

OMG DDS 미들웨어를 이용한 FMI기반 실시간 CPS 분산 시뮬레이션 프레임워크

FMI based Real-time CPS Distributed Simulation Framework using OMG DDS middleware

홍 석 준 *, 조 인 휘 *, 김 원 태**★

Seokjoon Hong*, Inwhhee Joe*, Wontae Kim**★

Abstract

To develop highly dependable CPS, M&S(modeling and simulation) is very important. It is not easy to model any CPS whole system in a single simulation tool because each simulation tool is optimized for modeling each different part of the CPS. The FMI is the standard for M&S between different simulation tools. The DDS is a communication middleware suitable for large-scale real-time data transmission. In this paper, we proposed FMI based CPS real-time distributed simulation framework using DDS. To evaluate the performance of the proposed framework, we performed distributed simulation using IEEE HLA/RTI and OMG DDS middleware and measured and compared the execution time of the entire simulation. From the simulation results, we can confirm that the simulation execution time using DDS is at least 1.14 times faster compared to execution time using HLA/RTI.

요 약

고신뢰 CPS 개발을 위해서는 M&S는 매우 중요하다. 각각의 CPS 시스템의 다른 부분을 모델링하기 위해 각각의 다른 시뮬레이션 프로그램들을 사용하기 때문에 어떤 CPS 전체 시스템을 하나의 시뮬레이션 툴에서 모델링하기는 쉽지 않다. FMI는 다른 시뮬레이션 툴들 간에 M&S를 위한 표준이다. 또한, DDS는 대규모의 실시간 데이터 전송에 적합한 통신 미들웨어이다. 따라서 이 논문에서는 DDS를 이용하는 FMI기반의 CPS 실시간 분산 시뮬레이션 프레임워크를 제안한다. 제안하는 프레임워크의 성능 평가를 위해서 IEEE HLA/RTI와 OMG DDS 미들웨어를 사용해서 분산 시뮬레이션을 수행하고 전체 시뮬레이션 수행 시간을 측정하고 비교해보았다. 성능 평가 결과를 통해서 DDS를 사용하는 시뮬레이션 수행 속도가 HLA/RTI를 사용하는 것에 비해 최소 1.14배 이상 빠른 것을 확인할 수 있었다.

Key words : CPS, DDS, HLA/RTI, FMU, Distributed co-simulation.

* Dept. of Computer Software, Hanyang University

** Dept. of Computer Science and Engineering, Koreatech University

★ Corresponding author

E-mail:wtkim@koreatech.ac.kr, Tel:+82-41-560-1485

※ Acknowledgment

This paper was supported by the National Research Foundation of Korea(No. 2017R1A2B4010875).

Manuscript received Mar. 9, 2018; revised Mar. 24, 2018 ; accepted Mar. 28, 2018

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서론

M&S(모델링&시뮬레이션)는 다양한 산업 혹은 과학 응용분야에서 복잡한 시스템(System of systems)을 설계하고 연구하기 위한 중요한 기술이다 [1].

특히, 자동차와 항공기와 같이 기계, 전기 등의 하드웨어 요소와 내장형 컴퓨터와 소프트웨어 요소들이 복합되어 시스템을 구성하고, 높은 신뢰성을 요구하는 CPS를 구현하기 위해서는 M&S가 필수적으로 요구된다 [2].

보통 시뮬레이션 툴들은 CPS를 구성하는 각 파트를 모델링하여 파트별로 시뮬레이션하는데 최적화되어 있기 때문에 전체 CPS를 통합해서 시뮬레이션하는 것은 매우 어렵다. 또한, 보통 시뮬레이션 툴들은 각 툴에서 사용하는 모델의 형식이 정해져있기 때문에 다른 툴에서 모델을 사용하는 것은 쉽지 않다.

따라서, 전체 CPS 시스템을 통합해서 시뮬레이션하기 위해서는 그 CPS 시스템을 구성하는 여러 구성 요소들을 모델링하고 각 모델을 시뮬레이션하는 프로그램들 간 시뮬레이션 데이터들을 연동하여 전체 시뮬레이션이 수행되도록 하는 방법이 필요하다.

CPS 시스템의 서로 다른 시뮬레이션 모델을 연동하는 것을 위한 대표적인 표준으로 FMI(Functional Mock-up Interface) 표준이 있다. FMI는 서로 다른 시뮬레이션 모델의 연동을 위해서 시뮬레이션 모델을 FMU(Functional Mockup Unit)로 표준화하고 이를 접근하기 위한 인터페이스에 대해서 정의한다 [3].

본 논문에서는 각 분산된 노드들의 시뮬레이션 모델인 FMU들을 FMI를 기반으로 시뮬레이션하면서 통신 미들웨어를 통해서 상호 연동하여 시뮬레이션하는 프레임워크를 제안한다. 또한, 분산된 여러 모델들은 체인 형식으로 연결되는 경우로 가정하였고, 이런 형태의 모델들의 분산 시뮬레이션을 위한 동기화 기법을 제안한다.

그리고 분산 시뮬레이션의 경우, 분산된 노드 간 시뮬레이션 데이터를 전달해주는 것이 필요한데 자동차와 항공기와 같은 복잡한 CPS에서 HILS(Hardware-in-the-Loop simulation)같은 시뮬레이션을 하기 위해서는 실시간의 데이터 전송 요구사항도 만족되어야 한다 [4]. 따라서 본 논문에서는 큰 규모의 실시간 데이터 전송에 적합한

DDS 통신 미들웨어를 사용하는 분산 시뮬레이션 구조를 제안한다.

본 논문은 2장에서 FMI와 통신 미들웨어인 HLA/RTI(High Level Architecture/Run-Time Infrastructure) 그리고 DDS(Data distribution service)에 대해서 간략히 알아보고, 다음으로 CPS 관련 분산 시뮬레이션하는 기존 연구에 대해서 살펴본다 [5][6]. 3장에서는 제안하는 DDS를 사용하는 FMI기반의 실시간 CPS 분산 시뮬레이션 구조에 대해서 소개하고, 4장에서는 HLA/RTI와 DDS를 사용하여 어떻게 분산 연동 시뮬레이션을 구현하였는지 설명하였다. 마지막으로 5장에서는 3, 4장에서 제안한 구조로 실제 시뮬레이션을 수행하였을 때, 결과를 비교 분석한다. 마지막으로 6장에서 본 논문의 결론을 맺는다.

II. 관련 연구

FMI는 XML파일들과 C 코드(혹은 DLL/공유 라이브러리로 컴파일된 것)를 사용하는 동적 모델의 모델 교환(ME : Model Exchange)과 동시 시뮬레이션(CS : Co-simulation) 방식을 모두 지원하는 시뮬레이션 툴 독립적인 표준이다. 또한, 이때 모델을 구현하는 XML파일들과 C코드를 FMU(Functional Mockup Unit) 라고 한다 [3].

모델 교환 방식의 FMI는 다른 시뮬레이션 환경에서 재사용하거나 통합하기 위해 어떤 모델링 툴에서나 하나의 모델을 나타내는 FMU 생성을 허용하는 것이다. 동시 시뮬레이션 방식의 FMI는 동시 시뮬레이션 환경에서 두 개 혹은 그 이상의 시뮬레이션 툴들의 연동을 위한 표준이다. 이 방식의 FMI는 또한 동시 시뮬레이션 환경에서 실행되는 모델이나 서브 시스템들을 나타내는 하나의 마스터와 슬레이브들 간의 통신을 관리하는 루틴들을 정의한다. 슬레이브들은 마스터에 의해 알려진 미리 정해진 입력과 출력값들의 집합들을 갖고 있고, 마스터는 시뮬레이션 실행동안 슬레이브들을 셋팅하고 조정하는 책임이 있다.

위 두 가지 방식의 FMI는 또한 모델을 해석하는 솔버(Solver)의 위치가 다른데, 모델 교환 방식에서는 툴에서 제공되어야 하지만, 동시 시뮬레이션 방식은 FMU에 솔버가 포함되어 자체 해석이 가능한 특징이 있다.

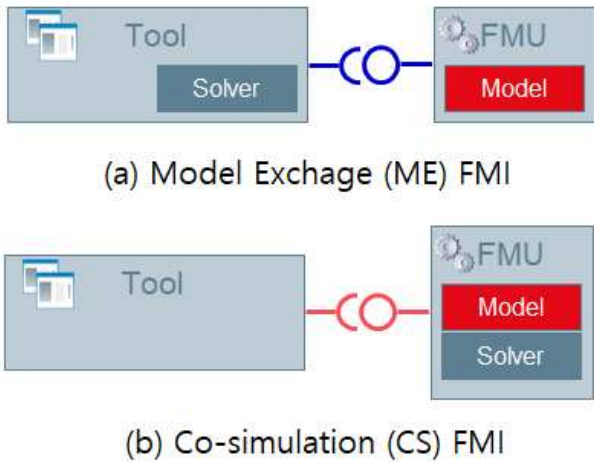


Fig. 1. Two types of FMI
그림 1. 두 가지 방식의 FMI

HLA는 분산 컴퓨터 시뮬레이션 시스템을 위한 IEEE 표준(IEEE 1516)이다 [5]. HLA 표준에서 하나의 분산 시뮬레이션을 페더레이션(Federation)이라고 하고 이것은 RTI(Run-Time Infrastructure)를 사용하여 서로 통신할 수 있는 HLA 시뮬레이션 객체들인 페더레이트(Federate)들로 구성된다.

DDS는 OMG(Object management Group)에서 표준화된 Pub/Sub(Publisher/Subscriber)방식의 통신 미들웨어이다 [6]. DDS의 Pub/Sub 방식의 통신은 서버 클라이언트 방식에 비해서 대규모 데이터 전송과 실시간 전송에 적합하다. DDS에서 도메인은 각 응용을 통신을 위해 묶어주는 개념이다. 하나의 참여자(Participant)는 하나의 도메인에서 응용이 되며 출판자(Publishers), 구독자(Subscriber), Datawriters, DataReaders 및 Topics을 생성하는데 사용된다. 또한, 출판자와 구독자는 메시지 혹은 데이터를 전송하고 수신하는데 사용되며, Datawriter와 DataReader는 실제 데이터를 쓰고 읽는 주체이고 이때 각각의 출판자와 구독자는 Topic을 통해 연결된다.

그림 2에서 볼 수 있듯이 HLA RTI는 RTI라는 미들웨어가 동작되어 이를 통해서 모든 메시지가 전달되고, DDS는 RTPS(Real-time Publish Subscribe wire protocol)을 통해서 전달되어지게 된다.

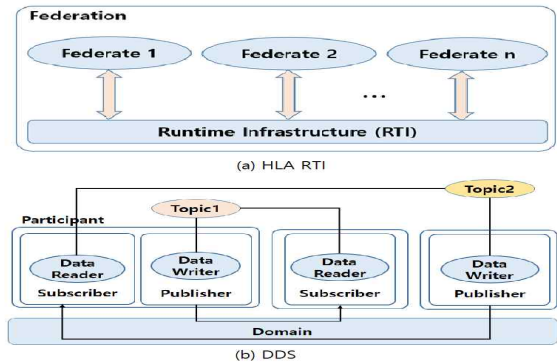


Fig. 2. The comparison of HLA/RTI and DDS middleware operations
그림 2. HLA/RTI and DDS 미들웨어 동작방식 비교

또한, 이전 HLA RTI와 DDS의 데이터 전송 성능 비교 연구를 통해서 DDS가 HLA RTI에 비해 빠른 업데이트 속도와 낮은 전송지연 시간을 갖는 것을 확인하였다 [7]. 따라서 본 논문에서는 실시간 분산 시뮬레이션을 위해 DDS를 사용하는 구조를 제안한다.

위의 표준들을 사용한 CPS 관련 분산 시뮬레이션 관련 이전 연구로는 스마트 그리드에서의 주요 응답을 위한 통합 시뮬레이션을 위해 Virgil이라는 플랫폼을 기반으로 하여 FMI를 사용하여 연동하여 시뮬레이션하는 연구와 FMI와 HLA RTI를 결합하여 분산 시뮬레이션을 수행하는 구조를 제안하는 연구가 수행되었었다 [8][9].

III. DDS 미들웨어를 이용한 FMI 기반 실시간 CPS 분산 시뮬레이션 구조

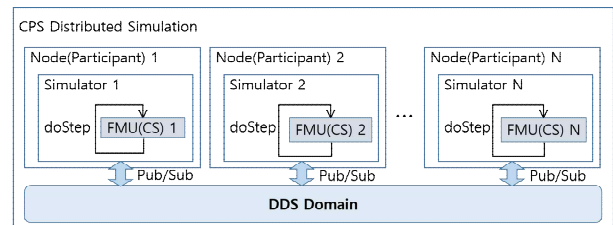


Fig. 3. The distributed co-simulation architecture
그림 3. 분산 연동 시뮬레이션 구조

그림 3은 이 논문에서 제안하는 DDS를 사용하는 FMI기반의 CPS 분산 시뮬레이션 프레임워크의 구조를 보여준다.

제안하는 구조의 세부 요소의 역할은 다음과 같다. 먼저 각 노드는 FMU 모델을 시뮬레이션

할 수 있는 컴퓨터 혹은 임베디드 시스템 내부의 DDS의 참여자가 될 수 있다. 다음으로 각 노드에서는 FMU 모델을 시뮬레이션할 수 있는 시뮬레이션 프로그램을 통해서 FMU모델의 시뮬레이션 입력값을 받아 FMI 인터페이스를 이용하여 시뮬레이션을 수행(doStep)하고 결과값을 DDS 미들웨어를 통해서 전달한다. 또한, 이 구조에서의 FMU 모델은 동시 시뮬레이션이 가능하도록 하기 위해 CS(co-simulation) 타입을 사용하도록 한다.

본 논문에서 제안하는 FMU 모델간 시뮬레이션 데이터 연결 구조는 체인방식으로서 각 FMU 모델의 입력과 출력은 그림4와 같이 매핑된다. 이 논문에서의 CPS 시스템은 모든 각 요소들이 하나의 입력값을 받고 하나의 출력값을 전달한다고 가정하였다.



Fig. 4. The input/output mapping of each FMU
 그림 4. 각 FMU의 입력과 출력의 매핑

이때의 모든 FMU 모델들은 체인 형태로 연결되어 있으며, 각 FMU 모델의 입력값 (u)과 출력값 (y)의 관계는 다음의 식으로 표시된다.

$$u(s) = y(s-1), (s = 1, \dots, N) \quad (1)$$

위 공식은 현재 FMU 모델의 입력값이 이전 FMU 모델의 출력값을 사용한다는 것을 의미한다. 즉, FMU2의 입력값은 FMU1의 출력값이 되고 FMU2의 출력값은 FMU3의 입력값이 된다.

IV. 성능 평가를 위한 FMI기반 분산 시뮬레이션 프로그램의 구현

1. HLA/RTI와 DDS를 사용하는 FMI기반의 상호 시뮬레이션 구조

아래의 그림은 본 논문에서 구현한 HLA/RTI와 DDS를 사용하는 분산 연동 시뮬레이션 구조를 보여준다.

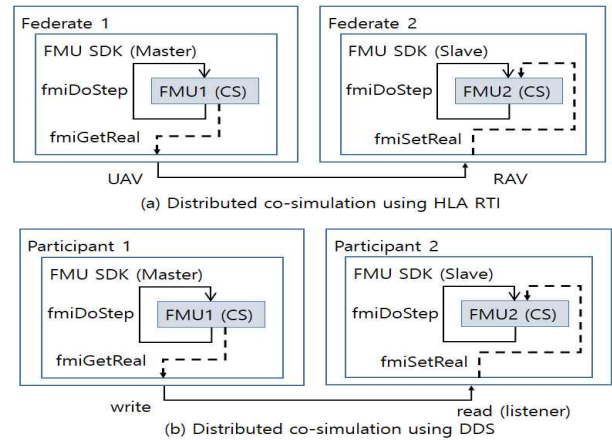


Fig. 5. The distributed co-simulation using HLA/RTI and DDS
 그림 5. HLA/RTI와 DDS를 사용하는 분산 연동 시뮬레이션

먼저 제안하는 구조를 HLA RTI와 DDS를 이용하여 노드를 구성하기 위해 각각 페더레이트(Federate)와 참여자(Participant)를 사용하였다. 다음으로 각각의 Federate와 Participant 프로그램 내부에서는 FMU SDK 시뮬레이션 프로그램을 실행하여 FMI 인터페이스 라이브러리를 사용하여 FMU 모델의 입력값을 셋팅(fmiSetReal)하고, 시뮬레이션을 수행(fmiDoStep)하고, 결과 값을 얻어(fmiGetReal)올 수 있도록 프로그램을 구현하였다.

분산 연동 시뮬레이션을 위해서 HLA/RTI기반 연동의 경우, FMU 모델의 시뮬레이션 결과는 하나의 페더레이트에서 다른 페더레이트로 전달되어지고, DDS기반의 연동의 경우 한 참여자에서 또 다른 참여자로 전달되어진다. 또한, FMU 모델의 시뮬레이션 데이터를 전송하기 위해서 HLA/RTI 경우에는 오브젝트 클래스(Object Class)가 DDS의 경우에는 Topic이 사용된다. 이때 HLA/RTI에서 사용하는 프로그래밍 인터페이스는 UAV(Update Attribute Values)와 RAV(Reflect Attribute Values)이며 DDS의 경우, write와 read함수를 사용한다.

2. 체인 형식으로 연결된 FMU 모델들의 분산 시뮬레이션을 위한 동기화 기법

3장의 그림 4와 같이 두 개 혹은 그 이상의 FMU 모델들이 체인 형식으로 연결되어 분산 시뮬레이션되는 경우, 이전 FMU 모델의 시뮬레이션

결과가 다른 FMU 모델에게 전달되는 경우, 각각의 FMU 시뮬레이션 스텝들은 다른 FMU 시뮬레이션들과 정확히 동기화가 이루어져야 한다. 시뮬레이션이 동기화되지 않을 경우 전체 시뮬레이션 결과는 완전히 잘못될 수 있다.

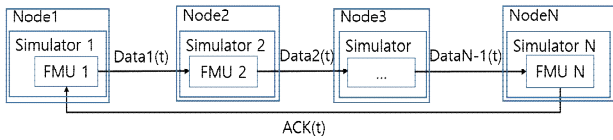


Fig. 6. The distributed co-simulation synchronization using data and Ack message

그림 6. 데이터와 Ack 메시지를 사용하는 분산 시뮬레이션에서의 동기화

이 방법을 사용하면 이전 FMU 시뮬레이션이 수행되고 그 결과가 전송된 후에 다음 FMU 시뮬레이션을 구성하여 전체 FMU 시뮬레이션이 순차적으로 수행될 수 있다. 또한 현재 시뮬레이션 단계가 t 인 경우 그림에 표시된 것처럼 FMU 시뮬레이션은 입력이 있을 때 단계 t 를 시뮬레이션하고 그 결과를 연결된 FMU 시뮬레이션에 전달하게 된다. 그림 6와 같이 연동된 모델의 전체 시뮬레이션을 위해서는 현재 단계 t 에서의 연동시 시뮬레이션에서는 마지막 FMU가 Ack 메시지를 첫 번째 FMU에게 전송함으로써, $t+1$ 단계에서 시뮬레이션이 수행된다. 시뮬레이션 단계가 여전히 전체 시뮬레이션 마지막 단계인 End값보다 작으면 시뮬레이션을 계속 수행하고 만약 End값이 되면 수행을 마치게 된다.

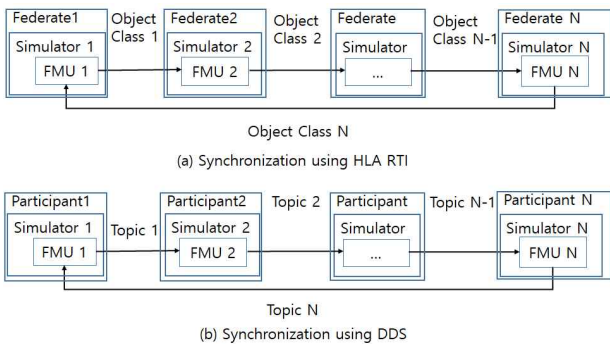


Fig. 7. The synchronization using HLA/RTI and DDS

그림 7. HLA/RTI와 DDS를 이용한 시뮬레이션 동기화

그림7은 HLA/RTI 및 DDS를 사용하여 동기화를 구현한 것을 각각 보여준다. HLA/RTI와 DDS

는 각각 오브젝트 클래스(Object Class)와 Topic의 이름을 기준으로 Publish와 Subscribe를 한다. 또한 HLA RTI와 DDS를 이용하는 분산 시뮬레이션에서 모두 자신의 인접 노드들과만 연결되어 데이터를 송수신하기 위해서는 그림과 같이 i 번째($i=1, \dots, N$) Simulator i 는 Object Class (혹은 Topic) i 를 Publish하고 Object Class (혹은 Topic) $i-1$ (i 가 1인 경우 N 으로 대체) Subscribe해야 한다.

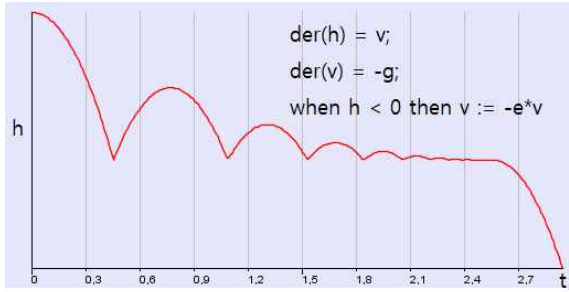
이를 통해서 각 노드들은 전체적으로 체인 혹은 링 형식으로 연결되어 동작하게 되고 각 시뮬레이션 노드들은 이전 노드로부터 자신의 FMU 모델의 시뮬레이션 입력 데이터를 수신(Subscribe)하여 시뮬레이션을 수행하고 자신의 시뮬레이션 데이터의 결과 값을 연결된 다음 노드로 전송(Publish)할 수 있게 된다.

V 시뮬레이션 환경 및 실험결과

본 논문에서 제안한 분산 시뮬레이션 구조를 테스트하기 위해 다음과 같은 시뮬레이션 환경을 사용하였다. 먼저 HLA/RTI와 DDS 구현을 위해서 오픈 소스인 CERTI의 HLA/RTI과 OpenDDS를 사용하였다 [10][11]. 다음으로 FMI기반의 시뮬레이션을 위해서는 FMU SDK라는 오픈 소스 시뮬레이션을 사용하였다 [12].

또한, 본 논문에서는 CPS 모델을 위해 FMU SDK 오픈 소스에서 제공하는 하이브리드 동적 모델 예제인 Bouncingball FMU를 선택하였다. 이 모델은 CPS 시스템에서 많이 볼 수 있는 하이브리드 동역학을 아주 간단히 보여줄 수 있기에 본 논문의 FMU 모델로 선택하였다.

그림 8의 그래프는 FMU모델 1개를 독립적으로 실행했을 때, 시뮬레이션 시간에 따른 공의 높이 (h) 변화를 나타내고, 아래 설명은 이 모델의 입출력 값으로 사용되는 파라미터들을 나타낸다.



h : height [m], used as state
 v : velocity of ball [m/s], used as state
 der(h) : velocity of ball [m/s]
 der(v) : acceleration of ball [m/s]
 g : acceleration of gravity [m/s], a parameter
 e : a dimensionless parameter

Fig. 8. A bouncing ball FMU example (co-simulation type)

그림 8. 바운싱 볼 FMU 예제

분산 시뮬레이션을 위한 파라미터 설정값은 표1과 같다.

Table 1. The simulation parameters
 표 1. 시뮬레이션 파라미터 설정값들

Simulation parameter	Value
FMU model	bouncingBall.fmu (CS Type)
FMU initial value	h=1, e=0.7, g=9.81
Num of FMU	2~7
Start time(sec)	0
End time (sec)	1,2, ... ,10
Step size (sec)	0.1

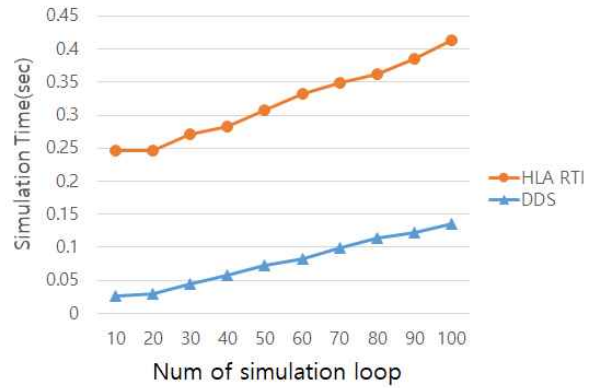
또한, 각 FMU의 시뮬레이션 입력값과 출력값은 bouncingBall FMU의 double 타입의 h(공의 높이)값으로 설정하였다. 즉, 현재의 FMU 모델의 높이(h) 입력값은 이전 노드의 FMU모델 시뮬레이션 결과 값을 사용하고 한 스텝의 시뮬레이션을 수행하고, 현재 FMU 모델의 시뮬레이션 결과 값을 다음 노드에 전달한다.

아래 시뮬레이션 결과는 오픈 소스코드 CERTI HLA/RTI와 OpenDDS를 사용하여 FMU SDK 시뮬레이터를 직접 연결하고 시뮬레이션한 결과이다.

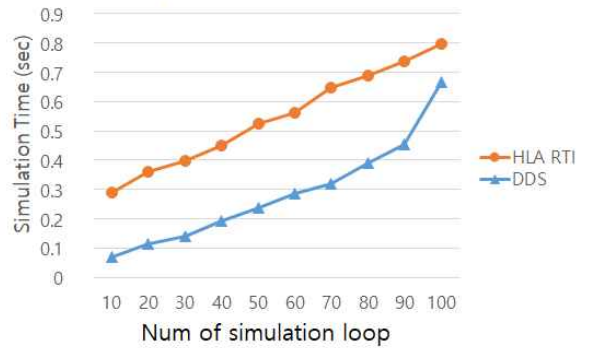
먼저 그림 9의 3개의 그래프는 시뮬레이션 연동

하는 FMU개수를 각각 2, 5, 7개로 설정한 경우에 대해서 각 FMU 시뮬레이션 반복횟수를 증가시키면서 테스트한 결과이다.

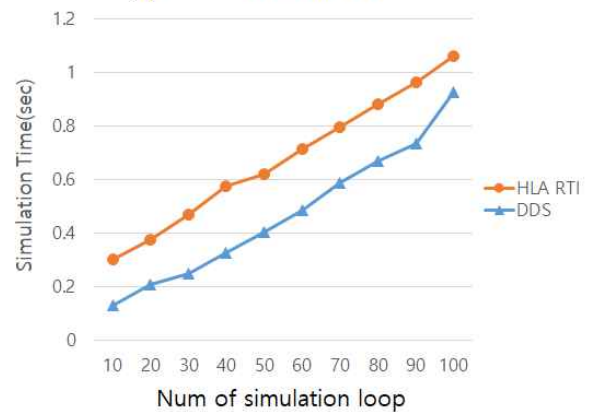
시뮬레이션 반복횟수는 시뮬레이션 종료 시간으로부터 구해질 수 있다. 즉, 시뮬레이션 종료시간이 1부터 10까지 1씩 증가하면, 시뮬레이션 step size가 0.1이므로 시뮬레이션 Loop는 10회에서 100회까지 증가된다.



(a) 2 nodes in simulation



(b) 5 nodes in simulation



(c) 7 nodes in simulation

Fig. 9. Simulation results as the number of loop increases
 그림 9. Loop 회수 증가에 따른 시뮬레이션 측정 결과

이 결과 그래프들을 통해서 DDS를 사용하여 연동 시뮬레이션하는 것이 HLA/RTI를 사용하는 것보다 많이 빠름을 알 수 있다. 또한, 연동하는 FMU 개수가 증가할수록 시뮬레이션 Loop 횟수 증가에 따른 전체 시뮬레이션 시간 증가폭이 큼을 확인할 수 있었다.

이는 제안하는 분산 시뮬레이션 구조가 체인구조로 되어 있고 각 FMU모델에서의 시뮬레이션 결과가 다음 FMU모델에까지 전달되어지는 데이터의 전송속도가 HLA RTI에 비해서 DDS를 사용한 경우가 빠르기 때문이다.

다음으로 그림 10의 결과 그래프들은 각각의 FMU에 대한 시뮬레이션 Loop 횟수가 각각 10, 50, 100회인 경우 FMU의 개수를 증가시키면서 전체 시뮬레이션 시간의 변화를 측정해본 것이다.

이 결과 그래프들을 통해서도 역시 HLA/RTI를 사용하는 것보다 DDS를 사용하는 것이 빠름을 볼 수 있고 시뮬레이션 Loop 횟수가 클수록 연동하는 FMU 개수가 증가할 때 전체 시뮬레이션 증가폭이 큼을 알 수 있었다.

위의 결과들을 통해서 DDS는 HLA/RTI에 비해 전체적으로 시뮬레이션 속도가 전체적으로 빠른 것을 알 수 있고 시뮬레이션 노드와 시뮬레이션 Loop가 증가하더라도 최소 1.14배 이상 빠른 것을 확인할 수 있었다.

VI 결론

CPS 시스템은 일반적으로 복잡한 여러 구성 요소로 이루어져있기 때문에 한번에 하나의 시뮬레이션 프로그램으로 전체 시스템을 시뮬레이션하는 것은 매우 어렵다. 따라서 본 논문에서는 FMI를 기반으로 DDS 미들웨어를 사용하여 체인형식으로 실시간 CPS 모델 시뮬레이션이 연동되는 분산 환경에서의 시뮬레이션 구조를 제안했다.

또한, 오픈 소스코드인 FMU SDK와 CERTI, 그리고 OpenDDS를 사용하여 제안한 시뮬레이션 환경을 구축하여 분산 시뮬레이션을 하였고 전체 시뮬레이션 수행 시간을 측정해보았다.

시뮬레이션 결과를 통해 DDS 기반 통신의 전체 시뮬레이션 속도가 HLA/RTI를 사용하는 것에 비해 최소 1.14배 이상 빠른 것을 확인할 수 있었다. 이것은 DDS 통신 미들웨어가 HLA RTI에 비해 대규모 실시간 데이터 전송에 좋은 성능을 갖기 때문이다.

따라서, CPS 실시간 시뮬레이션 연동이 필요한 HILS와 같은 경우 제안하는 DDS를 사용하는 FMI기반의 분산 시뮬레이션을 사용할 경우 좀 더 실시간 CPS 시뮬레이션 요구사항을 만족하는 환경을 구축할 수 있을 것으로 기대된다.

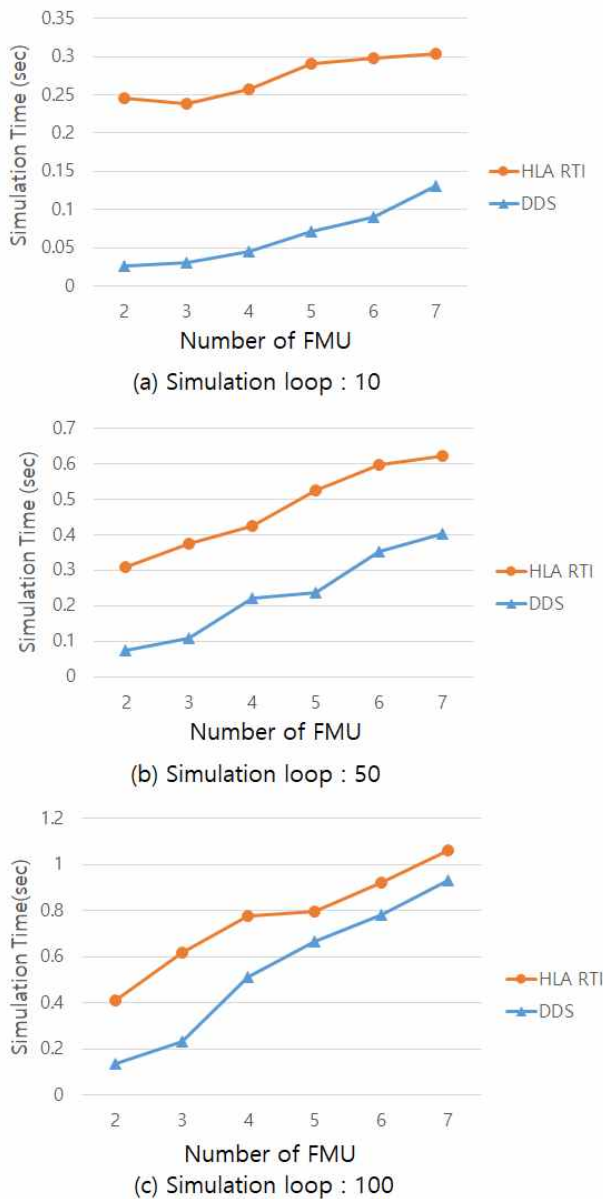


Fig. 10. Simulation results as node number increases
 그림 10. 노드 개수에 따른 시뮬레이션 측정결과

References

- [1] J. A. Sokolowski, C. M. Banks, *Principles of modeling and simulation: a multidisciplinary approach*. John Wiley & Sons, 2011
- [2] J. Banks, *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. Wiley, New York, NY, USA, 1998.
- [3] Functional mock-up interface for model exchange and co-simulation, Version 2.0, Information Tech for European Advancement, Tech. Rep., Modelisar, Aug 2012.
- [4] C. Lina, Z. Qiang, "The Design and Implementation of real-time HILS based on RTX platform" *Industrial Informatics (INDIN), 2012 10th IEEE conference*, 2012, pp276-280
- [5] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification*, IEEE Std. 1516.1-2000, Aug. 2010.DOI: 10.1109/IEEESTD.2010.5557728
- [6] Data Distribution Service for Real-time Systems Specification, Ver 2.2, OMG, 2014.
- [7] J. R. M. Salio, Real Time Simulation with HLA and DDS, Nads, 2012
- [8] S. Chatzivasileiadis, M. Bonvini, J. Matanza, R. Yin, T. S. Noudui, E. C. Kara, R. Parmar, D. Lorenzetti, M. Wetter, S. Kiliccote, "Cyber-physical modeling of distributed resources for distribution system operations", Proc. IEEE, vol. 104, no. 4, pp. 789-806, Apr. 2016.DOI: 10.1109/JPROC.2016.2520738
- [9] A. Garro, A. Falcon, "On the integration of HLA and FMI for supporting interoperability and reusability in distributed simulation," SpringSim-TES/DEVS, 2015, pp. 9-16.
- [10] "CERTI," <https://savannah.nongnu.org/projects/certi>
- [11] "OpenDDS," <http://www.opendds.org>
- [12] "QTronic. FMU SDK (FMU Software DevelopmentKit)," <http://www.qtronic.de/de/fmusk.html>

BIOGRAPHY

Seokjoon Hong (Member)



2006 : BS degree in Information and Communication, Hanyang University
2008 : MS degree in Electronics and Computer Engineering, Hanyang University

2017 : PhD degree Electronics and Computer Engineering, Hanyang University
2008~2011. SW Engineer, Infobank Inc.
2017~ : Post. Doc in Hanyang University

Inwhhee Joe (Member)



1983.2: BS degree in Electronics Engineering, Hanyang University
1995.2: MS degree in Information and Communication, Arizona University

1998.2: PhD degree in Information and Communication, Georgia Tech University
1998~2000 : Oak Ridge National Institute Researcher
2000~2002 : Bellcore Lab (Telcordia) Scientist
2002~ : Professor, Hanyang university

Won-Tae Kim (Member)



1994.2: BS degree in Electronics Engineering, Hanyang University
1996.2: MS degree in Electronics Engineering, Hanyang University
2000.8: PhD degree in Electronics Engineering, Hanyang University

2001.1~2005.2 : CTO, Rostic Technologies Inc.
2005.3~2015.8 : CPS Team Manager, ETRI
2015.9~ : Assistant professor, Koreatech university