

OSEK-OS를 위한 Matlab 도구상자와 제어시스템의 연계 모의실험에 관한 연구

김승훈, 선우명호
 한양대학교 자동차공학과

A co-simulation study on a control system with the matlab toolbox for OSEK-OS

Seunghoon Kim, Myoungho Sunwoo
 Department of Automotive Engineering, Hanyang University

Abstract - In real-time control system, it is essential to confirm the timing behavior of all tasks because these tasks of real-time controller have to finish their processes within the specified time intervals called a deadline. In order to satisfy this objective, the timing analysis of a real-time system such as a schedulability test must be performed during the system design phase. This paper presents a Matlab toolbox for simulation of real-time control system based on OSEK-OS, which is one of the most widely adopted real-time operating systems in automotive industry. The toolbox allows the user to explore the timely behavior of control algorithms, and to study the interaction between the object of the OSEK-OS, such as task, scheduler, and resource etc.

1. 서 론

최근 산업 전반에 걸쳐 임베디드 시스템에 대한 관심이 높아지면서 실시간 제어에 대한 연구가 활발히 진행되고 있다. 특히 IT산업을 비롯하여 공장 자동화, 자동차등의 분야에서 실시간 제어 시스템을 제외한 설계는 생각할 수도 없게 되었다.

실시간 제어 시스템에서 시간에 대한 고려는 매우 중요하다. 실시간 제어 시스템에서 태스크(task)들이 같은 리소스를 공유하는 등의 이유로 상호 시간지연이 발생하게 되어 고유의 시간 제약인 마감시간(deadline)을 만족하지 못하는 경우가 발생하기도 한다. 실시간 제어 시스템의 경우 이런 결과는 제어 성능에 막대한 영향을 끼칠 수 있을 뿐만 아니라 때론 심각한 결과를 낳을 수도 있다. 예를 들어 자동차에서 사람의 안전과 직접 관련된 실시간 제어 시스템과 같은 경우 정해진 시간 안에 시스템이 반응하지 않는다면 심각한 문제가 될 것이다.

따라서 이러한 태스크들 간의 시간에 대한 동작의 검증이 매우 중요하다. 이러한 검증을 위한 태스크들 간의 상호 지연이나 스케줄링 등의 시간 해석은 시스템의 설계 단계에서부터 검토되는 것이 효과적인 것이다.

이러한 관점에서 시작된 연구가 Co-Design이다.[1]. 즉 스케줄링과 제어를 따로 생각하지 않고 설계를 하는 방법이다. 이러한 설계에 대한 가능성을 제시해 준 것이 A. Cervin의 도구상자로 실시간 제어 시스템에서의 Co-Simulation을 가능케 하였다.[2] Cervin은 이 도구상자를 이용하여 제어이론과 시간해석을 한 시스템에서 효과적으로 수행하였다.

최근 자동차 산업에서도 실시간 OS에 대한 관심이 높아지고 있다. 그 중 하나인 OSEK-OS를 CO-Design

하기위해 Cervin의 도구상자를 응용해 보았지만 스케줄 방법이나 태스크 구조 등이 매우 달라 OSEK-OS를 시험하기에는 매우 부 적절하다고 판단하였다.

이번 연구에서는 OSEK-OS를 효과적으로 Co-Design 할 수 있게 도와주는 Matlab의 도구 상자를 제시하고 그 가능성에 대하여 연구하였다.

2. 본 론

2.1 OSEK-OS

OSEK/VDX는 자동차 산업에서 분산 제어 시스템을 위하여 개발된 개방형구조 (Open - ended architecture)에 대한 업계 표준을 제시하고자 유럽을 중심으로 구성된 공동 프로젝트다. 그 중 OSEK-OS 실시간 운영 체제부분으로 태스크(task)관리, 동기화, 데이터 교환, 리소스 관리, 인터럽트 관리 등을 위한 서비스와 프로세싱 메커니즘을 제공한다.[3]

2.2 도구상자 모델

제어 시스템의 개발 도구 중 하나인 MathWorks사의 Matlab을 사용하여 OSEK-OS를 기반으로 하는 시스템을 실시간 모의시험을 할 수 있는 도구상자를 제작하였다.

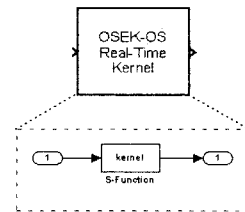


그림 1 S-function으로 제작된 OSEK-OS 도구상자.

2.2.1 커널 모델

도구상자의 커널은 Matlab Simulink의 시스템함수(S-Function)를 사용하여 제작하였다.[5]

커널은 OSEK-OS에서 Suspend, Ready 그리고 Running의 상태(state)를 가진 기본 태스크(Basic Task)와 기본 태스크의 상태에 이벤트를 사용하는 Waiting 상태를 포함하는 확장 태스크(Extended Task)를 모두 지원한다(그림 2).

OSEK-OS의 Conformance Class중 ECC1과 BCC1만을 지원하므로 태스크의 다중 활성화 및 복수 우선순위 설정은 지원하지 않는다.[3][4]

OSEK-OS의 각 Object 들은 데이터 구조체 형태로 정의 되어 커널 안에서 사용된다.

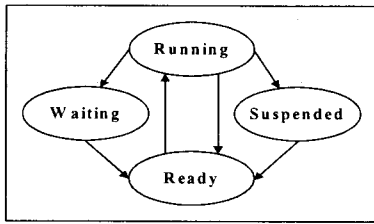


그림 2 태스크 상태도

2.2.3 스케줄러 모델

스케줄링 방법에는 보통 Tick-Driven 방법과 Event-Driven 방법이 있는데 Cervin의 도구상자의 경우 Tick-Driven 스케줄러를 사용하였지만 여기서는 OSEK-OS와 같은 Event-Driven 스케줄링 방법을 사용하였다. (2)[4] 스케줄방식은 고정우선순위 (Fixed-Priority) 스케줄링 방식으로 처음에 사용자가 각 태스크에 우선순위를 부여한다. 우선순위는 OSEK-OS와 같이 숫자가 클수록 높다.

2.2.3 태스크 모델

태스크는 시간에 대한 시험을 위해 해당 코드 외에 실행시간(Execution Time)을 포함하고 있다. 태스크 내에서 시간에 관계된 명령들은 하나 이상인 경우가 많다. 따라서 태스크의 실행시간을 시간 문제를 포함하는 명령의 따라 각각 부여 할 수 있어야 한다. 따라서 한 태스크를 시간에 영향을 미치는 명령들을 중심으로 하나 또는 여러 개의 케이스 문으로 나눈다(그림3). 각 케이스문의 실행시간은 각 케이스 문들이 실행될 때 그 케이스문에서 읽어 온다. 한 태스크의 총 실행시간은 모든 케이스문의 합과 같으며 각 케이스문은 모두 순서대로 실행된다.

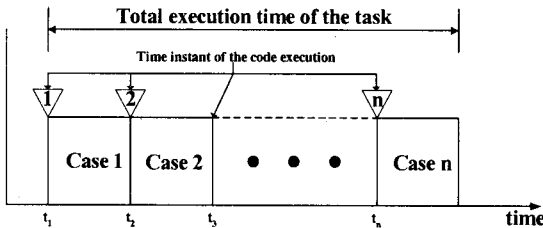


그림 3 태스크의 실행 모델

주어진 케이스문에 해당하는 명령들은 그림 3에서 표시된 것과 같이 케이스문이 실행된 그 시간에 모두 실행되어 버린다. 그 후 케이스문 실행 시 참조 했던 실행 시간만큼 가상의 CPU에서 시간을 소비한 후 다음의 케이스문을 실행한다. 이때 명령어들의 실행 위치가 매우 중요하다. 입력을 읽거나 업데이트를 하는 등의 일들은 해당 명령을 실행 후 시간을 소비해야 하고(t_1 에서 프로그램은 모두 실행된 후 t_2 까지 시간 소비), 출력을 내보내거나 메시지를 보내는 등의 명령의 경우에는 시간을 먼저 소비 후 해당 명령을 실행해야 한다(t_1 에서부터 t_2 까지 시간을 소비하고 t_2 에서 해당 명령을 실행). 위와 같이 리소스나 메시지 같이 시간에 관계된 OSEK-OS의 Object들을 사용하고자 할 때는 해당 코드들의 위치를 신중히 결정해야 한다.

OSEK-OS에서 제공하는 태스크 및 다른 Object의 서비스들을 도구상자에서도 대부분 지원하며 그 사용법도 거의 같다.

2.3 모의시험 및 결과

2.3.1 모의시험 모델

제작된 도구상자를 검증하기 위해 자동차의 부품 중 하나인 전자 스로틀 밸브의 제어 시스템(ETC)을 이용하였다. 여기서는 OSEK-OS를 기반으로 하는 PI 제어기를 제작하였다. (그림4)

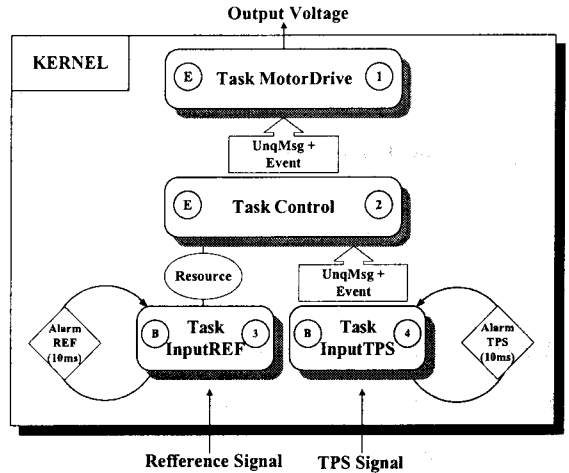


그림 4 전자 스로틀 밸브의 소프트웨어 설계

그림 4는 도구상자를 테스트하기 위해 PI 제어기를 여러 태스크로 분리 한 것이다. 태스크 이름 왼쪽에 있는 E는 Extend로 확장 태스크를, B는 Basic으로 기본 태스크를 나타낸다. 오른쪽에 있는 숫자는 각 태스크의 우선순위를 나타낸다.

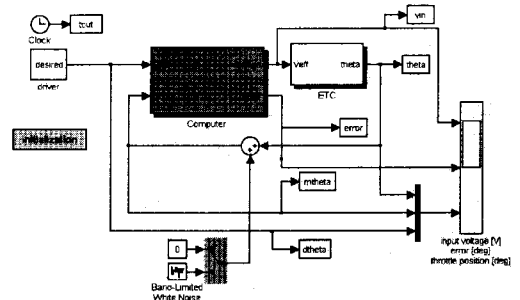


그림 5 OSEK-OS 실시간 커널을 이용하여 구성한 전자 스로틀 밸브 모델의 Simulink Diagram

그림 5는 전자 스로틀 밸브 모델을 나타내고 있으면 중앙에 컴퓨터에 해당하는 부분이 그림 6으로 이곳에 OSEK-OS 커널이 들어 있다. 그림 6에서 스케줄 블록에서는 각 태스크들의 시간에 대한 정보를 제공한다.

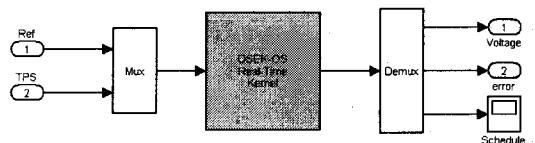


그림 6 컴퓨터 블록의 내부 구조.

2.3.2 모의시험

그림 5의 모델을 사용하여 모의실험을 수행하였다. 표 1은 각 태스크들의 실행시간과 우선순위를 나타낸다. 그림 7이 컴퓨터 블록에서 살펴 볼 수 있는 태스크 스케줄 상태도이다. 태스크 상태도는 우선순위 순으로 그려

져 있다. 각 위치에서 0.25만큼 올라가면 Ready State, 0.5는 Waiting State, 0.75는 Running State를 나타낸다. 모든 태스크들은 서로 선점 (Preemption)이 가능하도록 설정되었다.

Task	Execution Time	Priority
InputTPS	1ms	4
InputREF	2ms	3
Control	1ms	2
MotorDrive	1ms	1

표 1 실행 시간과 우선순위

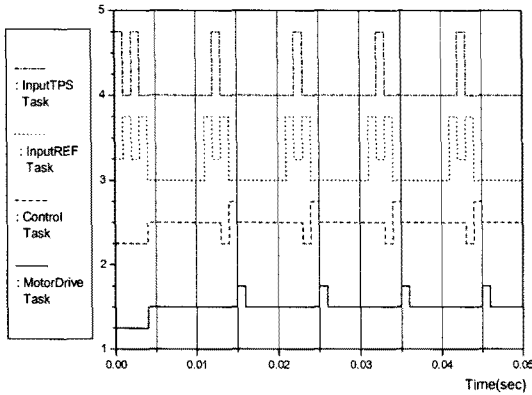


그림 7 태스크 상태

그림 8은 OSEK-OS를 기반으로 제작된 PI 제어기로 전자 스로틀 밸브 모델을 제어한 결과이다.

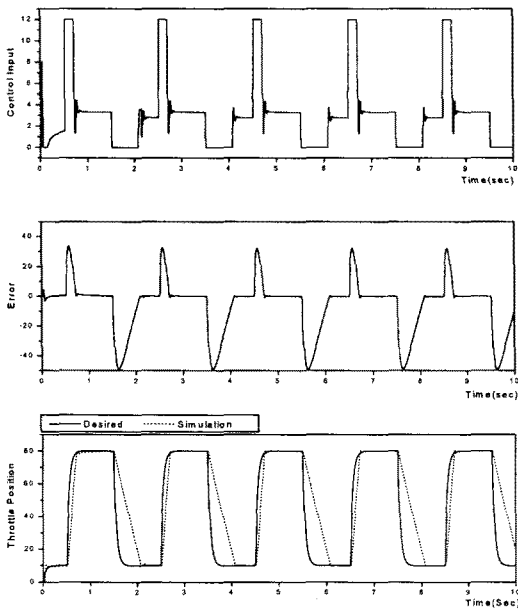


그림 8 전자 스로틀 밸브의 모의시험 결과

2.4.2 모의실험 결과 및 분석

그림 8에서와 같이 OSEK-OS기반의 PI 제어기가 제대로 작동하는 것을 확인할 수 있는데 이로써 제작된 도구상자가 Matlab Simulink를 이용한 제어시스템에 적용될 수 있음을 확인하였다.

시간에 대한 특성들은 그림 6에서 스케줄 블록에서 살펴볼 수 있는데 그림 7이 그 그래프이다. 여기서는 태스크의 현재 상태를 단계별로 나타내어 그 상태를 쉽게 확인할 수 있다.

태스크들이 일을 하기 시작하면 초기설정에 의해 모든 태스크들이 활성화된다(그림 7). 모든 태스크들은 활성화되어 suspend 상태에서 ready 상태(그래프에서 0.25만큼 올라간 상태)로 변환된 후 가장 우선순위가 높은 InputTPS 태스크가 먼저 실행(+0.75)되고 그 후 우선순위에 따라 나머지 태스크들이 실행된다. Control 태스크는 InputTPS 태스크가 보내는 메시지로 인한 이벤트로 waiting에서 ready상태로 전환되어 우선순위에 따라 실행이 되고 그 후 다시 waiting상태가 되어 이벤트를 기다린다.

처음 0.01초 동안에는 Control 태스크와 MotorDrive 태스크가 실행되지 못하는 것을 볼 수 있는데, 이것은 0.01초 동안에 Control 태스크가 우선순위에 밀려 waiting 상태가 아닌 상황에서 이벤트가 활성화되었으므로 Control 태스크는 아무 일도 하지 못한 것이다.

MotorDrive 태스크도 Control 태스크로부터의 메시지에 의한 이벤트로 실행되므로 0.01초 동안에는 실행되지 못하고 그 다음부터 실행되는 것을 확인할 수 있다.

여기서는 태스크의 실행 시간을 가상으로 설정하여 시험했다. 실제 OSEK-OS를 기반으로 하는 시스템을 개발시에 실제 하드웨어에서 코드를 수행하여 각 코드에 실행시간을 측정 후 이 모의시험을 시행하면 우선순위 (Priority)를 정하는 문제나 각 태스크들 간의 리소스 공유 문제등과 같이 시간에 관계된 문제들을 소프트웨어 상에서 해결을 할 수 있을 것이다.

3. 결 론

이번 연구를 통해 실시간 제어 시스템의 시간특성에 대한 시험을 소프트웨어 상에서 구현 가능한 도구상자를 제작하였고, 그 예로 자동차의 전자 스로틀 밸브의 제어 를 수행하였다. 시뮬레이션 시험 결과에서 보았듯 제작 된 도구상자를 이용하여 시간을 고려한 시뮬레이션이 가능하였고, 각 태스크들의 시간 동작 또한 쉽게 살펴 볼 수 있었다. 따라서 이 방법을 이용하여 OSEK-OS를 기반으로 하는 실시간 제어 시스템을 설계단계에서 시간을 고려한 시험을 통해 보다 효과적으로 개발할 수 있음을 확인하였다.

실제로 단일 CPU에서의 태스크들 간의 시간문제보다는 분산 제어 시스템에서의 시간문제가 더욱 중요하다. 이번 연구에서는 먼저 단일 CPU에서 대한 OSEK-OS 를 효과적으로 시험할 수 있는 도구상자를 제시하였다.

향후 연구에서는 OSEK-OS 도구상자를 기반으로 여러 CPU로 구성된 네트워크 시스템에서 시간의 특성에 대한 문제를 효과적으로 시험할 수 있는 CAN을 위한 도구상자에 대한 연구가 수행될 것이다.

[참 고 문 헌]

- [1] D. Seto, J. Lehoczky, L. Sha, and K. Shin. "On task schedulability in real-time control systems." In Proceedings of the IEEE Real-Time Systems Symposium, 1996
- [2] J. Eker and A. Cervin. "A matlab toolbox for real-time and control systems co-design"
- [3] Motorola, OSEK/VDX Operating System Ver2.1, 2000
- [4] Motorola, HC12 OSEK Operating System User's manual Rev1.7
- [5] Mathworks, Writing S-Functions Ver.4