

사이버 물리 시스템을 위한 실시간 시물레이션 기술 개발

김지연 · 김형종* · 강성주

Development of a Real-time Simulation Technique for Cyber-physical System

Jiyeon Kim · Hyung-jong Kim* · Sungjoo Kang

ABSTRACT

Heterogeneous physical systems and computational devices are incorporated on a large-scale in a CPS (cyber-physical system) environment. Simulations can be useful for the reliable behaviors of CPSs. Time synchronization is one of major technical issues for the simulations. In the CPS, distributed systems control themselves by interacting with each other during runtime. When some simulation models have high complexity, wrong control commands as well as incorrect data can be exchanged due to the time error. We propose a time synchronization algorithm for the hybrid model that has characteristics of both continuous time systems and discrete event systems. In addition, we develop a CPS simulator based on our algorithm. For the verification of the algorithm and the execution of the simulator, we develop an example hybrid model and simulate considering user controls as well as interactions among the distributed systems.

Key words : Cyber-physical system (CPS), Time synchronization algorithm, CPS hybrid simulator

요약

사이버 물리 시스템(cyber-physical system, CPS) 환경에서는 이기종의 물리 시스템과 연산 장치들이 대규모로 구축되기 때문에 사전에 시물레이션을 통해 시스템의 동작을 검증하는 것이 필요하다. CPS 시물레이션의 기술적 이슈 중 하나는 시간 동기화 문제이다. CPS 환경에서는 분산 시스템 간의 실시간 제어가 요구되기 때문에 모델의 복잡도가 높을 경우, 다른 모델과의 시간 오차로 인해 잘못된 데이터 및 제어 명령을 전송할 수 있다. 따라서 본 논문에서는 연속 시간 시스템 속성과 이산사건 시스템 속성을 모두 가지는 CPS 하이브리드 모델에 대하여 시간 동기화 알고리즘을 제시하고, 이를 탑재한 CPS 시물레이터를 개발한다. 또한, 시스템 간의 상호작용 뿐 아니라, 하드웨어를 통한 사용자 제어를 고려하여 CPS 시물레이션을 수행함으로써 제안된 알고리즘 및 시물레이터의 실행을 검증한다.

주요어 : 사이버 물리 시스템, 시간 동기화 알고리즘, CPS 하이브리드 시물레이터

1. 서론

사이버 물리시스템(cyber-physical system, 이하 CPS)은 네트워크를 통해 물리 시스템과 연산 체계를 통합함

*이 논문은 2014학년도 서울여자대학교 컴퓨터과학연구소 교내학술연구비의 지원을 받았음.

Received: 7 November 2014, **Revised:** 9 November 2014,
Accepted: 11 November 2014

*Corresponding Author: Hyung-jong Kim
E-mail: hkim@swu.ac.kr

Dept. of Information Security, Seoul Women's University

으로써 물리 프로세스와 상호작용 할 수 있는 컴퓨팅 시스템이다^{1, 2}. CPS 환경은 임베디드 소프트웨어를 탑재하고 있는 이기종의 물리 시스템들과 연산 장치들이 통합되어 복잡한 시스템으로 구성될 수 있으며, 이들이 가지는 제어 지식을 통해 이 복잡한 시스템을 제어할 수 있다. CPS는 교통, 국방, 발전, 보건 및 생명과학과 같은 다양한 영역에서 large-scale로 구축될 수 있지만, 복잡한 시스템 특성을 고려한다면 사전에 시물레이션을 통해 CPS 동작을 검증하고, 관련 기술을 개발하는 것이 안전한 운영을 위해 필요하다.

CPS 시물레이션의 기술적 이슈 중 하나는 시간 동기

화 문제이다. CPS는 분산 환경에 존재하는 시스템 간의 상호작용이 요구되기 때문에 만일 시스템 간의 시간 오차가 발생하는 경우, 잘못된 제어 명령에 의해 위험이 초래될 수 있다. 따라서 CPS 시뮬레이션에서 어느 한 시뮬레이터의 시간이 느려지거나 빨라지는 경우에는 분산 시뮬레이션 데이터를 신뢰할 수 없기 때문에 시뮬레이션을 중단해야 하지만, 사전에 합의된 오차가 허용되는 범위 내에서는 런타임(runtime)중 연산 속도를 조절함으로써 실시간 시뮬레이션을 지원할 수 있을 것이다. 따라서 본 논문에서는 CPS 시간 동기화 알고리즘을 개발하여 이를 CPS 시뮬레이터에 탑재하고, 알고리즘의 실행을 검증하고자 한다. 본 논문에서 개발하는 CPS 시뮬레이터는 이산 사건 시스템 모델 뿐 아니라, CPS 환경에 존재할 수 있는 다양한 연속 시스템의 시뮬레이션도 지원하도록 설계된 하이브리드 시뮬레이터이다. 또한, 다양한 분산 시뮬레이션과의 상호 운용성을 고려하여 미 국방성 모델링 및 시뮬레이션 표준인 HLA/RTI (high-level architecture/runtime infrastructure) 기반^[3]으로 개발한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 연구되었던 분산 시뮬레이션 시간 동기화 알고리즘을 살펴보고, 3장에서 제안하는 CPS 시간 동기화 알고리즘을 설계한다. 4장에서는 제시된 시간 동기화 알고리즘이 CPS 시뮬레이션 런타임에 어떻게 반영될 수 있는지 시뮬레이터 구현 측면에서 살펴보고, 5장에서 CPS 시뮬레이션을 실시 및 결과를 분석한다. 마지막으로 6장에서는 본 논문의 결론 및 향후연구를 제시한다.

2. 관련연구

기존에 제시된 분산 시뮬레이션의 시간 동기화 알고리즘의 종류는 크게 conservative 알고리즘과 optimistic 알고리즘으로 구분된다. 본 연구에서는 두 알고리즘의 기본 개념을 살펴보기 위하여 두 알고리즘의 가장 기본 원리를 제시하는 Chandy-Misra 알고리즘^[4] 및 Time Warp^[5] 알고리즘을 살펴본다.

2.1 Conservative 알고리즘

Conservative 알고리즘은 인과관계(causality)가 만족되지 않으면 그 조건이 만족될 때까지 기다렸다가 시뮬레이션을 진행하는 보수적인 시간 진행방법으로서 Fig. 1과 같이 lookahead를 고려하여 시간을 진행시킨다.

Lookahead는 프로세스가 메시지(m)를 받은 현재 시간

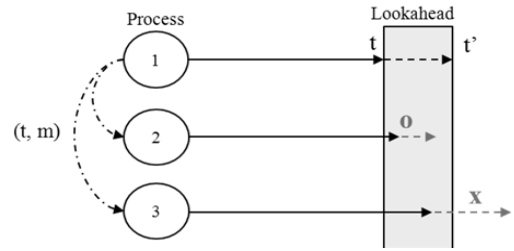


Fig. 1. A time progress of the conservative algorithm

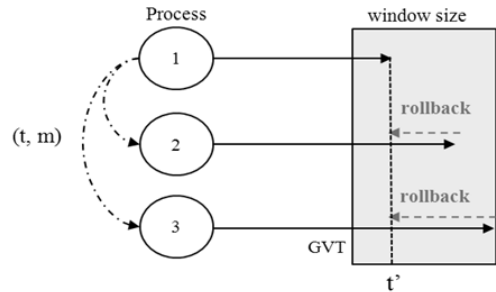


Fig. 2. A time progress of the optimistic algorithm

(t)과 다음 메시지를 받은 시간(t')의 차이를 의미하며 t'는 t까지 받은 메시지 히스토리를 통해 계산할 수 있다. Fig. 1에서 시뮬레이션 시간은 t이고, 프로세스 1은 시간 t'에 프로세스 2와 3에 메시지를 보낼 것이기 때문에 프로세스 2와 3은 시간 t' 이상의 시간을 진행시킬 수 없게 된다.

2.2 Optimistic 알고리즘

Optimistic 알고리즘은 메시지가 도착하는 즉시 처리하고, 인과관계를 만족하지 않는 상황이 발생되면 롤백(rollback)하여 시간 동기화를 진행한다.

Fig. 2와 같이 프로세스 1이 시간 t에 프로세스 2, 3에 메시지 m을 전달했다면, 프로세스 2, 3은 프로세스 1로부터 다음에 수신할 메시지 시간(t')을 고려하지 않고, 글로벌 시뮬레이션 시간과 윈도우 사이즈 내에서 다음 이벤트를 처리한다. 이후, 프로세스 2와 3이 진행한 시간보다 이전 시간인 t'에 프로세스 1이 메시지를 보낸다면 t'까지 시간을 되돌려서 다시 메시지를 처리하게 된다.

롤백 과정은 과거 이벤트가 발생한 시간으로 시뮬레이션 시간을 설정하는 과정(restoration)과 이전 상태를 취소시키는 과정(cancellation), 새로운 시간을 적용하여 다시 시뮬레이션을 진행시키는 과정(coasting forward)을 거치게 된다.

2.3 기존 분산 시뮬레이션 시간 동기화 알고리즘의 CPS 시뮬레이션 적용 방안

CPS 시뮬레이션의 경우, 분산 시스템들의 실시간 제어가 필요하기 때문에 시뮬레이션의 결과를 취소하고 과거 이벤트를 실행하는 것은 의미가 없게 된다. 따라서 optimistic 알고리즘의 적용은 어려우며 conservative 알고리즘과 같이 다음 발생될 이벤트 시간을 고려하여 시간을 진행시키는 것이 필요하다.

3. CPS 시간 동기화 알고리즘 설계

본 논문에서는 연속적인 속성과 이산적인 속성을 모두 가지는 CPS 환경의 하이브리드 모델 중, 연속 모델의 연산 주기(time step)를 조절함으로써 시간 동기화를 달성한다. 이를 위해서는 절대적인 시간을 의미하는 글로벌 시간(global time) 테이터를 기준으로 시뮬레이션의 실시간 여부를 판단하고, 실시간이 깨어진 경우에는 새로운 연산 주기를 갱신하는 과정이 필요하다.

현실적으로 시뮬레이션 시간이 실시간보다 빨리 진행되는 경우보다 모델의 복잡성 등의 이유로 시간이 느리게 진행되는 경우가 많다. 따라서 본 논문에서는 Fig. 3과 같이 CPS 시뮬레이터가 느리게 동작하는 경우에 대한 예제를 통해서 제안하는 시간 동기화 알고리즘을 설명한다.

Fig. 3의 모델 M1과 M2는 CPS 분산 시뮬레이션 환경에 존재하는 서로 다른 CPS 시뮬레이터에서 실행되는 모델이며 각 시뮬레이터에서 설정해야할 초기 지표는 다음과 같다.

- 1) 연속 모델의 초기 연산주기 (T_{initTS}) = 0.1
- 2) 글로벌 시간 수신 간격 (T_{intvGT}) = 1

글로벌 시간이 각 CPS 시뮬레이터에 전송되는 간격은 시뮬레이션 시작 전에 약속되며 일반적으로는 모든 시뮬레이터의 각 연산 주기 중 가장 긴 연산 주기보다 길게 설정해야 각 시뮬레이터가 1번의 연산 주기 동안 여러 번 글로벌 시간을 수신하는 것을 방지할 수 있다. 즉, 글로벌 시간을 수신할 때마다 실시간 진행 여부를 확인해야하기 때문에 연산 주기 내에서 두 번 이상 글로벌 시간을 수신하게 되면 실시간 진행 여부만 두 번 이상 확인해야하기 때문에 비효율적이다. 그러나 경우에 따라 특정 CPS 시뮬레이터의 시간 진행이 시뮬레이션에서 중요한 의미를 갖는다면 다른 시뮬레이터에는 비효율적이라 하더라도 특정 시뮬레이터에 적합하게 글로벌 시간의 수신 간격을 결정할 수 있을 것이다.

본 논문에서는 CPS 시뮬레이터의 실시간 진행 여부에 따라 연속 모델의 연산 주기를 갱신하기 위한 방법으로서 수식 (1)~(3)을 개발하였다.

$$R = \frac{T_{intvGT}}{T_{TS}} \tag{1}$$

$$E = T_{GT} - T_{ST} \tag{2}$$

$$T_{newTS} = \frac{T_{intvGT} + E}{R} \tag{3}$$

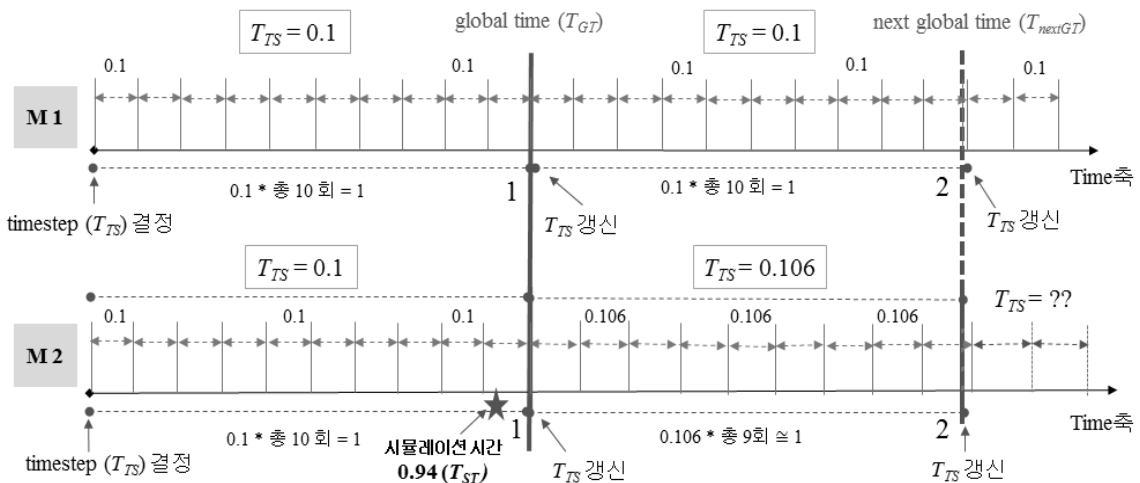


Fig. 3. An example of our time synchronization algorithm in a CPS simulation

수식 (1)에서 R(ratio)은 한 번의 글로벌 수신 간격 (T_{invGT}) 동안 몇 번의 연산이 일어나는지를 나타내는 지표이다. 실시간과의 오차를 나타내는 E (error)는 현재 시뮬레이션 시간 (T_{ST})과 현재 글로벌 시간 (T_{GT})의 차이로 계산하며 수식 (1)과 (2)를 통해 얻은 값은 궁극적으로 새로운 연산 주기(T_{newTS})를 결정하는 데에 활용된다. 새로운 연산 주기는 수식 (3)과 같이 글로벌 시간 수신 간격과 오차 범위 내에서 몇 번에 걸쳐 연산을 수행할 것인지를 판단하여 갱신할 수 있다.

Fig. 3 예제에서 각 시뮬레이터가 수행하는 시간 동기화 작업은 다음과 같은 절차에 의해 이루어진다.

- 1) 시간 0.1 마다 연속 모델의 연산을 수행
- 2) 시간 1에 글로벌 시간 수신 시, 실시간 진행여부 판단
 - 모델 M1 : $T_{GT} = 1, T_{ST} = 1$
→ 실시간으로 진행되는 것으로 판단됨
 - 모델 M2: $T_{GT} = 1, T_{ST} = 0$
→ 실시간보다 느리게 진행되는 것으로 판단
- 3) 새로운 연산주기 설정
 - 모델 M2 : $T_{newTS} = 0.12$

4. CPS 시간 동기화 알고리즘을 적용한 CPS 시뮬레이터 구현

3장에서 개발한 시간 동기화 알고리즘을 CPS 시뮬레이터에 반영하기 위해서는 분산 환경에 존재하는 각 CPS

시뮬레이터가 공통적으로 참고할 수 있는 시계가 필요하다. 본 논문에서는 이 시계 역할을 하는 시간 서버를 별도로 구현하고, 이 시간 서버가 일정한 주기마다 시간 데이터를 모든 CPS 시뮬레이터에 전송하도록 구현한다. 시간 서버가 전송하는 시간 데이터가 바로 3장에서 설명한 글로벌 시간(T_{GT})이며 각 시뮬레이터는 글로벌 시간을 활용하여 시뮬레이터의 실시간 진행 여부를 판단할 수 있게 된다. 본 논문에서는 RTI 기반으로 분산 시뮬레이션을 구현하기 때문에 글로벌 시간은 Fig. 4와 같이 RTI 통신을 통해 전달된다.

또한, 각 CPS 시뮬레이터의 경우에는 모델 실행 기능과 더불어 시간 동기화를 수행하기 위한 기능이 추가적으로 구축되어야 한다. 이를 위하여 첫 번째로는 런타임 동안 시간 서버로부터 주기적으로 전송되는 글로벌 시간을 수신하는 기능을 구축해야 하고, 두 번째로는 수신된 글로벌 시간을 CPS 시간 동기화 알고리즘에 적용하여 결과

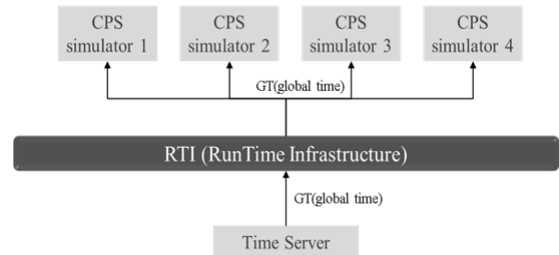


Fig. 4. A RTI-based transmission of the global time

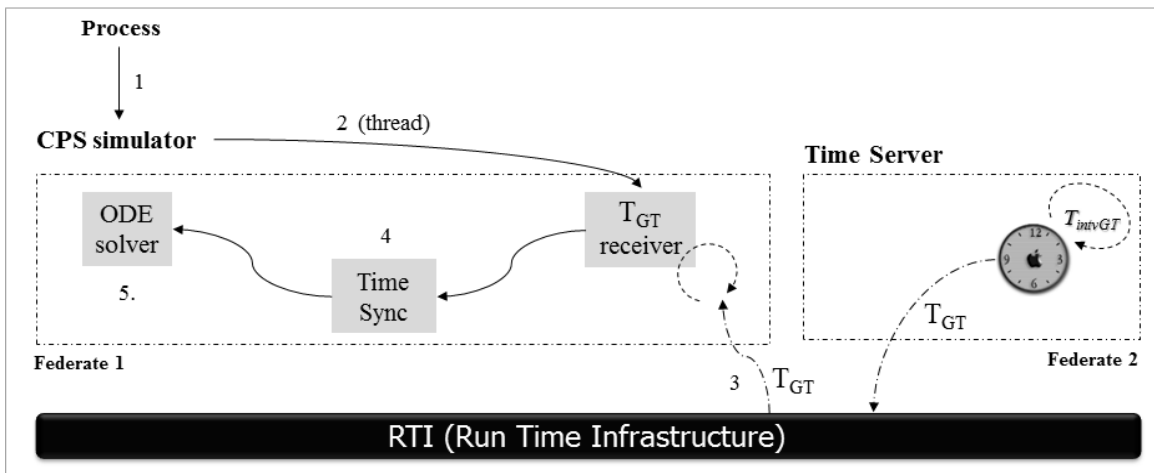


Fig. 5. A time synchronization procedure in terms of the process execution

를 시뮬레이터에 반영하여 모델을 실행하는 기능이 필요하다. 각 CPS 시뮬레이터에 시간 동기화 알고리즘이 적용되는 과정을 시뮬레이터의 프로세스 실행 측면에서 살펴보면 Fig. 5와 같다.

- 1) 주 프로세스인 CPS 시뮬레이터 시작
- 2) 글로벌 시간 수신 스레드(T_{GT} receiver) 생성 및 수신 대기
- 3) 원격지 상의 시간 서버(time server)로부터 글로벌 시간(T_{GT}) 수신 및 주 프로세스가 글로벌 시간 수신을 인지할 수 있도록 알림
- 5) 주 프로세스의 시간 동기화 과정 수행
- 6) 새로운 연산 주기(T_{newTS})를 연속 모델을 실행하는 ODE (ordinary differential equation) solver에 반영

각 시뮬레이터는 시간 서버로부터는 글로벌 데이터를 수신하지만, 다른 CPS 시뮬레이터와는 상호작용을 위한 데이터를 송수신하기 때문에 데이터의 종류를 구분하기 위해 송신자의 식별이 가능해야 할 것이다. 본 논문에서는 이를 RTI 통신 단계에서 식별할 수 있도록 Fig. 6과 같이 FED (Federation Execution Details)를 작성하였다.

```
(interactions
(class InteractionRoot reliable receive
(class RTIprivate reliable receive)
(class communication best_effort receive
(parameter command)
(parameter modelName)
) .....
)
```

Fig. 6. A FED file for the RTI-based CPS simulation

5. CPS 실시간 시뮬레이션 및 분석

본 장에서는 CPS 시간 동기화 알고리즘의 실행을 검증하기 위하여 시나리오를 기반으로 CPS 시뮬레이션 환경을 구축하고, RTI 기반으로 CPS 시뮬레이션을 수행한다.

5.1 시뮬레이션 시나리오

실제 CPS 환경에서는 CPS 시스템 간의 상호작용을 뿐 아니라, 하드웨어를 통한 사용자의 입력으로도 시스템 제어가 가능해야 한다. 따라서 본 논문에서는 현실 세계와 보다 유사한 환경에서 시뮬레이션을 수행하기 위하여 human interactive 시뮬레이션 기능을 추가로 개발하였고, 실제로 런타임 동안 사용자의 제어를 발생시키면서 실시간 시뮬레이션을 수행하였다.

Table 1은 본 논문에서 수행하는 시뮬레이션 환경에 존재하는 각 시뮬레이터의 기본 동작 및 입출력 정보를 보여주며, 이들은 Fig. 7과 같은 상호작용을 통해 시스템

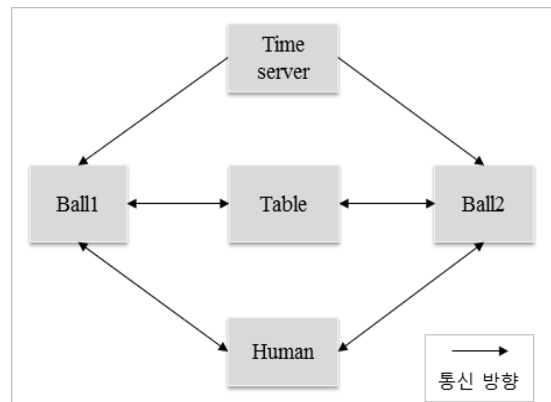


Fig. 7. Simulation models of our CPS simulation and their interactions

Table 1. Simulation models' behaviors and I/O data

시뮬레이션 모델	설명	입력	출력
Time server	일정한 주기마다 모든 원격지 시뮬레이터에 글로벌 시간 전송	-	글로벌 시간 (T_{GT})
Ball 1	런타임 동안 CPS 동기화 알고리즘을 반영하는 시뮬레이터로서 Ball의 좌표 (방향, 위치값) 정보를 Table에 전송	글로벌 시간 (T_{GT})	Ball의 좌표 (방향, 위치)
Ball 2			
Table	Ball 1, 2의 좌표를 수신하여 table 벽에 부딪히면 방향전환	Ball의 좌표 (방향, 위치)	방향전환 명령
Human	사용자의 입력을 받아 Ball 1, 2의 방향 전환	사용자의 키보드	방향전환 명령

을 제어한다.

CPS 시간 동기화 알고리즘의 동작은 Time server와 Ball 1, 2의 상호작용을 통해 확인할 수 있고, human interactive 시뮬레이션은 Human과 Ball 1, 2의 상호작용에서 확인할 수 있다.

결과적으로 이러한 시나리오를 기반으로 CPS 시뮬레이션을 수행한다면 Ball 1과 2의 시뮬레이션 결과를 포함함으로써 사용자 제어와 시스템 제어가 공존하는 현실과 유사한 환경에서도 CPS 시간 동기화 알고리즘이 정확하게 동작하는지 실행을 검증해 볼 수 있을 것이다.

5.2 시뮬레이션 결과 분석

CPS 시간 동기화 알고리즘의 동작을 확인하기 위해서는 Ball 1 또는 Ball 2가 글로벌 시간을 제대로 수신하여 실시간 알고리즘을 적용하는지를 확인해 보아야 한다. Fig. 8과 Fig. 9는 Ball 1의 이러한 실행 결과를 보여준다.

Fig. 8에서 Ball 1은 스레드로 동작하는 T_{GT} receiver를 통해 글로벌 시간을 수신하면, 이를 주 프로세스가 인지할 수 있도록 플래그를 갱신하고, Fig. 9와 같이 이를 확인한 주 프로세스는 글로벌 시간과 시뮬레이션 시간을 비교하여 시뮬레이터의 실시간 진행여부를 판단하고, 동기화가 필요한 경우, 연산 주기를 변경하여 시뮬레이션을 진행하는 것을 볼 수 있다.

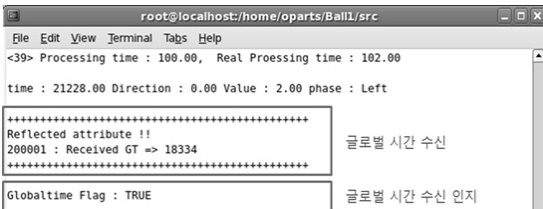


Fig. 8. Receiving the global time of the Ball 1 model

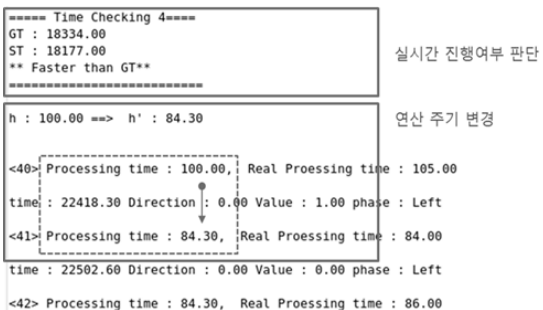


Fig. 9. A result of the time synchronization of the Ball 1 model

Fig. 10은 연산 주기가 100이고, 글로벌 시간 수신 간격이 1000으로 설정된 Ball 1이 5번의 글로벌 시간을 수신하는 시간동안 CPS 시간 동기화 알고리즘을 적용하여 시뮬레이션 한 결과 그래프이다.

그래프에서 보면 시뮬레이션 시간 약 2000 및 4000에 글로벌 시간과의 오차가 발생하였고, 이 오차를 시간 동기화 알고리즘을 적용하여 런타임 중에 극복하기 위한 시도가 시간 약 2000-3000, 그리고 4000-5000에 이루어졌음을 볼 수 있다. 여기서 시간 앞에 ‘약(대략)’을 붙인 이유는 글로벌 시간이 수신되는 시간, 그리고 CPS 시뮬레이터의 주 프로세스가 스레드에서 수신한 글로벌 시간을 확인하는 매우 짧은 프로세서의 처리 시간에 의해 약간의 지연이 발생되기 때문이다.

시간 2000-3000에는 실시간 보다 느려진 시뮬레이션 시간을 만회하기 위하여 다음 글로벌 수신 시간까지 연산 주기가 100보다 크게 갱신되었고, 기본 값보다 큰 연산 주기에 의해 기존에는 다음 글로벌 수신 시간까지 10번의 연산을 수행하였다면, CPS 시간 동기화 알고리즘에 의해 9번만 연산을 수행함으로써 느려진 속도를 만회할 수 있게 된다. 반대로 시간 약 4000-5000에는 빨라진 속도를 감소시키기 위해 연산주기가 100보다 작게 갱신되었으나, 글로벌 시간 전송 및 확인 지연이 발생된 것을 처리하여 11번이 아닌 10번의 연산을 수행하여 실시간으로 진행하게 된다.

Fig. 11은 Table과 Ball 1의 상호작용 결과를 보여준다. Ball 1에서 Table로부터 방향 전환 메시지를 받은 후, 방향을 ‘right’에서 ‘left’로 바꾸어서 진행하고, 연산을 수행할 때마다 위치가 ‘9->10->9->8’과 같이 변하는 것을

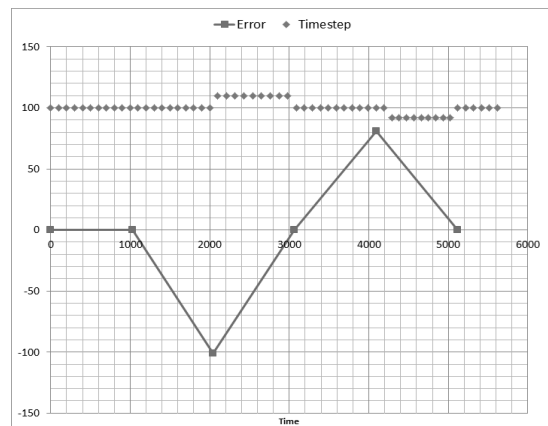


Fig. 10. The change of the timestep during the CPS simulation

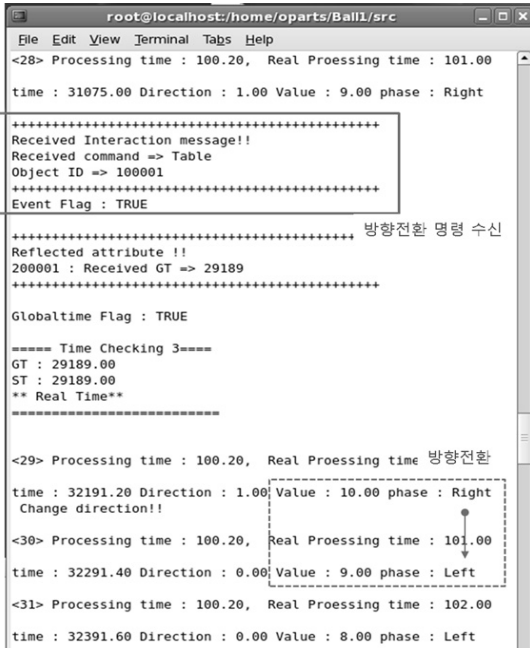


Fig. 11. The change of the Ball 1 model through interaction with the Table model

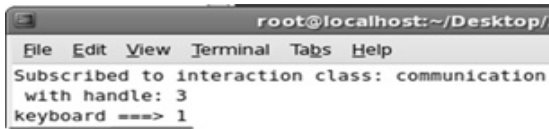


Fig. 12. Simulation controls by the Human model

확인할 수 있다. 또한, Table과의 상호작용 과정 사이에 Time server와의 상호작용에 의해 시간 동기화 과정이 수행되는 것도 확인된다.

CPS 실시간 시뮬레이션에서 사용자 제어를 가능하게 하는 human interactive 시뮬레이션 결과는 Fig. 12, Fig. 13과 같다. 먼저 Fig. 12는 제어를 원하는 Ball 1을 키보드를 통해 선택하는 과정이며, Fig. 13은 선택된 Ball 1에서 사용자 제어 명령을 반영하여 방향을 ‘right’에서 ‘left’로 바꾸어 진행하는 결과를 보여준다.

이와 같은 시뮬레이션 결과, 5.1장에서 작성한 시나리오와 같이 Ball 1이 Time server, Table, 그리고 Human과의 상호작용을 통해 프로그램 된 제어사항을 정확하게 수행하고 있음을 확인할 수 있다. 이는 본 논문에서 제안한 CPS 시간 동기화 알고리즘의 뿐 아니라, 이를 반영하여 개발한 CPS 시뮬레이션 엔진 및 시뮬레이션 모델의 설계 및 구현, 그리고 실행을 검증하는 결과이다.

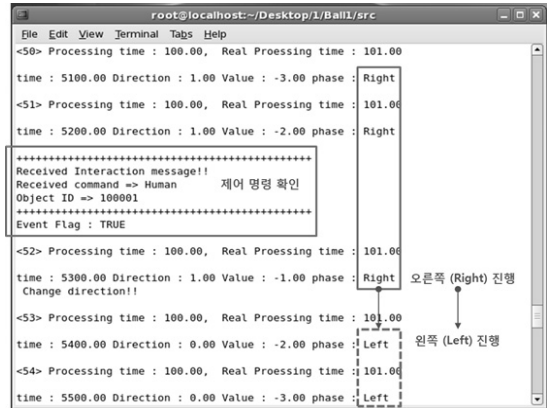


Fig. 13. The change of the Ball 1 model by the Human model

6. 결론 및 향후 연구

CPS는 임베디드 소프트웨어를 탑재한 이기종의 물리 시스템들이 연산을 통합함으로써 시스템을 제어하는 컴퓨팅 기술이다. CPS는 large-scale로 구축되어 복잡한 시스템을 구성하기 때문에 안전한 CPS 운영을 위해서 시뮬레이션을 수행하는 것이 필요하다.

CPS는 분산 환경의 시스템 간의 실시간 제어가 필요하기 때문에 동기화 되지 않은 시간을 기반으로 상호작용하면 잘못된 데이터 및 제어 명령에 의해 위험이 발생할 수 있다. 따라서 본 논문에서는 CPS 시뮬레이션을 위한 시간 동기화 알고리즘을 설계하고, 이를 탑재한 CPS 시뮬레이터를 개발하였다. 또한, 제안된 알고리즘 및 이의 실행을 검증하기 위하여 시나리오를 기반으로 CPS 실시간 시뮬레이션을 수행하였다. 이 때, 보다 현실 세계와 유사한 시뮬레이션 환경을 구축하기 위하여 CPS 시스템 간의 상호작용 뿐 아니라, 하드웨어를 통한 사용자와의 상호작용을 고려하여 사용자의 제어가 이루어지는 상황에서도 제안된 알고리즘을 반영한 실시간 시뮬레이션이 수행되는 것을 확인하였다. 본 논문에서 수행한 시나리오 기반의 CPS 시뮬레이션을 통해 제안된 알고리즘 뿐 아니라 알고리즘이 적용된 시뮬레이터의 설계, 구현, 그리고 실행을 검증할 수 있다.

CPS 시뮬레이션은 적용 분야마다 요구하는 시뮬레이션 결과의 정확도와 실시간성 정도가 다르다. 즉, 적용 분야의 특성에 따라 시뮬레이션 시 우선시하는 지표가 다르다. 따라서 각 도메인의 전문가가 시뮬레이션 모델 실행을 위한 적절한 연산 주기 및 오차 허용 범위 등을 제시할 수 있어야 하며, 이러한 데이터를 반영하여 CPS 실시간

시물레이션을 수행함으로써 시간 동기화 알고리즘의 성능을 검증하는 것이 필요하다.

References

1. Dan Henriksson and Hilding Elmqvist, "Cyber-Physical Systems Modeling and Simulation with Modelica", 8th International Modelica Conference, 2011.
2. Paul Bogdan and Radu Marculescu, "Cyberphysical systems: Workload Modeling and Design Optimization", IEEE Design and Test of Computers, July/August 2011.
3. "RTI 1.3 - Next Generation Programmer's Guide Version 3.2", Defense Modeling and Simulation Office, April 2000.
4. K. Mani Chandy and Jayadev Misra, "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs", IEEE Transaction on Software Engineering, Vol. SE-5, No. 5, September 1979.
5. David Jefferson and Henry Sowizral, "Fast Concurrent Simulation using The Time Warp Mechanism, Part 1: Local Control", Rand corporation, December 1982.
6. "Cyber-Physical Systems: Executive Summary", CPS Steering Group, March 2008.
7. Radhakisan Baheti and Helen Gill, "Cyber-physical System", The Impact of Control Technology, 2011.
8. R.M. Fujimoto, "Time Management in the High Level Architecture", Simulation Special Issue on High Level Architecture, vol. 71, no. 6, 388-400, 1998.
9. Carolyn Talcott, "Cyber-Physical Systems and Events", Software-Intensive Systems, LNCS 5380, pp. 101-115, 2008.
10. P. E. Wellstead, Introduction to Physical System Modelling, Control Systems Principles, 2000.
11. Huang-Ming Huang et al., "Cyber-Physical Systems for Real-Time Hybrid Structural Testing: A Case Study", ICCPS'10, Stockholm, Sweden, April 2010.



김 지 연 (kimjy@andrew.cmu.edu)

2007 서울여자대학교 정보보호공학과 공학사
2013 서울여자대학교 일반대학원 컴퓨터학과 이학박사
2014~현재 Carnegie Mellon University 박사후연구원

관심분야 : 사이버 보안, 모델링 및 시물레이션, 클라우드 컴퓨팅, VoIP 보안



김 형 종 (hkim@swu.ac.kr)

1996 성균관대학교 정보공학과 공학사
1998 성균관대학교 정보공학과 공학석사
2001 성균관대학교 전기전자 및 컴퓨터공학과 공학박사
2001~2007 한국정보보호진흥원 수석연구원
2004~2006 미국 Carnegie Mellon University CyLab Visiting Scholar
2007~현재 서울여자대학교 정보보호학과 부교수

관심분야 : 취약점 분석 및 모델링, 이산사건 시물레이션 방법론



강 성 주 (sjkang@etri.re.kr)

2003 한양대학교 전자전기공학부 졸업
2005 한양대학교 전자통신전파공학과 석사
2010 한국과학기술원/Carnegie Mellon University 소프트웨어공학석사
2005~현재 한국전자통신연구원 선임연구원
2014~현재 충남대학교 컴퓨터공학과 박사과정

관심분야 : 사이버 물리 시스템 (CPS), 자율제어, 모델링 및 시물레이션, 소프트웨어 공학