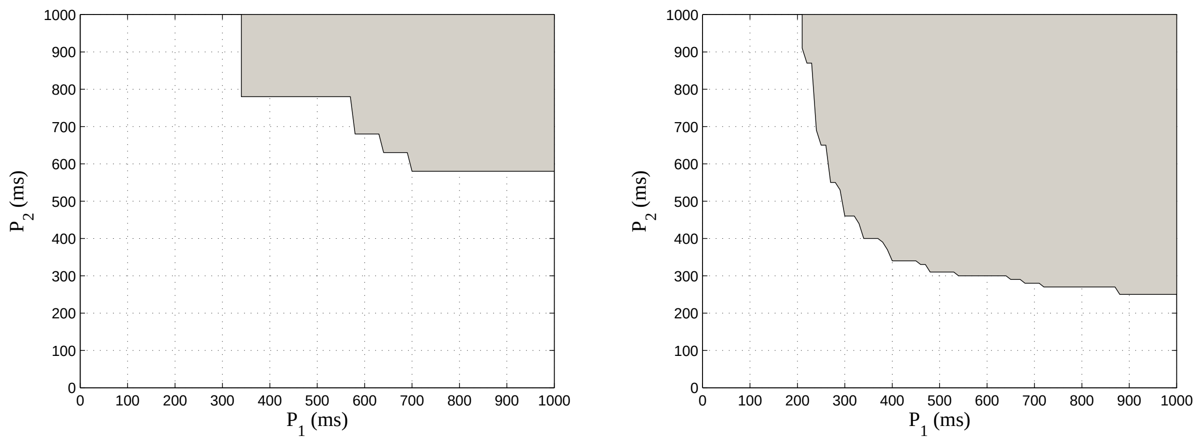
First, we compare the feasible region of the periods $\mathbf{p} = (p_1, p_2)$ by the per-resource analysis in [24] with that by the per-job analysis in [14]. As explained in Fig. 2, the traditional per-job analysis has the multiple visit problem, which overestimates the end-to-end response time. This overestimation will reduce the feasible region of $\mathbf{p} = (p_1, p_2)$ that satisfies the end-to-end response time constraints in (6).

The feasible regions of the per-job analysis and the per-resource analysis are given as the grey areas in Fig. 4. As shown in the figure, our per-resource analysis gives a significantly larger feasible region than the conventional per-job analysis. Consequently, we can confirm from Fig. 4 that the per-resource analysis gives a tighter bound for the end-to-end response time compared to the per-job analysis. Note that the non-smooth boundaries of both regions are mainly due to the ceiling operation in the response time analysis.

5.3 Comparison of the Stability Regions

Now, we look into the control performance of the system. In particular, in order to see the effect of both the control metric and the response time analysis, we introduce the following objective in (1) for comparison with our objective function: $U_{primitive}(\mathbf{p}) = -\sum_{i=1}^N p_i$. [4] Though

[4] The minus sign is used for consistency because the overall problem in (1) is maximization and $\sum_{i=1}^N p_i$ should be minimized.



(a) Feasible region by the per-job analysis

(b) Feasible region by the per-resource analysis

Fig. 4: Comparison of the feasible regions of the per-job analysis and the per-resource analysis.

$U_{primitive}$ is a natural objective function that has been often used in the literature, $U_{primitive}$ consider no plant dynamics, and hence do not differentiate control loops from the control theoretic viewpoint. On the contrary, our objective function of the minimum stability radius in (6) reflects the plant dynamics as given in (8) for the overall robustness.

With the introduction of $U_{primitive}$, we have the following four combinations for solving the optimization problem of (1): ($U_{primitive}$, per-job analysis), ($U_{primitive}$, per-resource analysis), (U_{ours}, per-job analysis), and (U_{ours}, per-resource analysis). Table 2 shows the optimal solution $\mathbf{p}^* = (p_1^*, p_2^*)$ to each combination and the corresponding stability radius $\mu(\mathbf{p}^*) = (\mu_i(p_1^*), \mu_i(p_2^*))$. Note that the underlined values are the period that gives the smallest stability radius and the corresponding stability radius between p_1^* and p_2^*.

In Table 2, if we compare the results in each column, we can notice that $U_{primitive}$ gives a smaller aggregate of the periods than U_{ours}. In fact, we can easily expect this result from the objective of each formulation. However, the smallest stability radius (underlined in Table 2) is smaller with $U_{primitive}$. This fact indicates that $U_{primitive}$ improves the control performance of one transaction at the expense of the other one, which results in a severe unbalance between the stability radius of two transactions. One interesting point in Table 2 is the fact that a smaller one between p_1^* and p_2^* does not always give a larger stability radius, which causes from the fact that the stability radius is not only a function of the period, but also of the system dynamics as given in (8).

From the robustness perspective of the whole CPS system, the vulnerable point will be the weakest component among those consisting the entire system. Hence, in order to improve the robustness of the whole system, it is required to balance the robustness of each

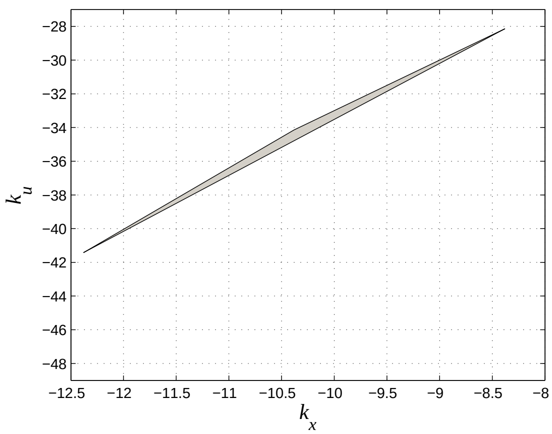
Table 2: The optimal solution $\mathbf{p}^*$ to the respective formulation and the corresponding stability radius.

	Per-job analysis	Per-resource analysis
U_{prim}	$\mathbf{p}^* = (340, \underline{780})$ $\mu(\mathbf{p}^*) = (0.5525, \underline{0.0740})$	$\mathbf{p}^* = (340, \underline{400})$ $\mu(\mathbf{p}^*) = (0.5525, \underline{0.2442})$
U_{ours}	$\mathbf{p}^* = (700, \underline{580})$ $\mu(\mathbf{p}^*) = (0.3303, \underline{0.1377})$	$\mathbf{p}^* = (650, \underline{290})$ $\mu(\mathbf{p}^*) = (0.3532, \underline{0.3510})$

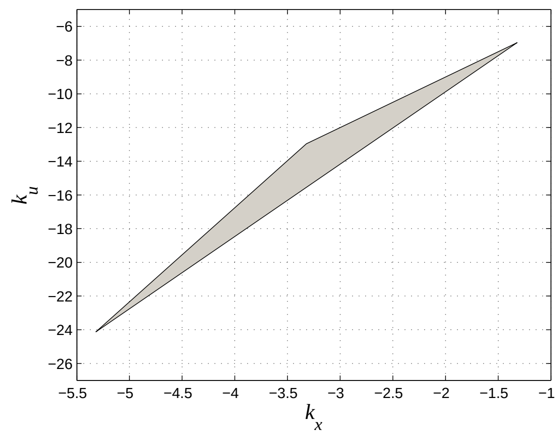
component as much as possible, which is actually done by our objective U_{ours} in (6). Figure 5 clearly shows this point in a geometrical manner. The stability region of the transaction with the smallest stability radius in Table 2 are shown in Fig. 5. As we can expect from the analysis, the stability region increases either with the per-resource analysis or with the proposed objective. In particular, by comparing Fig. 5(a) and Fig. 5(d), we can conclude that our proposed approach of the robustness objective with the per-resource analysis significantly increases the stability region, which will in turn improve the robustness of the whole CPS system.

6 Conclusion

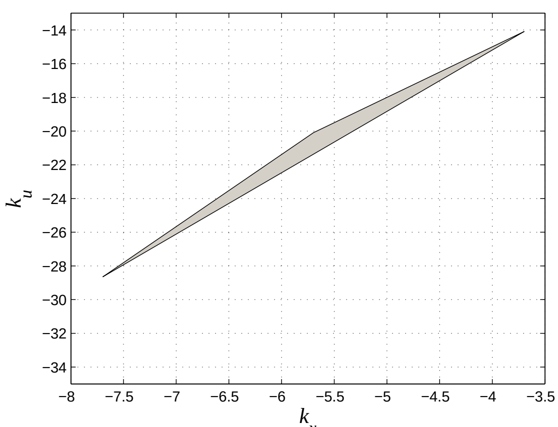
In this paper, we have investigated the problem of real-time scheduling and control co-design in a multi-resource system from the perspective of the robustness of the whole system. Our work has several distinguishing features from previous related studies. First, instead of the utilization bound as the schedulability condition, we have adopted the end-to-end response time analysis for multiple-resource systems. Second, we have investigated the control performance degradation with the conventional per-job response time analysis due to the multiple visit problem. Finally, we have shown that the adopted per-resource response time analysis, together



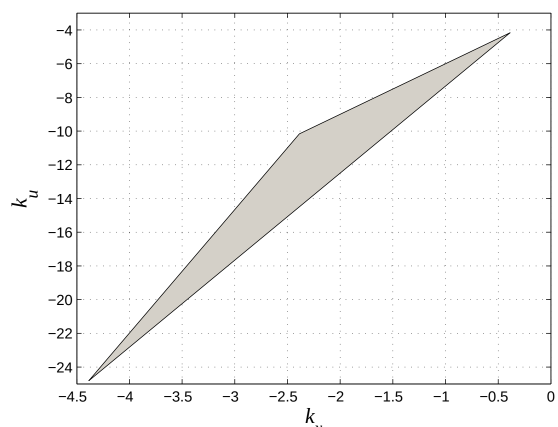
(a) Stability region Λ for the primitive objective with the conventional per-job analysis



(b) Stability region Λ for the primitive objective with the per-resource analysis



(c) Stability region Λ for our robustness objective with the conventional per-job analysis



(d) Stability region Λ for our robustness objective with the per-resource analysis

Fig. 5: Stability region Λ of the transaction with the smallest stability radius.

with an appropriately chosen metric for control performance, can significantly improve the robustness of the system .

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0101-15-0557, Resilient Cyber-Physical Systems Research).

References

[1] J. A. Stankovic, I. Lee, A. Mok, and R. Rajkumar, “Opportunities and obligations for physical computing systems,” *IEEE Computer*, vol. 38, no. 11, pp. 23–31, Nov. 2005.

[2] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, “Cyber-physical systems: A new frontier,” in *Proc. of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Jun. 2008.

[3] K.-J. Park, R. Zheng, and X. Liu, “Cyber-physical systems: Milestones and research challenges,” *Computer Communications*, vol. 36, no. 1, pp. 1–7, Dec. 2012.

[4] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin, “On task schedulability in real-time control systems,” in *Proceedings of the 17th Real-Time Systems Symposium*, Dec. 1996.

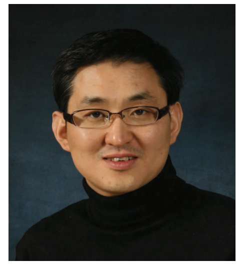
[5] D. Seto, J. P. Lehoczky, and L. Sha, “Task Period Selection and Schedulability in Real-Time Systems,” in *Proceedings of the 19th Real-Time Systems Symposium*, Dec. 1998.

[6] K.-E. Årzén, A. Cervin, J. Eker, and L. Sha, “An Introduction to Control and Scheduling Co-Design,” in

Proc. of the 39th IEEE Conference on Decision and Control, Dec. 2000.
[7] D. Simon, P. Hokayem, J. Lygeros, and E. Camacho, “State of the art in control/computing co-design,” The FeedNetback Project (http://www.feednetback.eu/), Tech. Rep., Mar. 2009.
[8] L. Palopoli, C. Pinello, A. Bicchi, and A. L. Sangiovanni-Vincentelli, “Maximizing the stability radius of a set of systems under real-time scheduling constraints,” *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1790–1795, Nov. 2005.
[9] F. Zhang, K. Szwaykowska, W. Wolf, and V. Mooney, “Task scheduling for control oriented requirements for cyber-physical systems,” in *Proc. of the 29th Real-Time Systems Symposium*, Nov.–Dec. 2008.
[10] A. Davare, Q. Zhu, M. D. Natale, C. Pinello, S. Kanajan, and A. L. Sangiovanni-Vincentelli, “Period optimization for hard real-time distributed automotive systems,” in *Proc. of the 44th Design Automation Conference*, Jun. 2007.
[11] E. Bini and A. Cervin, “Delay-aware period assignment in control systems,” in *Proc. of the 29th Real-Time Systems Symposium*, Dec. 2008.
[12] M. Joseph and P. Pandya, “Finding Response Times in a Real-Time System,” *BCS Computer J.*, vol. 29, no. 5, pp. 390–395, Oct. 1986.
[13] K. Tindell, A. Burns, and A. Wellings, “An Extendible Approach for Analyzing Fixed Priority Hard Real-Time Tasks,” *J. Real-Time Systems*, vol. 6, no. 2, pp. 133–151, Mar. 1994.
[14] K. Tindell and J. Clark, “Holistic schedulability analysis for distributed hard real-time systems,” *Microprocessing and Microprogramming - Euromicro Journal*, vol. 40, no. 2-3, pp. 117–134, Apr. 1994.
[15] J. Sun and J. Liu, “Bounding the end-to-end response times of tasks in a distributed real-time system using the direct synchronization protocol,” Department of Computer Science, University of Illinois at Urbana-Champaign, Tech. Rep. UIUCDCS-R-96-1949, Jun. 1996.
[16] J. C. Palencia, J. J. G. García, and M. G. Harbour, “Best-case analysis for improving the worst-case schedulability test for distributed hard real-time systems,” in *Proc. of the 10th Euromicro Workshop on Real-Time Systems*, 1998, pp. 35–44.
[17] J. C. Palencia and M. G. Harbour, “Schedulability analysis for tasks with static and dynamic offsets,” in *Proc. of the 19th IEEE Real-Time Systems Symp.*, Dec. 1998, pp. 26–37.
[18] J. C. Palencia and M. G. Harbour, “Exploiting precedence relations in the schedulability analysis of distributed real-time systems,” in *Proc. IEEE 20th Real-Time Systems Symp.*, Dec. 1999, pp. 328–339.
[19] R. Henia and R. Ernst, “Improved offset-analysis using multiple timing-references,” in *Proc. of the conference on Design, Automation and Test in Europe*, Mar. 2006, pp. 450–455.
[20] P. Jayachandran and T. Abdelzaher, “A Delay Composition Theorem for Real-Time Pipelines,” in *Proc. of the 19th Euromicro Conference on Real-Time Systems*, Jul. 2007.
[21] P. Jayachandran and T. Abdelzaher, “Transforming Distributed Acyclic Systems into Equivalent Uniprocessors Under Preemptive and Non-Preemptive Scheduling,” in *Proc. of the 20th Euromicro Conference on Real-Time Systems*, Jul. 2008.
[22] K.-J. Park, M.-K. Yoon, K. Kang, and C.-G. Lee, “Scheduling and control co-design under end-to-end response time constraints in cyber-physical systems,” in *Proc. of the First International Workshop on Cyber-Physical Networking Systems (CPNS 2011), in conjunction with IEEE INFOCOM 2011*, Apr. 2011.
[23] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, “Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised,” *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, Apr. 2007.
[24] M.-K. Yoon, C.-G. Lee, and J. Han, “Migrating from per-job analysis to per-resource analysis for tighter bounds of end-to-end response times,” *IEEE Transactions on Computers*, vol. 59, no. 7, pp. 933–942, Jul. 2010.
[25] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems*. Prentice Hall, 1997.
[26] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*. Springer, 2007.

Kyung-Joon Park received his B.S. and M.S. degrees from the School of Electrical Engineering and Ph.D. degree from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea. He is currently an associate professor in the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea. He was a postdoctoral research associate in the Department of Computer Science, University of Illinois at Urbana-Champaign (UIUC), IL, USA from 2006 to 2010. His current research interests include modeling and analysis of resilient cyber-physical systems. He is currently serving on the editorial boards of Wiley Transactions on Emerging Telecommunications Technologies. He has served as lead guest editor for the special issue on cyber-physical systems in Computer Communications. He is a recipient of the Gold Prize in the Samsung InsideEdge Thesis Competition.

Man-Ki Yoon is a PhD candidate at the University of Illinois at Urbana-Champaign. He received his bachelors degree in Computer Science and Engineering at Seoul National University, Korea in 2009. His research interests include secure embedded systems, multicore architecture, real-time scheduling, and machine learning. He is a recipient of Qualcomm Innovation Fellowship 2013, Qualcomm Roberto Padovani Scholarship 2014, and Intel PhD fellowship 2014.



Chang-Gun Lee received the BS, MS, and PhD degrees in computer science and engineering from Seoul National University, Korea, in 1991, 1993, and 1998, respectively. He is currently a professor in the School of Computer Science and Engineering, Seoul National University, Korea. Before he joined SNU at 2006, he was an assistant professor in the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, from 2002 to 2006, a research scientist in the Department of Computer Science, University of Illinois, Urbana-Champaign, from 2000 to 2002, and a research engineer in the Advanced Telecomm. Research Lab., LG Information and Communications, Ltd. from 1998 to 2000. His current research interests include realtime embedded systems, cyber-physical systems, ubiquitous systems, QoS management, wireless ad hoc networks, and flash memory systems. He is a member of the IEEE and the IEEE Computer Society.