

<https://doi.org/10.7236/IIIBC.2019.19.1.103>

IIBC 2019-1-14

AUTOSAR:타이밍 보호 메커니즘 적용 가능한 마감시간 준수 스케줄링 방법

AUTOSAR : Deadline-Compliant Scheduling Method Applicable to Timing Protection Mechanisms

김주만[†], 김선종^{*}, 김병철^{*}, 권혁승^{*}

Joo-Man Kim[†], Seon-Jong Kim^{*}, Byoung-Chul Kim^{*}, Hyeog-Soong Kwon^{*}

요약 자동차 전장 시스템에는 여러 개의 전자 제어 장치에 시간 제약성을 갖는 다수의 응용프로그램들이 탑재되어 안전하게 수행될 수 있는 방법이 제공 되어야 한다. 본 논문은 자동차 전장용 실시간 운영체제 규격인 AUTOSAR에서 실시간성 태스크의 스케줄링시에 각 태스크의 마감시간을 준수할 수 있는 효율적인 스케줄링 기법은 제시하고, 타이밍 보호 메커니즘을 적용하는 방안을 제안한다. 고정 우선 순위나 동적 우선순위 태스크의 유연한 마감 시간 보장하기 위한 동적 비선점 알고리즘과, 오류 발생에 대한 실행 시간 모니터링 가능 위치를 제시하고, AUTOSAR 타임 보호 메커니즘 구현 방안을 제시하였다.

Abstract The automotive electronic system should provide a method that can be safely performed by loading a number of application programs having time constraints in several electronic control devices. In this paper, we propose a timing protection mechanism for AUTOSAR, which is a real - time operating system specification for automotive field, in order to observe the deadline of each task when scheduling real - time tasks. We propose a dynamic non-preemption algorithm to guarantee a flexible deadline for fixed priority or dynamic priority tasks, and a location where execution time can be monitored for errors, and suggest ways to implement the AUTOSAR time protection mechanism.

Key Words : AUTOSAR, Realtime, Time Protection, Scheduling, Operating System

1. 서 론

오늘날 자동차는 전자제품으로 분류될 정도로 기계적 장치 보다 전자 기능에 의한 편의성과 안정성에 자동화 및 지능화 되고 있다. 자동 주차 기능, 자율 주행 기능, 충돌 인식 등 차량의 첨단 서비스 개발로 인해 자동차용 전자 제어 장치를 제어하기 위하여 증가되는 소프트웨어 복잡도에 따라 자동차 전체 제조비용 중에 소프트웨어 개

발에 대한 비중이 높아 졌으며, 자동차 업계에서는 전체 제조 비용을 줄이고 소프트웨어 재사용성과 이식성을 위해 자동차 전장 제어 소프트웨어인 AUTOSAR OS 가 표준으로 자리잡아 많은 제조사들은 이러한 플랫폼하에 개발이 이루어지고 있다.^[1]

표준이 없는 상황에서 제조사마다 서로 다른 하드웨어 인터페이스와 프로토콜 때문에 전장용 장치간에 호환성이 떨어져 제품의 신뢰성이 저하되고 비용문제가 매년

^{*}정희원, 부산대학교 IT응용공학과 교수

[†]정희원, 부산대학교 IT응용공학과 교수

접수일자 2019년 1월 20일, 수정완료 2019년 2월 7일

게재확정일자 2019년 2월 8일

Received: 20 January, 2019 / Revised: 7 February, 2019 /

Accepted: 8 February, 2019

[†] Corresponding Author: joomkim@pusan.ac.kr

Dept. of Applied IT Engineering, Pusan National University, Korea

발생하였으나, 표준 플랫폼을 채용함으로써 반복적인 소프트웨어 개발 및 관리에 드는 비용 문제를 해결할 수 있다. AUTOSAR 표준에 따라 표준 인터페이스를 제공하고 모듈성, 확장성, 유연한 이식성 등의 이점을 가지게 되었다.

전장용 미들웨어 개념의 AUTOSAR 표준은 현재 4.4.0 버전까지 릴리즈 되었으며, 자동차 미들웨어를 구조화 하여 계층에 따라 요구되는 기능을 포함하고 있다. AUTOSAR OS에서 정의하고 있는 자동차 ECU용 OS는 정적 구성, 실시간성, 우선순위 기반 스케줄링, 보호 기능(메모리, 타이밍 등), 하드웨어 제어와 같은 특징을 가진다.^[1]

자동차 전장 시스템에는 많은 ECU가 자동차 네트워크에 결합된 구조로 타임 트리거드 분산 실시간 AUTOSAR 플랫폼으로 표준화된 실행 환경으로 구축된다.^[8] 특히 안전과 관련되거나, 이동을 위한 동력 기능 그리고 편의성을 위한 다양한 태스크가 AUTOSAR 운영체제하의 태스크에 의해 수행된다.^[1] 이러한 태스크는 각각 중요도에 따라 우선순위가 부여되고, 실행 시간과 마감시간에 의해 정확한 수행을 하여야 한다. 우선 순위를 정하는 문제는 매우 중요하다. 중요한 태스크에게 우선 순위를 높여주어 선점 가능하게 한다.^[5] 그러나 특정 태스크의 실행 시간을 초과하게 되어 동일 세션에서 수행 중인 다른 태스크 마감 시간을 놓쳐 전체 시스템에 오류가 전파되는 현상을 방지하기 위하여 시간 보호 메커니즘을 제공하고 있다.^{[1][3][6]}

본 논문의 구성은 2장에서 AUTOSAR의 타이밍 보호 메커니즘에 대하여 규격 관점에서 설명하고, 3장에서는 타이밍 보호 메커니즘의 규격에 준하여 구현하는 방안을 제시한다. 4장에서는 마감 시간 준수를 위한 제안된 알고리즘을 제시하였고, 5장에서 알고리즘 기반으로 시뮬레이션을 통한 결과와 분석을 통하여 본 논문의 기여도를 제시하였다. 마지막 6장에서 논문의 결론을 맺는다.

II. AUTOSAR 타이밍 보호 개념

1. 타이밍 보호 메커니즘의 정의

실시간 시스템에서 시간 오류는 실행시에 태스크나 인터럽트가 마감시간을 초과하여 생기는 현상이다. AUTOSAR 운영체제의 마감시간 모니터링은 시간 오류

를 발생시키는 태스크나를 정확하게 명시하지 못한다. 우리는 마감시간 모니터링을 태스크와의 상반 동작으로 제한하며, 인터럽트의 타이밍 모니터링은 제외한다.

마감시간의 위배는 관련성이 없는 다른 태스크나 ISR이 오랫동안 상호 간섭 혹은 중지로 인하여 발생된다. 이런 경우 이러한 결함은 관련 없는 시스템내의 모든 다른 태스크로 전파된다.^[6] 따라서 시간 오류로 마감시간을 위배한 태스크의 원인이 아니라, 시간 오류가 최초로 감지된 시점을 주의하여 볼 필요가 있으며, 타이밍 보호 매커니즘을 통한 복구 또는 삭제할 수 있는 방법으로 시스템을 안정화 할 필요성이 있다.

태스크 또는 ISR의 AUTOSAR OS는 고정 우선 순위의 선점 스케줄링에서 태스크 또는 ISR이 마감 시간을 지키는지에 대한 요인은 다음과 같다.

- 시스템 내의 태스크/ISR의 실행시간
- 낮은 우선순위의 태스크가 자원을 잠금하여 점유할 수 있는 허용시간
- 태스크의 도착율

안전하고 정확한 타이밍 보호를 위해 OS가 태스크가 해당 마감시간을 지킬 수 있도록 런타임 시 이러한 세 가지 요소를 제어할 수 있어야 한다.

2. 타이밍 보호 요구사항

운영체제는 응용 프로그램에서 발생하는 시간관련 오류가 다른 응용프로그램으로의 전파되는 것을 차단하여야 한다. 타이밍 오류의 범주는 지정된 실행 시간을 초과하거나 지정된 도착시간을 초과하여 지연하는 경우이다. 이와 같이 지정된 값의 범위를 초과하는지 여부를 조사할 필요성이 시간 보호의 목적이다. AUTOSAR OS는 고정 우선순위의 선점 OS를 실행하는 ECU의 시간 정확성은 각 태스크의 스케줄링 가능성을 통하여 보장된다.^{[3][6]} 스케줄링 가능성은 각 태스크의 실행 빈도, 실행 주기, 자원 접근과 점유 시간의 요소를 사용하여 실시간 시스템에서의 마감시간을 만족하는지를 결정하는 것이다.^[7]

시간 보호의 범위는 응용 프로그램이 기능적 동작 실패로 실행 시 분석 모델을 위배하지 않도록 보장하는데 있다. 먼저 시간 오류가 초기 테스트에서 공급 업체의 결함 있는 해당 모듈을 배제함으로써 오류 수정에 필요한 비용을 절감 할 것이고, 오류가 발생하였을 때, 이를 감지하고 태스크의 중요성을 감안하여 계속 실행할 것인지, 아니면 덜 중요한 태스크의 마감시간 오류를 예상하고,

복구할 것인지를 결정하게 된다.^[3]

3. AUTOSAR OS 실행 모델

AUTOSAR의 타이밍 보호 매커니즘은 타이밍 예산 모니터링을 통하여 실행된다. 따라서 타이밍 보호를 위하여 타이밍 오류를 이해하고, 각 태스크 모델과 이들의 스케줄링 모델 및 스케줄 가능성 분석을 통한 안전한 스케줄링 기법의 도입이 필요하다.^[7]

AUTOSAR OS에서 자원이나 태스크 및 우선순위 레벨의 결정은 시스템 구축 시에 할당되는 정적 구성 멀티태스킹 방법으로 구현된다. OS는 각 태스크의 가장 우선순위가 높은 레벨의 태스크를 실행하는 고정 우선순위 방식으로 각 태스크의 주기나 마감시간에 따라 우선순위를 결정하는 비율 단조 스킴인 RMS 방식이 일반적으로 사용된다^[3]. RMS 방식의 고정 우선순위 방식이지만, 공유자원을 보유하는 낮은 태스크로 인한 우선순위 역전 현상에 대한 우선순위 상한 프로토콜에 의하여 낮은 우선순위가 일시적으로 높은 순위값을 부여 받을 수 있다.

우리는 태스크 $T_i = (C_i, T_i, P_i)$ 로 정의한다. C_i 는 WCET 값으로 방해 없이 최대 실행 가능 시간으로 분석적 방법으로 구할 수 있다. T_i 는 마감시간 D_i 는 주기값과 동일한 것으로 간주한다. P_i 는 클수록 우선순위가 높으며, 태스크의 중요도는 자동차의 안전과 관련된 중요 부품에 부여되는 ASIL 레벨을 가진다.

4. 타이밍 오류 및 오류의 전파

타이밍 오류는 WCET를 초과하는 경우에 발생한다고 정의한다. 이러한 타이밍 오류는 오류가 발생하지 않고서 마감 시간을 위반하게 되는 다른 태스크에게 오류를 유발하는 결과를 낳는다.

표 1에서 처럼 태스크 T_1, T_2 그리고 T_3 의 시스템을 고려한다. 그림 1은 표 1에서 처럼 타이밍 오류 없이 마감 시간을 지키며 동작하고 있다.

표 1. 예제 타이밍 파라미터
 Table 1. Timing Parameter for Example

Task T_i	WCET C_i	Deadline D_i	Priority P_i
T_1	1ms	4ms	3
T_2	2ms	6ms	2
T_3	2ms	7ms	1

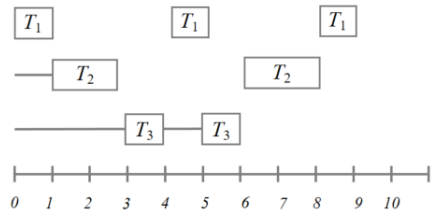


그림 1. 타이밍 버짓 오류가 없는 실행 차트
 Fig. 1. Execution Chart without Timing Budget Error

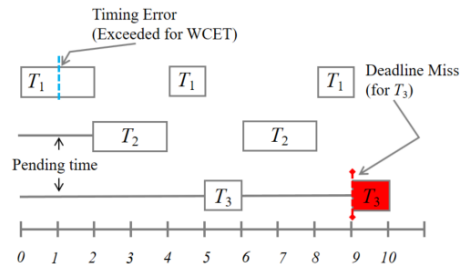


그림 2 타이밍 예산 오류로 인한 마감시간 오류 발생
 Fig. 2. Deadline error due to timing execution budget error

그림 2는 T_1 이 타임 버짓으로 설정된 1ms 값을 초과하여 2ms로 수행하므로써 타이밍 오류가 발생하였으며, 따라서 T_2 는 1ms 지연되어 결국 T_3 가 마감시간을 초과하는 오류로 인한 타이밍 실패를 보여주고 있다.

5. AUTOSAR 타이밍 안정성 규격

차량의 전장 시스템에서 전기, 전자 및 프로그래밍 시스템의 개발에 있어서 안전 기능에 대한 표준으로 ISO 26262 준수하여야 한다.^[2] 차량의 전장 시스템의 복잡도가 증가함에 따라 각각의 구성요소들을 개발할 때 이 표준에 의하여 규제를 받는다. 전자 조향 장금 시스템이나, 크루이즈 시스템 또는 브레이크 시스템에서의 오류를 피하기 위하여 특정 ECU가 오류를 감지하고 이들 오류에 대하여 어떤 제어 방식으로 대응할 수 있는 방법을 개발하여야 한다.^[3] ISO 26262는 안전한 시스템을 실현하기 위하여 ECU의 개발이 어떻게 수행되어야 하는지를 설명하는 표준 규격이다. 이 표준에는 시스템의 위험을 분류하는 4가지 자동차 무결성 안전 수준(ASIL:Automotive Integrity Safety Levels)를 정의한다. ASIL은 운전자와 도로위의 다른 운전자에게 미치는 위험성에 대하여 A, B, C, D 그리고 중요하지 않음으로 분류하며, D 등급이 안전이 가장 중요한 등급이다.^{[1][2][4]} QM은 ASIL에 따라

개발하지 않아도 되는 모듈로서 안전 등급에 구애를 받지 않는다.

III. 타이밍 보호 모니터링 구현

1. TPM : 타이밍 보호 모니터링

한 태스크에서 다른 태스크로 타이밍 오류의 전파는 실행 시에 동작하는 TPM에 의하여 감지하고 오류 전파를 방지 할 수 있다. AUTOSAR은 태스크 기반으로 선택적으로 실행되는 운영체제 서비스의 일종으로 TPM을 지정하고 구현하여야 한다. 실행시간 모니터링은 실행 시간을 모니터링하며 WCET에 해당하는 실행시간 예산과 비교하며 예산이 소진되는 오류를 감지한다. 태스크들이 활성화 되는 도착시간 모니터링은 이미 설정된 값인 “Time Frame” 이내에 어떤 빈도로 도착하는 지를 모니터링 하며 그 범위를 초과하면 타이밍 오류를 감지한다. 잠금 시간 모니터링은 우선순위를 높이거나 인터럽트를 비활성화 하여 태스크의 잠금 시간을 제한한다. 각각의 자원이나 태스크들은 “lock time budget”을 설정할 수 있는데, 해당 자원에 대하여 예산 보다 더 오랫동안 잠금 시간을 감지하면 타이밍 오류를 발생 하도록 한다. AUTOSAR는 실행 시간 모니터링을 통하여 WCET 위반에 대한 타이밍 오류를 감지할 수 있는 프레임워크를 제공한다.

3. 타이밍 보호의 실행 스케줄

AUTOSAR 운영체제는 타이밍 보호 메커니즘을 활용하는 방법으로 3가지 전략을 제시하고 있다. 첫번째는 실행 예산이라는 정적으로 구성된 상한값을 설정하여 태스크의 실행시간이 초과하지 못하도록 함으로써 시간 오류를 방지하고 있다. 두 번째는 태스크가 자원을 잡고 있는 시간 그리고 태스크에 의하여 OS인터럽트가 중지되는 시간 및 모든 인터럽트가 태스크나 ISR에 의하여 지연되거나 중지되기를 허용할 수 있도록 보장하는, 정적으로 구성된 잠금 예산이라는 상한 값인 잠금 시간 보호기법을 사용하여 시간 오류를 방지한다. 세 번째는 태스크가 스케줄링 동안 중지 상태 혹은 대기 상태에서 준비 상태로 전환하는 시간 즉, 도착시간에 대하여 정적으로 구성된 시간 프레임이라는 하한 값으로 도착 시간 보호 함으로써 시간 오류를 방지한다.

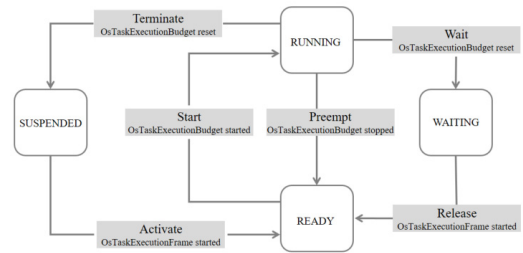


그림 3. 태스크 상태 전환 모델과 시간 보호기법 상호연동
Fig. 3. Time protection interaction with the task state transition model

도착 시간 보호는 태스크의 상태가 대기 중인 것에는 상관없이 연속적으로 활성화 되는 시간 간격을 제어한다. READY나 RUNNING상태에 있는 태스크의 활성화는 새로운 태스크의 활성화로 보아지며, 따라서 도착시간 보호는 큐에 도착하는 도착율을 제어하여 시간 오류를 방지하는 기법이다.

그림 3은 실행시간 보호와 도착시간 보호가 AUTOSAR OS의 태스크 상태 전이 모델과 상호 연동하는 방법을 보여주고 있다.

IV. 마감시간 준수 모니터링 방법

AUTOSAR에서 태스크는 고정 우선 순위로 사전 스케줄 가능성 조사를 통하여 우선순위가 부여된다.^[7] ASIL 등급에 따라 우선순위 결정되는데, ASIL D 등급은 자동차의 가장 높은 안전을 요구하므로 중요한 레벨로 분류한다. 그러나 이러한 중요도 레벨은 태스크들 간에 상호 간섭이나 다른 이유로 지연 상황이 발생하였을 때 우선 처리 방법을 제시할 것이고, 일반적으로 고정 우선순위에 따라 처리 될 것이다. 따라서 AUTOSAR에서는 시간 보호 메커니즘으로 시간 오류가 발생한 태스크로 인하여 ASIL 등급의 높은 태스크의 실행에 방해 되지 않도록 소거할 수 있는 프레임워크를 제공하고 있다.

자동차의 각 중요 부품에 부여되는 ASIL 등급과 고정 우선순위로써의 RM 방법으로 각 태스크의 우선순위가 부여된다. ASIL 등급은 중요한 태스크로서 가능한 덜 중요한 태스크로부터의 영향을 받지 않도록 하여야 함이다. 따라서 본 논문에서는 이러한 혼합된 방식으로 중요한 태스크의 각 레벨과 RM 우선순위에서의 스케줄링에서 중요한 태스크가 타이밍 오류가 발생할 때, 결정하여야

할 방법을 제시한다.^{[3][5]}

- 비선점 동적 우선순위 스케줄 방법
- 타이밍 오류시 낮은 수준의 태스크 재시동

1. 비선점 동적 우선순위 스케줄링

```

Input : Task List and the timing parameters
Output : select next scheduable Task Ti
Variables :
    util : integer // utilization
    wind : integer //
    ratei, ratej : float // utility for each task
    calculate the utilization and verification
    while hyperperiod is gone do
        select = T0;
        for i=1 to n do // for every tasks
            ratei = Ci / Di;
            if Ci != 0 then
                if Ci == Di then select = Ti;
                for j=i+1 to n //
                    ratej=Cj / Dj;
                    if Cj == 0 then continue;
                    if Cj+1 ≥ (Di - Ci) then select Ti;
                    if Cj == Dj or Ci ≥ Dj or ratei < ratej then select = Tj ;
                calculate idle and execution time for each tasks
                and change the next usage rate.
        done
    end
    
```

그림 4. 동적 비선점 스케줄 알고리즘
 Fig. 4. Dynamic non-Preemptive Schedule Algorithm

AUTOSAR는 고정 우선순위의 선점 스케줄링 기법으로 RM 기법을 가정하고 있다. 스케줄 가능성은 모든 태스크가 ECU에서 수행되는 이용율을 기반으로 결정하는데, 기존의 실시간 스케줄링 기법으로 동적인 방법의 EDF나 고정형의 RM 스케줄링에서, EDF는 이용율 $U =$

$$\sum_{i=1}^n \frac{C_i}{D_i} \leq 1 \quad \text{이고,} \quad \text{RM 기법은 이용율}$$

$$U = \sum_{i=1}^n \frac{C_i}{D_i} \leq n(\sqrt[3]{2} - 1) \text{로서 태스크의 수가 많아지면}$$

$\lim_{n \rightarrow \infty} n(\sqrt[3]{2} - 1) = \ln 2 \approx 0.69$ 69%에 접근한다. 또한 두 알고리즘 모두 선점으로 인한 부하를 무시하지 못한다. 제안하는 알고리즘은 비선점이지만 $U \approx 1$ 에 근접하는 이용율로서 효율적인 결과를 도출한다. 그림 4는 본 연구에서 제안한 스케줄링 기법으로 혼합된 우선순위 레벨 즉, 자동차의 안전성을 고려한 중요도에 따라 ASILD가 가장 높은 레벨이지만, 고정 우선순위에서는 기본적으로 주기가 짧은 태스크부터 우선순위가 부여된다. 그림 4의 알고리즘은 비선점 스케줄링으로 중요도가 높은 ASILD에서 우선적으로 고려하면서 주기 즉 마감시간과 최악실행 시

간인 WCET를 고려하여 다음번 스케줄 대상을 선정하도록 하며, 태스크 수가 늘어나더라도 약 1에 가까운 이용율에 대한 스케줄 가능성 및 스케줄링 기법을 제시하였다.

2. 타이밍 오류시 복구 시나리오

본 알고리즘에 의하여 특정 태스크가 실행 시간이 WCET를 초과하여 수행하면, 다음 수행할 태스크에게 그 지연이 전가되어 마감시간 위반 오류가 발생 할 수가 있다. 타이밍 보호 모니터링의 부분으로 구현된 알고리즘에서는 타이밍 오류를 감지하면 가능한 중요한 태스크의 실행은 보장하고, 마감시간을 위반하는 태스크의 중요성이 낮은 태스크는 시작시의 실행시간과 주기값으로 재 설정하여 재 시작 가능하게 하여 마감시간 오류로 시스템이 중지되지 않도록 한다.

V. 시뮬레이션 결과 및 평가

1. 시뮬레이션 환경

시뮬레이션은 C 프로그램으로 알고리즘을 구현하였고, 다양한 타이밍 속성을 갖는 태스크 리스트를 대상으로 실험하였다. 각 태스크의 각 주기들의 최소공배수에 해당하는 과주기(hyperperiod) 동안 스케줄 가능성 판단 할 수 있다. 다양한 태스크 집합으로 실험을 하였으며, 마감시간을 초과 태스크, 마감시간을 유발한 WCET를 초과한 태스크에 대하여 감지 할 수 있도록 하였다.

태스크 중요도 레벨인 ASILD, ASILC 그리고 자동차의 엔터테인먼트와 같은 중요하지 않은 QM 레벨등 3개의 태스크에 대하여 수행하였다. 또한 고정 우선순위로 주기가 짧은 태스크를 높은 우선순위로 하는 고정 우선순위가 부여되어 있다.

수행 결과는 과주기 동안의 태스크 실행 차트를 구할 수 있으나, 제한된 화면으로 편의상 40ms동안의 실행 차트를 캡처하여 그 결과를 분석하고 설명하고자 한다.

2. 시뮬레이션 결과 및 분석

그림 5는 이용율 0.92의 스케줄 가능한 정상적인 태스크 셋으로 과주기 동안 정상적으로 동작됨을 확인하였다. 기호중 'I'표시는 MCU의 휴지 시간(idle time)을 나타낸다. 처음 이용율이 태스크 0 보다 태스크 1이 높으므로 먼저 선택되었고, 다음 태스크 2는 2ms 남겨두고 2ms 실행

하여야 하니 즉시 실행되었다.

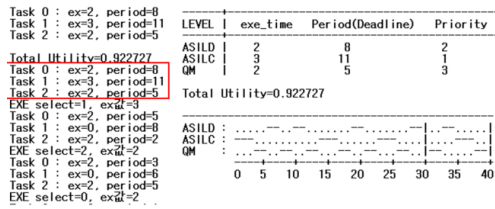


그림 5. 시뮬레이션 결과 1 (WCET 초과 없음)
 Fig. 5. Simulation Result 1(No WCET Exceeded)

그림 6은 태스크 1의 실행 시간이 3ms에서 5ms 초과하여 수행한 결과로서 결국 중요도가 낮은 태스크 2가 마감시간을 넘기는 오류가 발생하였다. 그림6에서 ‘*’는 WCET 즉, 실행시간 버짓을 초과하는 순간이고, 다음 타임프레임에서 태스크2가 ‘X’로 마감시간 초과 오류가 발생함을 알 수 있다.

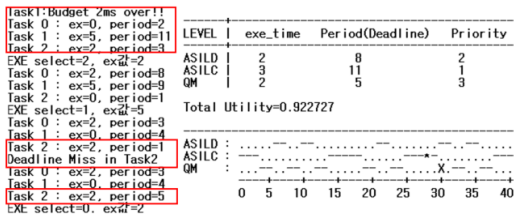


그림 6. 시뮬레이션 결과 (WCET 초과)
 Fig. 6. Simulation Result (WCET exceeded)

마감시간을 넘긴 태스크2는 다음 타임 프레임에서 초기 설정된 시간 속정을 이어 받아 재 수행 됨을 보여주고 있다.

VI. 결론

AUTOSAR는 자동차 전장용 실시간 운영체제 표준 규격이다. 자동차 전장은 안전을 기반으로 이동이나 엔터테인먼트 까지 엄격한 안전을 준수 할 수 있는 방법들을 모색하였고, 이에 AUTOSAR에서는 안전을 위한 실행시간 모니터링 기법을 제시하고 있다. 개발자는 이러한 표준 규격에 따라 모니터링을 구현하여야 함을 의미한다.

본 논문에서는 AUTOSAR의 안전 규격인 타이밍 보

호 메커니즘이 실행시간 오류를 감지하여 중요한 태스크의 실행을 보장하고, 중요하지 않는 태스크의 재실행을 통한 안전한 시스템을 구현함에 있어서, 지원하는 동적 비선점 스케줄링 기법과 특정 태스크가 타임 버짓을 초과 수행시점의 감지와 낮은 레벨 태스크의 마감시간 오류에 대한 복구를 수행하는 알고리즘을 제안 하였고, 시뮬레이션을 통하여 결과를 검증하였다.

본 결과는 AUTOSAR 표준 프레임워크의 시스템에 구현하여 실제적인 검증을 위한 연구의 지속이 필요할 것이다.

References

- [1] Classic Platform Release Overview AUTOSAR CP Release 4.4.0, Oct. 2018. <http://www.autosar.org/>
- [2] ISO 26262/ISO/DIS 26262 Road vehicles - Functional safety - Part 1-10, URL <http://www.iso.org>
- [3] A. Burns and R. I. Davis, “Mixed Criticality Systems - A Review,” Department of Computer Science, University of York, UK, Tech. Rep., 2014.
- [4] C. L. Liu and J. W. Layland, “Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment,” Journal of the ACM, vol. 20, no. 1, pp. 46 - 61, Jan. 1973.
- [5] C. Ficek, N. Feiertag, K. Richter, “Applying the AUTOSAR timing protection to build safe and efficient ISO 26262 mixed-criticality systems”, Proc. of Embedded Real Time Software and Systems (ERTS2) , 2012.
- [6] T. Piper, S. Winter, O. Schwahn, S. Bidarahalli, and N. Suri, “Mitigating Timing Error Propagation in Mixed-Criticality Automotive Systems,” in IEEE 18th International Symposium on Real-Time Distributed Computing - ISORC, 2015, pp. 102 - 109.
DOI:<http://doi.ieeecomputersociety.org/10.1109/ISORC.2015.13>
- [7] M. Asberg, M. Behnam, F. Nemati, and T. Nolte, September 2009, Towards Hierarchical Scheduling

in Autosar, IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'09).

- [8] M. Park, M. Yoon, K. Na and D. Kim, "A Study on Application of Time-Triggered Ethernet for Vehicle Network," The Journal of the Institute of Internet, Broadcasting and Communication(IIBC), Vol. 15. No.6, pp.79-88, Dec. 2015
 DOI: <http://dx.doi.org/10.7236/JIIBC.2015.15.6.79>

저자 소개

김 주 만(정회원)



- 1984년 2월 : 숭실대학교 전산학과 (공학사)
- 2003년 2월 : 충남대학교 컴퓨터공학과(공학박사)
- 1985년 ~ 2000년 : ETRI 운영체제 연구팀장(책임연구원)
- 1995년 ~ 1996년: 미국 Novell Inc 방문연구원
- 2001년 ~ 현재 : 부산대학교 IT응용공학과 교수
- 주관심분야 : 임베디드 시스템, 실시간OS
- Email : joomkim@pusan.ac.kr

김 선 중(정회원)



- 1989년 2월 : 경북대학교 전자공학과 (공학사)
- 1991년 2월 : 경북대학교 전자공학과 (공학석사)
- 1996년 8월 : 경북대학교 전자공학과 (공학박사)
- 2008년 ~ 2009년 8월 : 미국 조지아 대학교 교환교수
- 1997년 3월 ~ 현재 : 부산대학교 IT응용공학과 교수
- 주관심분야 : 신호 및 영상처리, 임베디드시스템
- Email : ksj329@pusan.ac.kr

김 병 철(정회원)



- 1994년 2월 : 영남대학교 전자공학과 공학사
- 1996년 2월 : 영남대학교 전자공학과 공학석사
- 1999년 2월 : 영남대학교 전자공학과 공학박사
- 2000년 3월 ~ 현재 : 부산대학교 IT 응용공학과 교수
- 주관심분야 : 무선통신시스템, 의료기기시스템
- Email : bckim@pusan.ac.kr

권 혁 승(정회원)



- 1995년 8월 : 영남대학교 전자공학과 공학박사
- 1996년 3월 ~ 2006년 2월 : 밀양대학교 정보통신공학과 부교수
- 2006년 3월 ~ 현재 : 부산대학교 IT 응용공학과 교수
- 2011년 ~ 2012년 : 미국조지아텍 방문교수
- 주관심분야 : 생체신호처리, 통신시스템
- Email : hskwon@pusan.ac.kr

※ 이 논문은 부산대학교 기본연구지원사업(2년)에 의하여 연구되었음.