

## 실시간 제어시스템의 확률적 응답시간 해석에 관한 연구

한 재 현<sup>1)</sup> · 신 민 석<sup>1)</sup> · 황 인 용<sup>1)</sup> · 선 우 명 호<sup>\*2)</sup>

한양대학교 대학원<sup>1)</sup> · 한양대학교 자동차공학과<sup>2)</sup>

### A Study on Probabilistic Response-time Analysis for Real-time Control Systems

Jaehyun Han<sup>1)</sup> · Minsuk Shin<sup>1)</sup> · Inyong Hwang<sup>1)</sup> · Myoungcho Sunwoo<sup>\*2)</sup>

<sup>1)</sup>Department of Automotive Engineering, Graduate School, Hanyang University, Seoul 133-791, Korea

<sup>2)</sup>Department of Automotive Engineering, Hanyang University, Seoul 133-791, Korea

(Received 20 December 2005 / Accepted 24 January 2006)

**Abstract :** In real-time control systems, the traditional timing analysis based on worst-case response-time (WCRT) is too conservative for the firm and soft real-time control systems, which permit the maximum utilization factor greater than one. We suggested a probabilistic analysis method possible to apply the firm and soft real-time control systems under considering dependency relationship between tasks. The proposed technique determines the deadline miss probability (DMP) of each task from computing the average response-time distribution under a fixed-priority scheduling policy. The method improves the predictable ability for the average performance and the temporal behavior of real-time control systems.

**Key words :** Probabilistic response-time analysis(확률적 응답시간 해석), Average performance(평균 성능), Firm real-time system(준경성 실시간 시스템), Soft real-time system(연성 실시간 시스템), Deadline miss probability(마감시간 초과 확률, DMP)

#### Nomenclature

$\tau$  : task  
 $\Gamma$  : job  
 $\Phi$  : initial phase  
 $T$  : period  
 $D$  : relative deadline  
 $J$  : jitter  
 $C$  : execution time  
 $I$  : invocation time  
 $\lambda$  : release time

$d$  : absolute deadline  
 $R$  : response time  
 $DMP$  : deadline miss probability  
 $V$  : job-level backlog at release-time of each job  
 $f(\cdot)$  : probability mass function, PMF

#### Subscripts

$i, j$  : task or job index

### 1. 서론

최근 강화되고 있는 배기 및 안전 규제를 만족시키고 사용자의 편의를 향상시키기 위하여, 많은 실

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

시간 제어시스템(real-time control system)이 자동차에 적용되고 있다.

실시간 제어시스템에서 시간에 따른 시스템의 거동은 시스템의 성능에 많은 영향을 준다. 최근 이런 실시간 제어시스템을 효과적으로 설계하기 위하여 시간 해석에 대한 연구가 많이 진행되었다. 스케줄 가능 여부 테스트(schedulability test)<sup>1-3)</sup>를 이용하여 주기적인 태스크들이 마감시간(deadline)을 만족할 수 있는지 여부를 판별할 수 있으며, 우선순위를 고려한 태스크들의 최악응답시간(worst-case response-time, WCRT)<sup>4-9)</sup>을 계산할 수 있다.

가장 일반적인 시간 해석 알고리즘인 최악 응답 시간 해석(worst-case response-time analysis) 알고리즘<sup>4-9)</sup>은 응답시간(response-time)이 반드시 마감 시간(deadline)을 만족해야 하는 경성 실시간 제어 시스템(hard real-time system)<sup>10)</sup>을 대상으로 하는 해석 방법이다. 그러나 이 결정론적인 시간 해석 기법을 준경성 실시간 제어시스템(firm real-time system; 응답시간이 마감시간을 만족하지 못 하여도 인명 손실이나 재산 손실 등의 결과를 초래하지 않는 실시간 제어시스템)<sup>10)</sup>이나 연성 실시간 제어시스템(soft real-time control system; 마감시간을 초과하여도 계산결과가 의미가 있으며 재난 또한 발생하지 않는 실시간 제어시스템)<sup>10)</sup>에 적용하기에는 시스템의 시간 해석 기준이 너무 엄격하다. 이러한 시스템은 마감시간을 초과하여도 오직 시스템의 성능 저하만을 초래하므로, 요구되는 일정한 기준의 시스템 성능을 만족한다면 마감시간 초과는 허용될 수 있다. 따라서 이러한 시스템에 적용할 수 있는 새로운 개념의 시간 해석 알고리즘이 필요하다.

최근에 확률통계의 관점을 적용하여 각 태스크의 가변 실행시간(variable execution time)을 해석<sup>11-17)</sup>하여 마감시간 초과 확률(deadline miss probability, DMP)을 유도할 수 있는 연구<sup>12-15)</sup>가 진행되었다. 그러나 가변 실행시간의 원인 중 하나인 각 태스크간의 의존성(dependency)을 고려하지 않은 채 진행되어 한계가 있다.

이 연구에서는 고정 우선순위 스케줄링 방식(static priority scheduling method)을 가지는 준경성

및 연성 실시간 제어시스템을 대상으로 기존의 확률적 응답시간 해석 알고리즘을 연구하고, 태스크간의 의존성을 고려한 새로운 알고리즘을 제시하며 비교·검증한다. 또한 각 태스크의 마감시간 초과 확률을 이용하여 실시간 제어시스템의 평균 성능을 평가한다.

## 2. 시스템 모델

### 2.1 태스크 모델

실시간 제어시스템은  $N$  개의 주기적인 태스크들의 집합  $S = \{\tau_1, \tau_2, \dots, \tau_N\}$  으로 구성된다. 이러한 태스크의 주기적인 수행을 작업(job)이라 하면, 각 태스크는 연속적인 작업들의 집합  $\tau_i = \{\Gamma_{i,1}, \Gamma_{i,2}, \dots\}$  으로 구성된다. 여기서  $\tau_i$  는  $i$  번째 태스크를,  $\Gamma_{i,j}$  는  $\tau_i$  의  $j$  번째 작업을 나타낸다. 각 태스크  $\tau_i$  의 모델은 다섯 개의 속성인  $(\Phi_i, T_i, D_i, J_i, C_i)$  으로 나타낸다. 여기서  $\Phi_i$  는  $\tau_i$  의 최초 호출시간(initial phase),  $T_i$  는 주기(period),  $D_i$  는 상대 마감시간(relative deadline),  $J_i$  는 지터(release jitter),  $C_i$  는 실행시간(execution time)을 나타낸다.

$\tau_i$  의 첫 번째 작업  $\Gamma_{i,1}$  은  $\Phi_i$  에서 호출되며,  $j$  번째 작업  $\Gamma_{i,j}$  는 호출시간(invocation time)  $I_{i,j}$  시점에서 호출된다. 여기서  $\tau_i$  는 주기  $T_i$  를 가지므로 각 작업의 호출시간( $I_{i,j} = \Phi_i + (j-1)T_i$ )은 최초 시기와 주기로 결정된다.  $\Gamma_{i,j}$  의 실행 시작시간  $\lambda_{i,j}$  (release time)은 지터를 고려하여  $I_{i,j} + J_i$  로 나타낼 수 있으며, 높은 우선순위를 가지는 태스크에 의한 선점(preemption)이 없을 경우  $\Gamma_{i,j}$  가 연속적으로 수행된다.

$\Gamma_{i,j}$  의 절대 마감시간(absolute deadline)과 응답시간은 각각  $d_{i,j} = I_{i,j} + D_i$  와  $R_{i,j}$  으로 나타내며,  $\tau_i$  의 응답시간은  $R_i$  으로 나타낸다. Fig. 1은 이 태스크 모델을 나타낸 것이다.

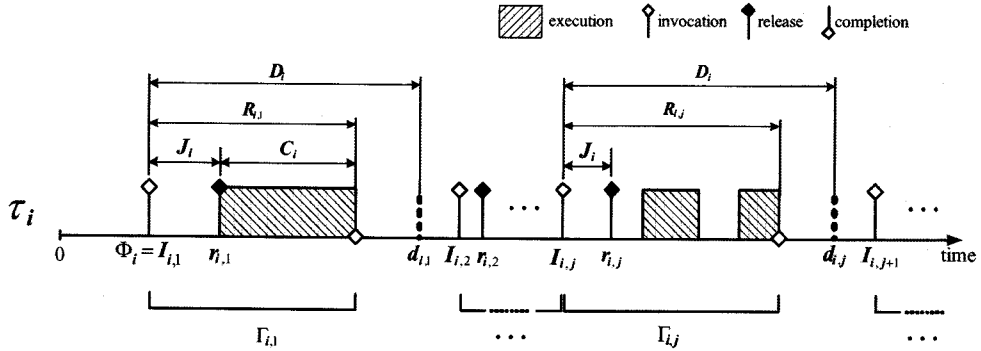


Fig. 1 Task model

2.2 확률적 시간 모델

확률적 해석을 위하여  $J_i, C_i, R_{i,j}, R_i$  를 이산확률변수(discrete random variable)라 하고, 각각의 확률 질량함수 PMF(probability mass function)를  $f_{J_i}(\cdot), f_{C_i}(\cdot), f_{R_{i,j}}(\cdot), f_{R_i}(\cdot)$  으로 나타낸다. 여기서 각각은  $f_{J_i}(j)=P\{J_i=j\}, f_{C_i}(c)=P\{C_i=c\}, f_{R_{i,j}}(r)=P\{R_{i,j}=r\}, f_{R_i}(r)=P\{R_i=r\}$  인 확률을 나타내며, Fig. 2는  $f_{C_i}(\cdot)$  의 예를 보여주고 있다.

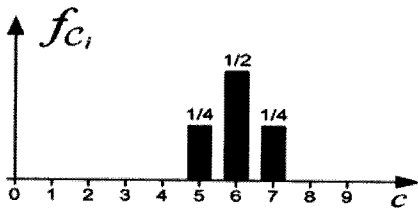


Fig. 2 Example of a task execution time PMF

결과적으로  $\tau_i$  에 대한 마감시간 초과 확률은 식 (1)과 같이  $f_{R_i}(\cdot)$  를 이용하여 결정한다.

$$DMP_i = P\{R_i > D_i\} = 1 - P\{R_i \leq D_i\} \quad (1)$$

3. 확률적 해석

3.1 의존성을 고려하지 않은 해석<sup>[12-15]</sup>

3.1.1 분석적 해석

태스크  $\tau_i$  의  $f_{R_i}(\cdot)$  는 n 개의 하이퍼주기(hyper-period 시스템 내 모든 태스크 주기의 최소 공배수)

내에 속해있는 모든 작업  $\Gamma_{i,j}$  에 대한  $f_{R_{i,j}}(\cdot)$  의 평균으로 나타낸다. 즉,

$$f_{R_i}(r) = \frac{1}{n \cdot m_i} \sum_{k=1}^n \sum_{j=1}^{m_i} f_{R_{i,j}}(r) \quad (2)$$

여기서  $m_i=T/T_i$  는 길이  $T$  인 하이퍼주기 내에 실행 시작하는 모든 작업  $\Gamma_{i,j}$  의 개수이다.

$R_{i,j}$  는 우선순위가 높은 작업들에 의한 선점(preemption)을 고려해야 하므로 식 (3)과 같이 표현될 수 있다.

$$R_{i,j} = V_{i,j} + C_i + \sum_{\Gamma_{k,j} \in H} C_k \quad (3)$$

여기서 확률변수  $V_{i,j}$  는  $\Gamma_{i,j}$  의 실행 시작시간에서  $\Gamma_{i,j}$  보다 우선순위가 높거나 같은 작업들의 미처리분(job-level backlog)이며  $H$  는  $\Gamma_{i,j}$  를 선점하는 모든 작업들을 실행 시작시간의 순서대로 배열한 집합이다. 따라서  $f_{R_{i,j}}(\cdot)$  는  $V_{i,j}$  의 PMF인  $f_{V_{i,j}}(\cdot)$  와  $f_{C_i}(\cdot)$  그리고  $f_{C_k}(\cdot)$  로 부터 얻을 수 있다.

$\Gamma'_k$  를  $H$  내의 k번째의 작업이라 하고,  $R_{i,j}^{\{k\}}$  를  $\Gamma_{i,j}$  를 선점하는 모든 작업들  $\Gamma'_1, \Gamma'_2, \dots, \Gamma'_k$  를 순서대로( $\lambda'_1 \leq \lambda'_2 \leq \dots \leq \lambda'_k$ ) 고려한 응답시간이라 한다. 또한  $f^k(\cdot)$  를 식 (4)와 같이 특정 시간구간 I에서만 값을 가지는 PMF  $f^1(\cdot)$  의 일부분이라고 정의하면,  $f_{R_{i,j}}(\cdot)$  는 [14]에서 제안한 분할(splitting) · 콘볼루

션(convolution) · 결합(merging) 연산과 수축(shrinking) 연산에 의하여 식 (5)와 (6)에 의해 계산될 수 있다.

$$f^A(r) = \begin{cases} f(r) & \text{if } r \in I \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$f_{R_{i,j}^{\infty}}(r) = (f_{V_{i,j}}^{\lambda_{i,j}, \infty}) \otimes f_{C_i}(r) \quad (5)$$

$$f_{R_{i,j}^{k,b}}(r) = f_{R_{i,j}^{k,b}}^{\lambda_{i,j}, \lambda_{k1}}(r) + (f_{R_{i,j}^{k,b}}^{\lambda_{i,j}, \infty}) \otimes f_{C_i}(r) \quad (6)$$

여기서  $f_{V_{i,j}}^{\lambda_{i,j}, \infty}(r) = f_{V_{i,j}}(r)$  이다.

두 개의 태스크로 이루어진 간단한 예를 살펴보자. 각 태스크의 속성은 Table 1과 같다.

Table 1 Parameters of the task

$\tau_i$	$\Phi_i$	$T_i$	$D_i$	$J_i$	$C_i$	$U_i^-$	$\bar{U}_i$	$U_i^+$
$\tau_1$	0	30	30	$f_{j_1}(j) = 0, f_{C_1}(c) = \frac{1}{19}$ for all $j$ for $1 \leq c \leq 19$		0.033	0.333	0.633
$\tau_2$	0	40	40	$f_{j_2}(j) = 0, f_{C_2}(c) = \frac{1}{29}$ for all $j$ for $1 \leq c \leq 29$		0.025	0.375	0.725
Total						0.058	0.708	1.358

Table 1에서  $U_i^-, \bar{U}_i, U_i^+$  는 최소, 평균, 최대 이용률(minimum, average, maximum utilization factor)을 나타낸다. 시스템의 최대 이용률은 1.358이고 평균 이용률은 0.708이므로, 여러 작업들 중 일부는 마감

시간을 만족하지 못할 것임을 알 수 있다.

시스템이 고정 우선순위 스케줄링 알고리즘인 RMS(rate monotonic scheduling) 알고리즘을 따른다고 가정하면  $\tau_1$ 의 최대 이용률은 1보다 작으므로,  $\tau_2$ 보다 우선순위가 높은  $\tau_1$ 의 작업들은 모두 마감 시간을 만족한다. 따라서 마감시간을 만족하지 않는 작업들은  $\tau_2$ 에 존재하므로  $\tau_2$ 에 대해서만 해석하기로 한다. 결과적으로 10개의 하이퍼주기를 고려한 PMF와 PDF(probability distribution function)은 Fig. 3과 같다.

Fig. 3의 (a)와 같이  $\Gamma_{2j}$ 는  $\tau_1$ 의 작업들에 의해 선점되므로  $T_1$ 과  $T_2$ 의 차이인 시간 10을 주기로 확률의 많은 변화를 보이며,  $\tau_2$ 의 마감시간 초과 확률  $DMP_2$ 는 Fig. 3의 (b)에서 보듯이 식 (7)과 같다.

$$DMP_2 = P\{R_2 > 40\} = 1 - P\{R_2 \leq 40\} = 0.16898 \quad (7)$$

### 3.1.2 시뮬레이션

이 절에서는 앞에서 분석적으로 해석한 예를 시뮬레이션 방법을 사용하여 해석하고, 앞의 결과와 비교하겠다.

시뮬레이션은 시간 축을 따라 실행 시작하는 작업들의 응답시간을 측정하여, 전체에 대한 해당 응답시간의 비율을 계산하여 나타내었다. 각 작업의 실행시간은 시뮬레이션 시행 전에 Table 1에 표기된

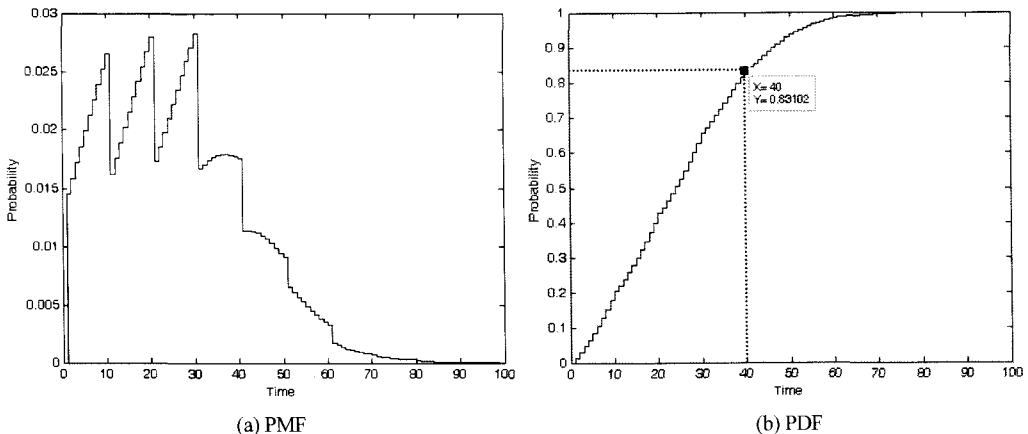


Fig. 3 Average response-time PMF and PDF of  $\tau_2$

확률을 적용하여 임의로 채택하였다.

시뮬레이션은 앞 절에서와 같이 10개의 하이퍼주기에 대한  $\tau_2$ 의 작업, 30개의 대하여 수행하였으며, 정확도를 높이기 위하여 총 2000회를 시행한 후 평균하였다.

Fig. 4는 시뮬레이션에 사용된 시간과 우선순위에 따른 태스크의 상태 천이를 나타내며, stuff는 미리 채택된 각 작업이 수행해야 하는 실행시간을 나타낸다.

Fig. 5와 같이 시뮬레이션의 결과와 분석적 해석 방법은 유사한 결과를 나타낸다.

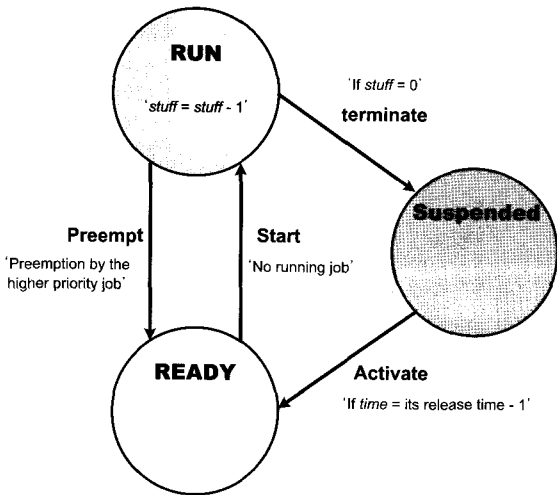


Fig. 4 State transition

### 3.2 의존성을 고려한 해석

앞 장에서는 한 태스크가 우선 순위가 높은 다른 태스크에 의해 선점되었을 경우와 이전 작업의 미처리분이 남았을 경우, 두 태스크간의 의존성을 무시하고 독립적이라 가정할 채, 콘볼루션 연산에 의해 태스크의 응답시간을 계산하였다. 그러나 실제 태스크의 실행시간에 많은 영향을 주는 for, while 반복문에 의한 실행 회수와 if, switch 조건문에 의한 수행 코드의 길이 등에 따른 연산 시간은 이전의 수행 결과나 다른 태스크의 실행에 영향을 받는다.

이 장에서는 태스크 간의 의존성을 고려한 해석 방법을 제시하겠다.

#### 3.2.1 의존성 모델

해석에 필요한 가정을 다음과 같다.

- 1) 한 작업의 실행시간은 작업이 실제 수행되기 바로 이전에 수행이 끝난 각 태스크들의 작업에 의해서만 영향을 받는다.
- 2) 한 태스크에서 연속된 작업에 대한 실행시간의 확률적 의존성은 주어진다.
- 3) 연속된 두 태스크에 대한 실행시간의 확률적 의존성은 주어진다.

의존성 모델은 기존의 태스크 모델이 가지는 속성  $(\Phi_i, T_i, D_i, J_i, C_i)$ 에 실행시간의 확률적 의존성을 추가한 형태를 지니며, 이 확률적 의존성은 결합 확률 질량함수 joint PMF(joint prob-ability mass function)

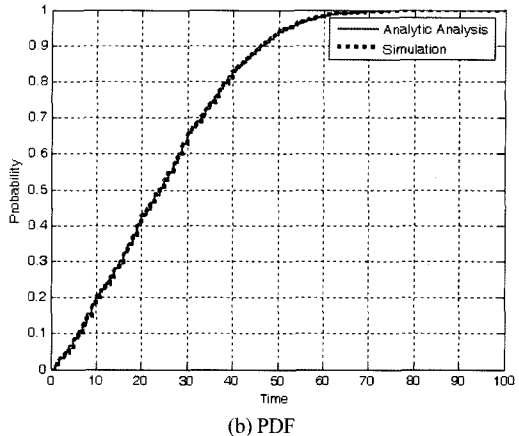
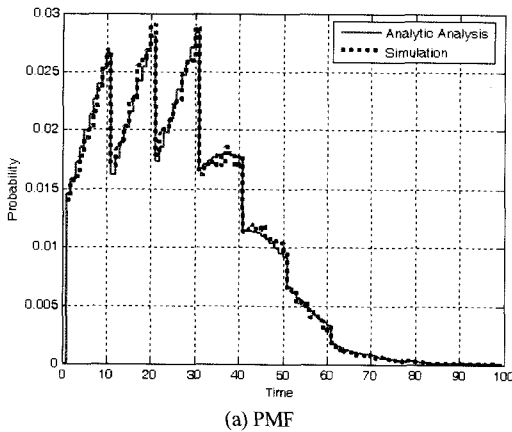


Fig. 5 Comparison between analytic analysis and simulation for the average response-time PMF and PDF of  $\tau_2$

로 나타낸다. 실행시간 이산 확률변수  $G_i, G_j, \dots, G_n$ 에 대한 결합 확률 질량함수  $f_{C_i, G_j, \dots, G_n}(c_i, c_j, \dots, c_n)$ 는 식 (8)과 같이 정의된다.

$$f_{C_i, G_j, \dots, G_n}(c_i, c_j, \dots, c_n) = P\{C_i=c_i, G_j=c_j, \dots, G_n=c_n\} \quad (8)$$

결합 확률 질량함수  $f_{C_i, G_j, \dots, G_n}(c_i, c_j, \dots, c_n)$ 는 다음의 일반적인 특성을 가지고 있다.

- ①  $0 \leq f_{C_i, G_j, \dots, G_n}(c_i, c_j, \dots, c_n) \leq 1$
- ②  $\sum_{c_i} \dots \sum_{c_n} f_{C_i, G_j, \dots, G_n}(c_i, c_j, \dots, c_n) = 1$
- ③  $P\{(C_i, G_j, \dots, G_n) \in A\} = \sum_{(c_i, c_j, \dots, c_n) \in A} f_{C_i, G_j, \dots, G_n}(c_i, c_j, \dots, c_n)$

특성 ③의 A는  $G_i, G_j, \dots, G_n$ 이 가질 수 있는 값의 집합을 나타낸다.

3.2.2 분석적 해석

각 태스크의 실행시간에 대한 PMF는 결합 확률 질량함수로부터 주변 확률 질량함수(marginal probability mass function, marginal PMF)를 이용하여 식 (9)와 같이 얻을 수 있다.

$$f_{C_i}(c_i) = P\{C_i=c_i\} = \sum_{c_j, \dots, c_k} f_{C_i, G_j, \dots, G_k}(c_i, c_j, \dots, c_k) \quad (9)$$

Joint PMF를 이용한 연산기호를  $\boxtimes$ 라 하고, 앞의 식 (5), (6)의 콘볼루션 연산을 joint PMF에 의한 연산으로 식 (10), (11)과 같이 대체한다.

$$f_{R_{i,j}^{\otimes}}(r) = (f_{V_{i,j}^{\lambda}} \boxtimes f_{C_i})(r) \quad (10)$$

$$f_{R_{i,j}^{\lambda}}(r) = f_{R_{i,j}^{\lambda, \lambda}}(r) + (f_{R_{i,j}^{\lambda, \infty}} \boxtimes f_{C_k})(r) \quad (11)$$

$\Gamma_{i,j}$ 의 실행시간에 영향을 미치는 작업들의 집합을  $C_{i,j}$ 라 하고, joint PMF에 의한 연산은 식 (12)과 같이 정의하며, 매번  $\Gamma_{i,j}$ 가 선점될 때 새로운 joint PMF를 구하여 연산을 수행한다.

$$(f_R^{\lambda} \boxtimes f_{C_k})(r) = \sum_{c+s=r} \sum_{\substack{s \in I, \\ c \in C_k}} \left( \frac{f_R^{\lambda}(s) f_{C_k, S_1}(c, s)}{\sum_{c \in C_k} f_{C_k, S_1}(c, s)} \right) \quad (12)$$

여기서 구간 I는  $[t_i, t_f)$  또는  $(t_i, t_f)$  이고, 확률변수  $S_I$ 는 I 이전에 실행 시작되었고  $C_k$  속성을 가진 작업을 제외한 모든 작업들에 대한 실행시간의 합이며, 미리 주어지지 않은 joint PMF  $f_{C_k, S_1}(c, s)$ 는 식 (13)과 같이 조건부 확률(conditional probability)에 의하여 얻을 수 있다.

$$f_{C_k, S_1}(c, s) = \begin{cases} P(C_k=c, S_1=s | S_1'=s') \cdot P(S_1'=s') & \text{if } S_1 \neq 0 \\ f_{C_k}(c) & \text{if } S_1 = 0 \end{cases} \quad (13)$$

여기서  $S_1'$ 를  $S_I$  연산에 사용된 작업들 중에서  $C_{i,j}$ 에 속하지 않는 작업의 실행시간으로 나타내면  $f_{C_k, S_1}(c, s)$ 는 반복적인 연산으로 얻어질 수가 있다. 콘볼루션 연산과 joint PMF를 이용한 연산의 차이는 Table 2의 예로 쉽게 알 수 있다. 먼저  $\tau_1$ 과  $\tau_2$ 에 대한 실행시간 PMF와 이들을 단순히 콘볼루션 연산한 결과는 Fig. 6과 같다.

Table 2 Parameters of the task in example 1

$\tau_i$	$\Phi_i$	$T_i$	$D_i$	$J_i$	$C_i$	$u_i^-$	$\bar{u}_i$	$u_i^+$
$\tau_1$	0	3	3	$f_{j_1}(j) = 0,$ for all j	$f_{C_1}(c) = \frac{1}{2},$ for $1 \leq c \leq 2$	0.333	0.5	0.667
$\tau_2$	0	4	4	$f_{j_2}(j) = 0,$ for all j	$f_{C_2}(c) = \frac{1}{2},$ for $1 \leq c \leq 2$	0.25	0.375	0.5
Total						0.583	0.875	1.167

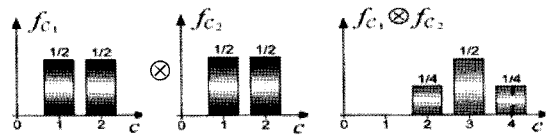


Fig. 6 Convolution between  $f_{C_1}$  and  $f_{C_2}$

그러나 Table 3과 같은 두 태스크에 대한 실행시간의 확률적 의존성을 안다면 식 (14)를 이용하여 Fig. 7과 같이 보다 정확한 연산을 수행할 수 있다.

Table 3 Joint PMF between  $C_1$  and  $C_2$

$C_1 \backslash C_2$	1	2	$f_{C_2}(\cdot)$
1	$\frac{2}{6}$	$\frac{1}{6}$	$\frac{1}{2}$
2	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{1}{2}$
$f_{C_1}(\cdot)$	$\frac{1}{2}$	$\frac{1}{2}$	

$$P\{C_i+C_j=c\} = \sum_{c_i+c_j=c} f_{C_i C_j}(c_i, c_j) \quad (14)$$

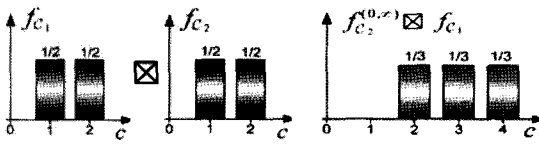


Fig. 7 Joint PMF operation between  $f_{C_1}$  and  $f_{C_2}$

다음은 Table 2의 두 태스크와 연이은 작업 모두 독립적이라고 가정한 후, 기존의 콘볼루션에 의한 연산과의 차이를 살펴보겠다. 태스크 모두 독립적 이므로 해석에 필요한 joint PMF는 Table 4, 5와 같으며, 이는 식 (15)와 (16)에 기인한다.

$$P\{C_i=c_i, C_j=c_j\} = f_{C_i}(c_i) \cdot f_{C_j}(c_j) \quad (15)$$

$$P\{C_i=c_i, C_i+C_j=c_i+c_j\} = f_{C_i}(c_i) \cdot f_{C_i+C_j}(c_i+c_j) \quad (16)$$

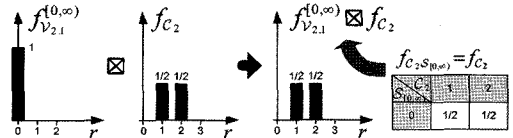
Table 4 Joint PMF between  $C_1$  and  $C_2$

$C_1 \backslash C_2$	1	2	$f_{C_2}(\cdot)$
1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$
2	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$
$f_{C_1}(\cdot)$	$\frac{1}{2}$	$\frac{1}{2}$	

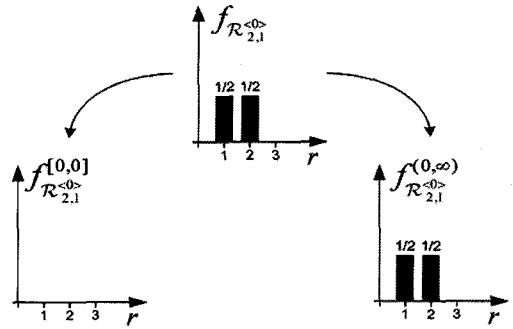
Table 5 Joint PMF between  $C_i$  and  $C_i+C_j$  ((i,j)=(1,2) and (2,1))

$C_i+C_j \backslash C_i$	1	2	$f_{C_i+C_j}(\cdot)$
2	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$
3	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$
4	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$
$f_{C_i}(\cdot)$	$\frac{1}{2}$	$\frac{1}{2}$	

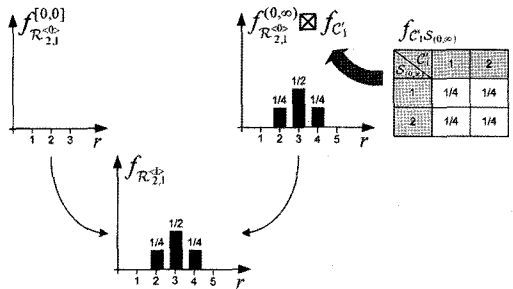
해석에 고려된  $\Gamma_{2,j}$ 는 총 30개이며 해석 과정은 Fig. 8과 같이 나타나고, 결과는 Fig. 9과 식 (17)과 같다.



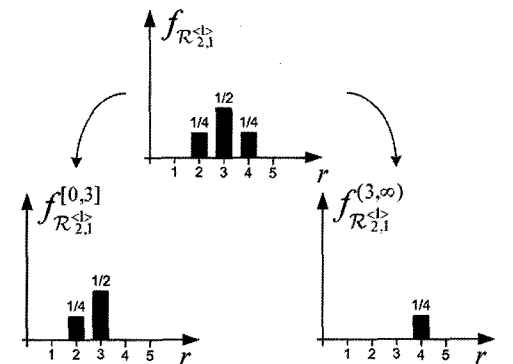
(a) Joint PMF operation between  $V_{2,1}$  and  $f_{C_2}$



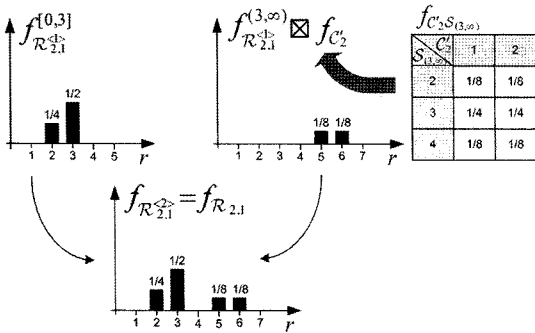
(b) Splitting operation because of the preemption of  $\Gamma'_1$



(c) Joint PMF operation and merging operation



(d) Splitting operation because of the preemption of  $\Gamma'_2$



(c) Joint PMF operation and merging operation

Fig. 8 Process of stochastic response-time analysis with joint PMF operation

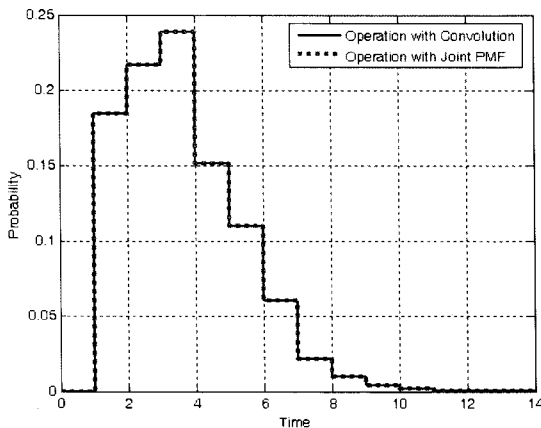
$$DMP_2 = P\{R_2 > 4\} = 1 - P\{R_2 \leq 4\} = 0.20775 \quad (17)$$

Fig. 9와 같이 콘볼루션 연산과 joint PMF에 의한 연산의 결과가 같으므로, joint PMF를 이용한 알고리즘은 콘볼루션 연산을 대체할 수 있다.

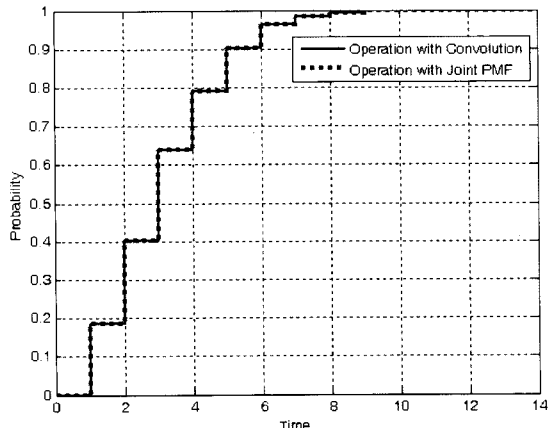
다음은 각 태스크들이 모두 독립적이지 않고 Table 3, 6과 같은 joint PMF를 가진다고 가정하면 Fig. 10과 식 (18)과 같은 결과를 가진다. 마찬가지로 해석된  $\Gamma_{2,j}$ 의 개수는 30개이다.

$$DMP_2 = P\{R_2 > 4\} = 1 - P\{R_2 \leq 4\} = 0.17329 \quad (18)$$

Fig. 10에서와 같이 의존성을 고려하지 않은 채 단순히 각 태스크의 실행시간 PMF를 콘볼루션하여 얻은 결과와 의존성을 고려한 해석은 상이하며,



(a) PMF



(b) PDF

Fig. 9 Comparison of convolution and joint PMF in PMF and PDF of  $R_2$  under independency between tasks

Table 5 Joint PMF between  $C_i$  and  $C_i + C_j$  ((i,j) = (1,2) and (2,1))

$C_i + C_j \backslash C_i$	1	2	$f_{C_i + C_j}(\cdot)$
2	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{1}{3}$
3	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{3}$
4	$\frac{1}{9}$	$\frac{2}{9}$	$\frac{1}{3}$
$f_{C_i}(\cdot)$	$\frac{1}{2}$	$\frac{1}{2}$	

16.6%의 DMP2 오차를 보인다. 따라서 태스크간의 의존적 실행시간을 추정할 수 있을 경우에 이와 같은 해석을 반복한다면, 실제 시스템을 설계할 때 태스크의 형태와 주기 등을 효과적으로 결정할 수 있다. 이는 실시간 제어시스템을 효과적으로 설계할 수 있으며, 개발기간과 개발비용을 줄일 수 있는 이점을 가져오게 된다.

#### 4. 결론

- 1) 이 연구에서는 자동차에 많이 적용되고 있는 실시간 제어 시스템에 대하여 태스크간의 의존성을 고려한 확률적 응답시간 해석방법을 제시하였다. 제안된 방법은 태스크에 대한 실행시간의 확률적 의존성을 결합 확률 질량함수 형태로 나타내어, 시스템의 평균 성능의 지표가 될 수 있는



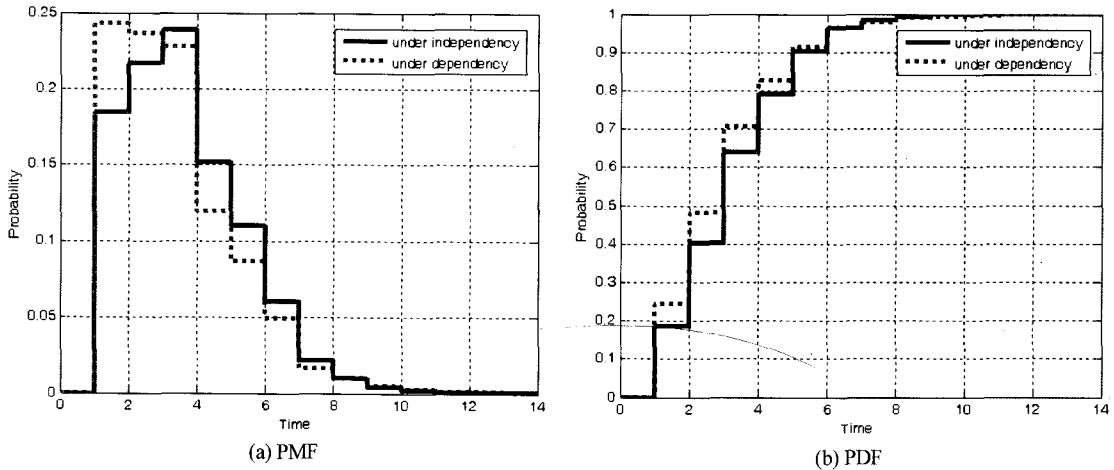


Fig. 10 Comparison of convolution and joint PMF in PMF and PDF of under dependency between tasks

각 태스크의 정확한 마감시간 초과확률을 얻어 낼 수 있게 하였다.

- 2) 향후 고정 우선순위 스케줄링 방식 외의 시스템에 대한 해석 방법을 연구하고, 태스크간의 실행 시간 관계뿐만 아니라 태스크간의 선행관계 (precedence relationship) 등을 고려한 해석으로 확장할 것이다. 최종적으로 차량의 실시간 제어 시스템으로 적용하여, 시스템의 확률적 평균성능을 도출할 것이다.

### 후 기

이 연구는 과학기술부의 국가지정연구실사업 (NRL)의 지원에 의하여 수행되었다.

### References

- 1) C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multi-programming in a Hard-real-time Environment," *Journal of the ACM*, Vol. 20, No.1, pp.46-61, 1973.
- 2) J. P. Lehoczky, L. Sha and D. Y. Ding, "The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behavior," In *Proceedings of the 10th IEEE Real-Time Systems Symposium*, pp.166-171, 1989.
- 3) K. Tindell, "Adding Time-offsets to Scheduling Analysis," Technical Report, YCS221,

Department of Computer Science, University of York, 1994.

- 4) M. Joseph and P. Pandya, "Finding Response Times in a Real-time System," *BCS Computer Journal*, Vol.29, No.5, pp.390-395, 1986.
- 5) K. Tindell, A. Burns and A. J. Wellings, "An Extendible Approach for Analyzing Fixed Priority Hard Real-time Tasks," *Real-Time Systems*, Vol.6, No.2, pp.133-151, 1994.
- 6) K. Tindell and J. Clark, "Holistic Scheduling Analysis for Distributed Hard Real-time Systems," *Microprocessing and Microprogramming*, Vol.40, pp.117-134, 1994.
- 7) K. Tindell, "Real Time System by Fixed Priority Scheduling," Ph. D. Dissertation, Department of Computer Science, University of York, 1994.
- 8) M. Shin, W. Lee and M. Sunwoo, "Holistic Scheduling Analysis of a CAN based Body Network System," *Transactions of KSAE*, Vol.10, No.5, pp.114-120, 2002.
- 9) J. Choi, M. Shin, M. Sunwoo and S. Han, "Timing Analysis of Distributed Real-time Control System using Response-time Analysis Tool," *Transactions of KSAE*, Vol.13, No.1, pp.194-203, 2005.
- 10) H. Kopetz, "Real-time Systems: Design Principles for Distributed Embedded Applications," Kluwer Academic Publishers, Boston/

Dordrecht/London.

- 11) T.-S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.-C. Wu and J. W.-S. Jiu, "Probabilistic Performance Guarantee for Real-time Tasks with Varying Computation Times," In Proceedings of the Real-Time Technology and Applications Symposium, IEEE, pp.164-173, 1995.
- 12) M. K. Gardner and J. W. S. Liu, "Analyzing Stochastic Fixed-priority Real-time Systems," In Proceedings of the 5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pp.44-58, 1999.
- 13) M. K. Gardner, "Probabilistic Analysis and Scheduling of Critical Soft Real-time Systems," Ph. D. Dissertation, Department of Computer Science, University of Illinois, 1999.
- 14) J. L. Diaz, D. F. Garcia, K. Kim, C.-G. Lee, L. L. Bello, J. M. Lopez, S. L. Min, O. Mirabella, "Stochastic Analysis of Periodic Real-time Systems," Proceedings of the 23rd IEEE Real-Time Systems Symposium, IEEE, pp.289-300, 2002.
- 15) K. H. Kim, "Analysis and Synthesis of Probabilistic Real-Time Systems," Ph. D. Dissertation, Department of Electrical Engineering and Computer Science, University of Seoul, 2004.
- 16) G. Bernat, A. Colin and S. M. Petters, "WCET Analysis of Probabilistic Hard Real-time Systems," Proceedings of the 23rd IEEE Real-Time Systems Symposium, IEEE, pp.279-288, 2002.
- 17) X. S. Hu, T. Zhou and E. H.-M. Sha, "Estimating Probabilistic Timing Performance for Real-time Embedded Systems," IEEE Transactions on VLSI Systems, Vol.9, No.6, pp.833-844, 2001.