

# 작업별 중요도 모드를 적용한 혼합 중요도 스케줄링에서 확률적 성능 평가 기법

## Probabilistic Performance Evaluation Technique for Mixed-criticality Scheduling with Task-level Criticality-mode

이재우(Jaewoo Lee)\*

### 초 록

혼합 중요도 시스템은 중요도가 다른 컴포넌트의 조합으로 이루어져 있다. 최근 자동차 시스템과 항공기 시스템에서 사용되는 ISO 26262와 DO-178B 표준에서는 컴포넌트를 중요도에 따라 분류하고 있다. 기존 혼합 중요도 시스템 연구에서는 시스템 모드를 통해서, 효율적이면서 안전한 스케줄링을 추구했다. 이러한 연구의 단점은 고중요도 모드에서 저중요도 작업의 성능 저하이다. 이러한 문제를 개선하고자 작업별 중요도 모드를 도입하여 저중요도 작업의 성능을 개선하고 확률적 성능 지표를 설계했다. 시뮬레이션을 통해서 기존 연구 대비 성능 향상 효과를 보였다.

### ABSTRACT

Mixed-criticality systems consist of components with different criticality. Recently, components are categorized depending on criticality by ISO 26262 standard and DO-178B standard in automotive and avionic domain. Existing mixed-criticality system research achieved efficient and safe scheduling through system-level criticality mode. The drawback of these approaches is performance degradation of low-criticality tasks on high-criticality mode. Task-level criticality mode is one method to address the problem and improve the performance of low-critical tasks. In this paper, we propose probabilistic performance metric for the approach. In simulation results with probabilistic performance metric, we showed that our approach has better performance than the existing approaches.

**키워드** : 실시간 시스템, 스케줄링 분석, 확률적 분석, 혼합 중요도 시스템  
Real-Time Systems, Scheduling Analysis, Probabilistic Analysis, Mixed-Criticality Systems

## 1. 서 론

기계 장치에 탑재되어 단순한 기능만을 수행 하던 내장형 시스템(embedded systems)이 컴퓨터 기술 발전에 따라 복잡하면서도 다양한 기능을 수행하며 외부 환경에 대응하는 **사이버 물리 시스템(Cyber Physical Systems, CPS)**으로 발전하고 있다. 이러한 시스템은 외부 물리 환경과의 상호작용(감지 및 동작)을 통해서 복잡화, 고성능화 등의 수많은 요구사항을 만족해야 한다. CPS는 물리적 환경과 상호작용하는 지능형 시스템으로 정의할 수 있다. 이러한 측면에서, CPS는 외부 환경과의 상호 작용 및 시스템 내부의 수많은 구성요소 간의 상호 작용을 이해하고, 해석하고, 제어하면서도 시스템의 안정성까지 유지하는 것이 요구된다. 자율주행 무인 자동차 시스템이 이러한 예에 해당한다. 무인 자동차는 교통 변화(외부 환경)를 이해하고 위성 및 교통 지도를 해석해서 최적의 주행 경로를 설계할 수 있어야 한다. 뿐만 아니라, 엔진, 브레이크 등의 내부 장치들을 안정적으로 제어하여 시스템의 안정성을 지속적으로 유지해야 한다. 물리적 환경과 상호작용하는 CPS는 시간적 정확성에 대한 연구가 다수 존재한다[5, 10].

CPS에서 복합적으로 제공되는 다양한 기능은 중요도가 다른 요소들로 구성된다. 예를 들어, 위의 무인 자동차 시스템은 자동차를 운전하는 데에 필요한 기능과 목적지까지의 노선을 설계하기 위한 기능으로 분류된다. 하지만 이 두 종류의 기능은 완전히 다른 중요도를 갖는다. 무인 자동차 시스템이 주행 중에 정지 신호를 감지했다고 가정해 보자. 이러한 상황에서는 브레이크의 작동이 그 어떤 기능보다 높은

중요도를 가진다. 혼합 중요도 시스템(mixed-criticality system)은 위의 무인자동차 예와 같이 중요도(criticality)가 다른 기능들로 구성된 시스템이다. 산업계에서는 자동차 시스템의 ISO 26262 표준과 항공기 시스템의 DO-178B 표준에서 컴포넌트의 중요도를 약 5단계로 정의하고 있다. 시간적 정확성을 확보하기 위해, 혼합 중요도 시스템에서 각각의 중요도에 따른 스케줄링 연구가 필요하다.

혼합 중요도 시스템의 스케줄링 연구는 크게 스케줄링 알고리즘 개발과 스케줄링 분석으로 이루어진다. 스케줄링 알고리즘은 시간적 안전성을 보장하기 위해 시스템에서 수행되는 각 혼합 중요도 작업을 어떤 순서로 실행해야 하는지를 기술하는 알고리즘이다. 이 알고리즘에 대한 스케줄링 분석은 주어진 혼합 중요도 작업 집합에 대한 스케줄링 가능 여부를 분석하는 것이 중요하다.

Vestal의 선구적인 연구[9] 이래, 혼합 중요도 시스템을 위한 많은 연구들이 존재한다[1-4, 6-9]. 대부분의 연구들은 시스템 수준 중요도 모드를 사용해서, 안전하면서도 효율적인 스케줄링 알고리즘을 개발했다. 시스템 수준 중요도 모드를 사용한 연구의 문제점은 모드 전환 이후 저중요도 작업에 대한 성능 저하이다. 이 문제의 해결을 위해, EDF-AD[7]은 작업 수준 중요도 모드를 제안하였다. 하지만, EDF-AD는 저중요도 작업의 중단 결정 방식의 임의성 문제와 랜덤한 스케줄링 시나리오에 의한 마감시간 만족 실패 비율(deadline miss ratio)에 대한 신뢰성 문제가 존재한다. 본 논문에서는 EDF-ADMS라는 새로운 스케줄링 알고리즘을 제시하여, EDF-AD의 임의성 문제를 해결하고자 한다. 신뢰성 있는 성능 평가를 위해, EDF-ADMS의

확률적 성능 평가 모델을 제안하고, 이 기준에 따라 성능 평가를 진행하고자 한다.

본 논문의 구성은 다음과 같다. 제 2장에서 논문의 배경에 대해 논하고, 제 3장에서는 스케줄링 알고리즘을 기술한다. 제 4장에서는 확률적 성능 지표를 제안하고, 제 5장에서는 제안하는 알고리즘을 기존 알고리즘과 비교 실험한다. 마지막으로 제 6장에서는 결론을 제시한다.

## 2. 배 경

### 2.1 실시간 시스템 및 혼합 중요도 시스템 소개

실시간 시스템(real-time system)은 시스템의 논리적 정확성만이 아니라 시간적 정확성도 요구되는 시스템이다. 실시간 시스템의 스케줄링 알고리즘(scheduling algorithm)은 시스템의 시간적 정확성을 보장하는 알고리즘이다.

스케줄링 알고리즘의 최적성(optimality)에 대하여 정의한다. 주어진 스케줄링 알고리즘에 대하여 그 알고리즘이 시스템 내의 모든 태스크의 시간적 제약 조건을 만족할 수 있으면, 시스템은 스케줄 가능(schedulable)하다고 한다. 어떤 시스템에 대하여 그 시스템을 스케줄 할 수 있는 어떤 알고리즘이 존재하면, 그 시스템은 실행 가능성이 있다(feasible)고 한다. 어떤 알고리즘이 태스크 모형에 따르는 모든 실행 가능성이 있는 시스템을 스케줄 할 수 있다면, 그 알고리즘을 최적(optimal)이라고 한다.

실시간 시스템에서는 우선순위 기반의 스케줄링 알고리즘이 많이 사용된다. 우선순위(priority) 할당 방식에 따라, 한 태스크의 모든 작업

은 동일한 우선순위를 가지는 정적 우선순위(fixed-priority) 알고리즘과 한 태스크에서 발생된 작업이라고 할지라도 상황에 따라 다른 우선순위를 가질 수 있는 동적 우선순위(dynamic priority) 알고리즘으로 분류된다. 정적 우선순위 알고리즘에서는 주기(period)가 작은 태스크가 높은 우선순위를 가지는 Rate Monotonic(RM) 알고리즘이 최적이고, 동적 우선순위 알고리즘에서는 가장 빠른 마감 시간(deadline)을 가진 작업에 가장 높은 우선순위를 할당하는 Earliest Deadline First(EDF) 알고리즘이 최적이라고 증명되었다.

본 논문은 태스크 집합(task set)  $\tau = \{\tau_i\}$ 을 고려한다. 한 태스크(task)  $\tau_i$ 은 잠재적으로 무한히 많이 발생할 수 있는 일련의 작업(job)들을 표현한다. 일반적인 실시간 태스크는 다음과 같이 표현된다:  $\tau_i = (T_i, C_i)$ , 여기서  $T_i$ 는 작업 간의 최소 분리 시간(혹은 주기(period))을 의미하고,  $C_i$ 는 최악 경우의 실행시간(worst-case execution time)을 의미한다. 본 논문에서는 각 태스크의 발생 시간(release time)에 대한 상대적인 마감 시간(relative deadline)은 주기와 동일하고, 태스크는 최대 한 개의 프로세서(processor)에서 실행할 수 있다고 가정한다.

혼합 중요도 시스템은 서로 다른 중요도(criticality)를 가진 태스크 집합을 다루는 실시간 시스템이다. 몇몇 연구들은 임의의 개수의 중요도 수준들을 고려하지만, 대부분의 기존 연구들이 이중 중요도(dual-criticality) 시스템을 고려하므로, 본 논문에서의 중요도는 두 가지, 즉, HI(높은)와 LO(낮은) 수준이 존재한다.

혼합 중요도 실시간 태스크는 다음과 같이 표현한다:  $\tau_i = (T_i, C_i^L, C_i^H, \chi_i)$ . 여기서  $T_i$ 는

주기를,  $C_i^L$ 는 **LO 중요도 최악 실행시간(LO-WCET)**을,  $C_i^H$ 는 **HI 중요도 최악 실행시간(HI-WCET)**을,  $\chi_i$ 는 태스크의 중요도(HI 혹은 LO)를 의미한다. LO-WCET와 HI-WCET를 예를 들어 설명하자면, 동일한 실행 코드에 대하여 LO-WCET는 측정으로 얻은 최악 실행 시간이고, HI-WCET는 정적 분석(static analysis)로부터 최악 실행 시간이다.

혼합 중요도 시스템의 활용도(utilization)에 대하여 정의한다. 태스크의 활용도는 다음과 같이 정의된다:

$$w_i^L = C_i^L / T_i, \quad w_i^H = C_i^H / T_i.$$

태스크 집합에 대한 활용도는 다음과 같이 정의된다:

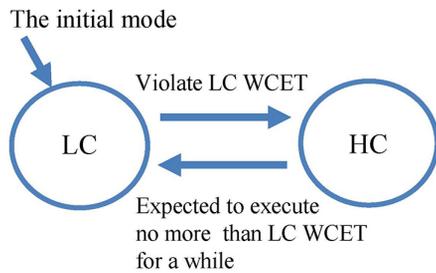
$$U_L^L = \sum_{\tau_i \in \tau_L} w_i^L, \quad U_H^L = \sum_{\tau_i \in \tau_H} w_i^L, \quad U_H^H = \sum_{\tau_i \in \tau_H} w_i^H.$$

혼합 중요도 시스템의 런타임 행동(runtime behavior)을 다음과 같이 정의 한다(<Figure 1> 참조). 모든 HI 태스크는 태스크 중요도 모드(태스크 모드)를 가지는데, 런타임에서 LO 모드로 초기화된다. LO 모드에서는 현재 작업을 포

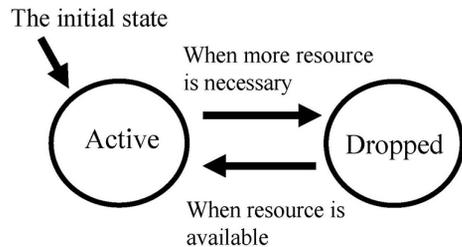
함한 현재 시점까지 발생한 모든 작업의 실행 시간이 LO-WCET을 넘지 않는다고 가정한다. **HI 행동(HI-behavior)**은 런타임에서 작업의 실행 시간이 LO-WCET 이상이고, HI-WCET 이하인 경우를 말한다. 태스크의 한 작업이 HI 행동을 보이면, 그 태스크의 태스크 모드를 LO에서 HI로 **모드 전환(mode-switch)**한다. 모든 LO 태스크는 실행 모드를 가지고 있다. 실행 모드가 **활성(active)**상태이면, 현재 태스크는 작업이 활성화된 상태임을 의미한다. 실행모드가 **중지(dropped)**상태이면, 현재 태스크는 태스크 중단 알고리즘에 의해 중단된 상태이다.

혼합 중요도 스케줄링 알고리즘은 혼합 중요도 시스템의 시간적 정확성을 만족하는 스케줄링 알고리즘이다. 혼합 중요도 시스템의 시간적 정확성은 다음과 같이 정의한다:

- 1) 모든 HI 태스크가 LO 모드이면, 모든 LO 태스크들과 모든 HI 태스크들의 각 작업들이 그 작업들의 마감 시간 이전에 LO-WCET만큼 실행하고 완료할 수 있다.
- 2) 어느 한 HI 태스크라도 HI 모드이면, 오직 모든 HI 태스크들이 그 작업들의 마감 시간 이전에 HI-WCET만큼 실행하고 완료할 수 있다.



(a) The task mode of a HC task



(b) The execution state of an LC task

<Figure 1> Task Model

## 2.2 관련 연구

Vestal의 선구적인 연구[9] 이래, 혼합 중요도 시스템에 대한 많은 연구가 제안되었다(자세한 조사는 [4] 참고). 대부분의 연구가 한 태스크가 HI 행동을 보이면, 모든 HI 태스크가 동시에 동일한 행동을 보일 수 있다고 가정한다. 이러한 시스템 수준의 모드 전환 가정 아래, 기존 연구[1, 2]는 모든 LO 태스크를 즉시 중단하는 비효율적으로 스케줄링하는 문제를 가지고 있었다.

후속 연구[3, 8]는 이 문제에 대한 해결책을 여러 가지 관점에서 제안하였다. Santy et al. [8]은 최대한 모드 전환을 연기하는 방법을 제안했고, Bate et al.[3]은 LO 모드로 복귀하는 스케줄링 알고리즘을 제안했다. 하지만, 여전히 실용적이지 못한 시스템 수준 모드 전환 가정을 벗어나지 못했다. 그 결과, 모드 전환 이후, 저중요도 작업들의 성능이 심각하게 저하되는 현상을 보였다.

최근 연구[6, 7]에서는 모드 전환의 범위를 세분화해서 적용했다. Huang et al.[6]에서는 모드 전환시 모든 LO 태스크를 중단시키는 대신, HI 태스크와 LO 태스크의 오프라인 일대다 매핑(off-line 1-to-n mapping)을 통해서, HI 행동을 보인 HI 태스크에 연결된 LO 태스크만 중단하는 알고리즘을 제안했다. Lee et al.[7]은 태스크 수준 중요도 모드를 제안하고, 어떤 태스크가 HI 행동을 보일 때 중단이 필요한 LO 태스크를 온라인에서 결정하는 태스크 중단 알고리즘을 제안했다.

본 논문에서는 Lee et al.[7]의 두 가지 문제점을 개선하고자 한다. 첫 번째로 저중요도 작업의 중단 결정 방식에서 임의성 문제를 개선

한다. 두 번째로 성능 평가 방식을 보다 신뢰성 있게 개선한다.

## 3. 작업별 모드 스케줄링 알고리즘

### 3.1 EDF-AD 스케줄링 알고리즘

Lee et al.[7]은 태스크 모드에서 중단된 LO 태스크의 개수를 최소화하는 EDF-AD(Adaptive Dropping)라는 스케줄링 알고리즘을 제안했다.

EDF-AD의 스케줄링 알고리즘을 설명하기 위해서는  $x$  파라미터를 설명할 필요가 있다. EDF-AD에서 HI 태스크는 실제 마감시간 대신 가상 마감시간(virtual deadline)을 기준으로 스케줄링 하는데, 이 가상 마감시간을 계산하는 데에  $x$ 가 필요하다. 각 태스크에 대한 가상 마감시간  $V_i$ 의 계산은 다음과 같다:

$$V_i = x \cdot T_i.$$

또한, HI 모드 선호 태스크를 정의할 필요가 있다:

$$\tau_D = \{\tau_i \in \tau_{HI} | u_i^L/x > u_i^H\}.$$

EDF-AD의 스케줄링은 다음과 같다. 첫째, HI 모드 선호 HI 태스크의 태스크 모드 초기값은 HI이고, 나머지 HI 태스크의 태스크 모드 초기값은 LO이다. 둘째, 모든 LO 태스크는 실제 마감시간을 기준으로 스케줄한다. 셋째, 모든 HI 태스크는 LO 모드에서는 가상 마감시간(virtual deadline), HI 모드에서는 실제 마감시간을 기준으로 스케줄한다. 넷째, HI 태스크의

모드 전환 순간에는 LO 태스크 중단 알고리즘을 기준으로 LO 태스크를 선택적으로 중단한다.

LO 태스크 중단 알고리즘은 다음과 같다. HI 태스크의 모드 전환으로 인해, 모든 LO 태스크가 실행되지 못하는 경우에는 다음의 온라인 스케줄링 테스트에 의해 중단된 LO 태스크를 선정한다:

$$U_{L1}^L + \frac{U_{H1}^L}{x} + xU_{L2}^L + U_{H2}^H \leq 1, \quad (1)$$

여기서  $\tau_{H1}$ 는 LO 모드인 HI 태스크 집합,  $\tau_{H2}$ 는 HI 모드인 HI 태스크 집합,  $\tau_{L1}$ 는 활성 상태의 LO 태스크 집합,  $\tau_{L2}$ 는 중단 상태의 LO 태스크 집합을 의미한다. 이때, 중단된 LO 태스크의 개수를 최소화하기 위해, 활성 LO 태스크 집합( $\tau_{L1}$ ) 중 태스크 활용도( $u_i^L$ )가 큰 태스크부터 중단한다. 하지만, 이러한 태스크 중단 알고리즘은 태스크 간의 관련성을 무시하고 공평하지 못하다는 비판이 있다. 이에 따라 본 논문에서는 태스크 간의 관련성을 고려한 중단 알고리즘을 제 3.2절에서 제안하고자 한다.

### 3.2 EDF-ADAMS 스케줄링 알고리즘

EDF-AD 태스크 중단 알고리즘의 임의성 문제를 해결하고자, 이 장에서는 EDF-ADAMS (Advanced task Dropping At Mode Switch) 스케줄링 알고리즘을 제안한다.

EDF-ADAMS에서는 혼합 중요도 태스크 집합이 주어질 때, 임의성 문제를 해결하기 위해 추가적인 정보를 요구한다. 각 HI 태스크에 대하여, HI 행동을 보일 때 중단 가능한 LO 태스크의 집합을 설계단계에서 선언하도록 한다.

EDF-ADAMS의 스케줄링 알고리즘은 EDF-AD와 동일하다. EDF-ADAMS와 EDF-AD의 차이점은 태스크 중단 알고리즘에 있다. 태스크가 HI 행동을 보일 때, 온라인 스케줄링 테스트(식 (1))를 이용하지만, HI 태스크에 대하여 중단 가능하다고 선언된 LO 태스크 집합 중 활성 상태의 LO 태스크가 중단 후보군에 오르게 된다. 실제 중단 선정 방식은 후보군 내에서 태스크 활용도가 큰 순서대로 선정한다.

### 3.3 EDF-ADAMS 스케줄링 분석기법

다음 정리에 의해 EDF-ADAMS의 스케줄 가능성을 분석할 수 있다.

**정리 1.** 주어진 태스크 집합에 대하여, 다음 두 식을 모두 만족하면 이 태스크 집합은 EDF-ADAMS에 의해 실행가능하다:

$$U_L^L + \frac{U_H^L - U_D^L}{x} + U_D^H \leq 1,$$

$$xU_L^L + U_H^H \leq 1.$$

증명:  $\tau_D$ 의 정의에 의해,  $U_L^L + \frac{U_H^L - U_D^L}{x} + U_D^H =$

$$U_L^L + \sum_{\tau_i \in \tau_H} \max(u_i^L/x, u_i^H)$$

이므로, Lee et al. [7]의 Theorem 6.5의 증명을 이용할 수 있다. ■

## 4. 확률적 성능 평가 기법

확률적 성능분석을 위해서 다음과 같은 성능 지표를 정의한다.

**정의 1**(기대 중단 비율, EDR). 주어진 태스크 집합과 HI 태스크의 태스크 모드에서, 기대 중단 비율(Expected Drop Ratio(EDR))은 총 LO 태스크 숫자에 대한 중단된 LO 태스크 개수의 기댓값의 상한으로 정의된다.

태스크 집합에 대한 EDR은 다음의 식으로 계산된다:

$$EDR = \sum_{\tau_m \in P(\tau_m)} \text{Prob}(\tau_m) \frac{\text{DROP}(\tau_m)}{nl},$$

여기서  $P(W)$ 는  $W$ 의 멱집합(power set),  $\text{Prob}(W)$ 는 LO 모드 HI-task가  $W$ 일 확률,  $\text{DROP}(W)$ 는 이때의 중단된 LO 태스크의 숫자이다.

EDR를 계산하기 위해,  $\text{Prob}(\tau_m)$ 과  $\text{DROP}(\tau_m)$ 을 알아야 한다.  $\text{DROP}(\tau_m)$ 의 경우 EDF-ADAMS의 온라인 스케줄링 테스트(식 (1))로부터 구할 수 있다.

일반적인 경우의  $\text{Prob}(\tau_m)$ 의 계산은 매우 복잡하기 때문에, 본 논문에서는 두 가지 가정을 도입하려고 한다. 첫 번째 가정은 HI 태스크 간 HI 행동을 보인 확률에 대한 독립성이다. 실제 실행시간 분포에 대한 정보를 얻기 힘들기 때문에, 우리는 이러한 독립성을 가정한다. 이러한 독립성은 확률적인 실시간 시스템 연구에서 흔히 사용되고 있다. 그러면, 우리는  $\text{Prob}(\tau_m)$ 을 다음과 같이 계산할 수 있다:

$$\text{Prob}(\tau_m) = \prod_{\tau_i \in \tau_m} (1-p_i) \prod_{\tau_i \in \tau_m} (p_i),$$

여기서  $p_i$ 는 HI 태스크  $\tau_i$ 가 HI 행동을 보일 확률을 의미한다.

두 번째 가정은 각 태스크의  $p_i$ 가  $p$ 로 동일하다는 것이다. LO-WCET는 동일한 WCET 평가 방법(예: 측정 기반 WCET 평가법)으로 추

정되기 때문에, HI 행동을 보일 확률(혹은 신뢰도)은 동일하다고 단순하게 가정할 수 있다. 따라서,  $\text{Prob}(\tau_m)$ 을 다음과 같이 다시 쓸 수 있다:

$$\text{Prob}(\tau_m) = (1-p)^k \cdot p^{nh-k}$$

여기서  $k$ 는  $\tau_m$ 의 태스크 개수이고,  $nh$ 은 HI 태스크 집합( $\tau_H$ )의 태스크 개수이다.

지금까지의 논의를 종합하여 EDR을 다음과 같이 쓸 수 있다:

$$EDR = \sum_{\tau_m \in P(\tau_m)} (1-p)^k \cdot p^{nh-k} \frac{\text{DROP}(\tau_m)}{nl}.$$

## 5. 실험

### 5.1 실험 구성

본 논문에서 혼합 중요도 스케줄링 알고리즘의 성능을 비교하기 위해 임의로 생성된 태스크 집합을 사용하였다. 스케줄링 알고리즘의 비교 대상은 다음과 같다:

- EDF: 혼합 중요도 태스크를 주기가  $T_i$ 이고 실행시간이  $C_i^H$ 인 일반 태스크로 단순 변환 후, 혼합 중요도 시스템이 아닌 일반적인 실시간 시스템 영역에서 흔히 사용되는 EDF 스케줄링 알고리즘 적용.
- EDF-VD[1]: 기존 혼합 중요도 시스템에서 높은 스케줄링 성능을 보이는 EDF 기반 스케줄링 알고리즘(시스템 수준 모드 전환 가정).
- ICG[6]: 모드 전환시 모든 LO 태스크를 중단시키는 기존 연구의 단점을 개선한 스케줄링 알고리즘(런타임 중요도 모드를 사용하지 않는다).

- EDF-AA[7]: 기존 혼합 중요도 시스템을 위한 Lee et al.[7]에서 제안한 두 가지 스케줄링 알고리즘 중 첫 번째 알고리즘(태스크 수준 모드 전환을 도입했지만, 스케줄링 성능은 낮다).
- EDF-ADAMS: 본 논문이 제안하는 스케줄링 알고리즘.

본 논문에서 혼합 중요도 스케줄링 알고리즘의 성능 비교를 위해 사용된 임의의 태스크 집합 생성 방법은 다음과 같다.  $U^b$ 를 LO 모드와 HI 모드의 활용도 상위 한계 값으로 정의한다( $U^b = \max(U_H^L + U_L^L, U_H^H)$ ). 태스크는 다음과 같이 생성한다(모든 변수는 균일(uniform)하게 뽑는다):

- $[T_L, T_U]$ : 태스크 주기  $T_i$ 의 범위.
- $[R_L, R_U]$ :  $R_i$ 의 범위( $R_i$ 를  $C_i^H/C_i^L$ 로 정의함).
- $[C_L, C_U]$ :  $C_i^L$ 의 범위.
- P: 태스크의 중요도( $\chi_i$ )를 HI일 확률은  $[0, 1]$ 에서 뽑는다. 1) 확률이 P보다 크면,  $\chi_i$ 에 HI를 할당하고,  $C_i^H = C_i^L \cdot R_i$ 으로 계산하고; 2) 그렇지 않다면,  $\chi_i$ 에 LO를 할당한다.

목표 시스템 활용도에 도달할 때까지, 태스크 집합에 태스크를 생성하는 것을 반복한다(즉,  $U_H^L + U_L^L > U^b$  혹은  $U_H^H > U^b$ ). 초과되는 마지막 태스크는 버린다. 초기 변수는 다음과 같이 주어진다:

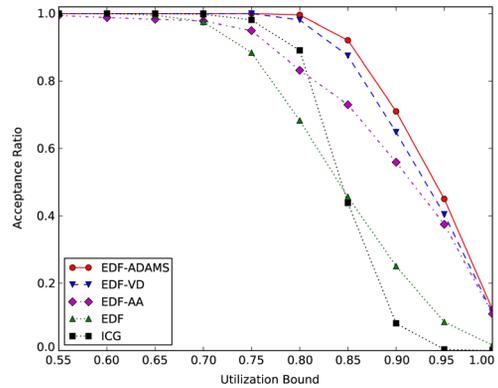
$$T_L = 10, T_U = 200, R_L = 1, R_H = 4,$$

$$C_L = 1, C_U = 10, P = 0.5.$$

## 5.2 실험결과

첫 번째 실험 결과(<Figure 2>)는 스케줄링 성능에 대한 비교이다. 다양한  $U^b$  값에 대하여

(0.55부터 1.0까지 0.05 간격으로), 스케줄링 알고리즘들의 스케줄링 성능을 비교했다. 각 x축 점에 대하여 임의의 태스크 집합을 1,000개 생성하고, y축은 각 알고리즘이 몇 퍼센트의 태스크 집합을 스케줄링 할 수 있는지를 나타냈다. <Figure 2>에서 알 수 있듯이, 70% 활용도 이하에서는 알고리즘간의 성능차이가 크지 않았다. 85% 이상의 활용도에서는 ICG 방식이 혼합 중요도 시스템을 고려하지 않은 EDF보다도 더 낮은 성능을 보였다. EDF-VD는 일반적으로 높은 스케줄링 성능을 보였다. EDF-AA는 예상대로 EDF-VD보다 낮은 성능을 보였다. 마지막으로, EDF-ADAMS는 모든 구간에서 EDF-VD보다 높은 성능을 보였다.



<Figure 2> Schedulability

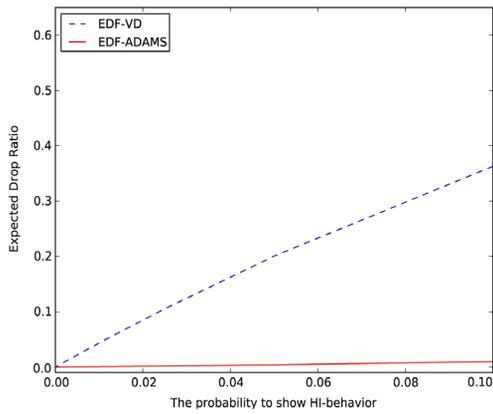
두 번째 실험은 기대 중단 비율(EDR)에 대한 성능 비교이다. 아무리 EDR 성능이 좋다고 하더라도, 기본적인 스케줄링 성능이 좋지 않다면, 실제로 사용하기 힘들기 때문에, 앞선 결과에 따라 EDF-VD와 EDF-ADAMS만을 비교한다.

x축에는 HI 행동을 보일 확률을 조절하면서 EDR를 측정하였다:  $p = \frac{0.1}{2^k}$  ( $k = 0, 1, \dots, 9$ ). 따라서,  $p = \{0.1, 0.05, \dots, 0.0002\}$ 이다. 활용도

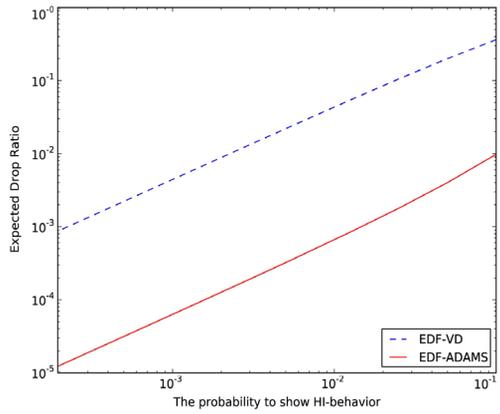
상한( $U^b$ )은 0.8로 설정했다. 5,000개의 임의의 태스크 집합에 대한 결과를 <Figure 3>에 표시하였다. EDR 성능 지표는 낮으면 낮을수록 성능이 좋음을 의미한다. p가 증가함에 따라, 제안하는 EDF-ADAMS의 EDR은 기존 연구인 EDF-VD에 비해 현저히 낮아진다. 즉, EDF-ADAMS가 p가 높을 때, 기존 연구에 비해 성능이 상당히 좋다는 것을 의미한다. 이러한 경향을 더욱 자세히 살펴보기 위해서 <Figure 4>에서는 로그 스케일(log scale)로 표시했다. p에 따라 EDF-ADAMS는 EDF-VD보다 40.2배 이상 낮

은 EDR을 보였다.

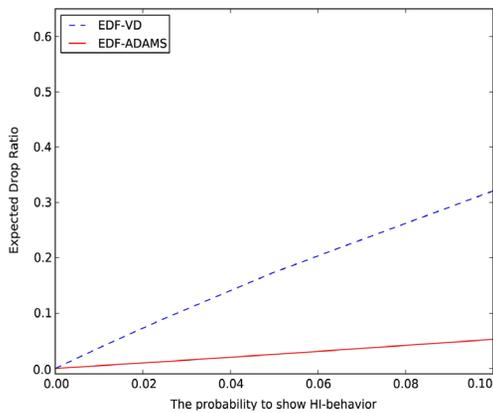
세 번째 실험도 기대 중단 비율(EDR) 성능 비교를 더 가혹한 환경에서 측정하였다. 이번에는 활용도 상한( $U^b$ )은 0.9로 설정했다. 동일하게 5,000개의 임의의 태스크 집합에 대하여 EDR을 측정하였다. <Figure 3>은 EDR 성능 비교를 표시하였고, <Figure 4>에서는 다시 로그 스케일로 성능을 표시하였다. 가혹한 환경에서는 성능 향상의 정도가 줄어들었다. p에 따라, EDF-ADAMS는 EDF-VD보다 6.4배 이상 낮은 EDR을 보였다.



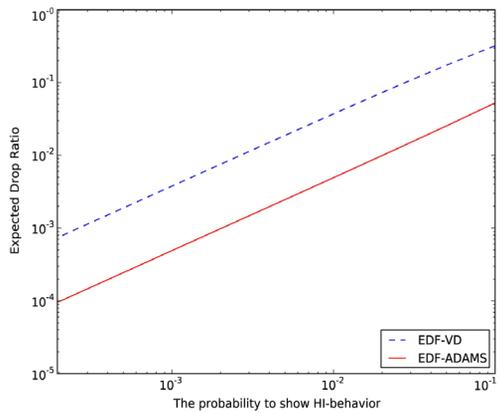
<Figure 3> EDR with  $U^b = 0.8$



<Figure 4> EDR with  $U^b = 0.8$ (log scale)



<Figure 5> EDR with  $U^b = 0.9$



<Figure 6> EDR with  $U^b = 0.9$ (log scale)

## 6. 결 론

혼합 중요도 시스템은 중요도가 다른 컴포넌트의 조합으로 이루어진 시스템을 말한다. 기존 연구에서는 시스템 모드를 이용해서, 효율적이면서 안전한 스케줄링을 추구했지만, 고중요도 모드에서 저중요도 작업의 성능 저하를 보였다. 본 연구는 이러한 문제를 개선하고자 태스크 수준 중요도 모드를 도입하고, 태스크 중단 임의성 문제를 해결한 새로운 태스크 중단 알고리즘을 제안한다. 또한, 기대 중단 비율이라는 확률적 성능 지표를 제안하고, 임의의 태스크 집합을 이용한 시뮬레이션 결과에서 기존 연구 대비 높은 성능 향상을 확인했다.

---

## References

---

- [1] Baruah, S., Bonifaci, V., Angelo, G. D., Li, H., Marchetti-Spaccamela, A., Van der Ster, S., and Stougie, L., "The Preemptive Uniprocessor Scheduling of Mixed-Criticality Implicit-Deadline Sporadic Task Systems," In the proceeding of Euromiro Conference on Real-Time Systems (ECRTS), 2012.
- [2] Baruah, S., Burns, A., and Davis, R., "Response-Time Analysis for Mixed Criticality Systems," In the proceeding of Real-Time Systems Symposium (RTSS), 2011.
- [3] Bate, I., Burns, A., and Davis, R. I., "A Bailout Protocol for Mixed Criticality Systems," In the proceeding of Euromiro Conference on Real-Time Systems (ECRTS), 2015.
- [4] Burns, A. and Davis, R., Mixed Criticality Systems-A Review, the tenth edition, Technical Report in University of York, 2018,
- [5] Chong, K.-W., Kim, J.-C., Kim, J.-I., and Lee, W.-J., "A System for Analyzing Data Transmission Time in Ubiquitous Sensor Network," The Journal of Society for e-Business Studies, Vol. 13, No. 2, pp. 149-163, 2008.
- [6] Huang, P., Kumar, P., Stoimenov, N., and Thiele, L., "Interference constraint graph -A new specification for mixed criticality systems," In the proceedings of Emerging Technologies and Factory Automation(ETFA), 2013.
- [7] Lee, J. W., Chwa, H. S., Phan, L. T. X., Shin, I. S., and Lee, I. S., "MC-ADAPT: Adaptive Task Dropping in Mixed-Criticality Scheduling," ACM Trans. Embed. Comput. Syst., Vol. 16, No. 5s, pp. 1-21, 2017.
- [8] Santy, F., George, L., Thierry, P., and Goossens, J., "Relaxing Mixed-Criticality Scheduling Strictness for Task Sets Scheduled with FP," In the proceeding of Euromiro Conference on Real-Time Systems (ECRTS), 2012.
- [9] Vestal, S., "Preemptive Scheduling of Multi-criticality Systems with Varying

Degrees of Execution Time Assurance,”  
In the proceeding of Real-Time Systems  
Symposium (RTSS), 2007.

[10] Yoon, S.-H., “Minimizing the Total Stretch

when Scheduling Flows of Divisible Re-  
quests without Interruption,” The Journal  
of Society for e-Business Studies, Vol. 20,  
No. 1, pp. 79-88, 2015.

## 저 자 소 개



이재우

2006년

2008년

2017년

2018년~현재

관심분야

(E-mail: jaewoolee@cau.ac.kr)

서울대학교 컴퓨터공학부 (학사)

서울대학교 컴퓨터공학부 (석사)

University of Pennsylvania, Ph.D in Computer and Information Science

중앙대학교 산업보안학과 조교수

실시간 시스템, 사이버 물리 시스템 보안