한국시뮬레이션학회 논문지
Vol. 17, No. 4, pp. 71-80 (2008. 12)

조선 공정 계획의 수립 완성도 향상을 위한 이산 사건 및 이산 시간 혼합형 시뮬레이션 프레임워크

차주환[1] · 노명일[2†] · 방경운[3] · 이규열[4]

Combined discrete event and discrete time simulation framework for the improvement of shipbuilding process planning

Ju-Hwan Cha· Myung-Il Roh· Kyung-Woon Bang· Kyu-Yeul Lee

ABSTRACT

In this study, a simulation framework, which can support developing various simulation systems for the improvement of process planning in shipbuilding such as the block erection, the block turn-over, and so on, is proposed. In addition, a simulation kernel, which is a key component of the simulation framework, is implemented according to the concept of the combined discrete event and discrete time simulation. To evaluate the efficiency and applicability of the proposed simulation framework, it is applied to the block erection process in shipbuilding. The result shows that the proposed simulation framework can provide the consistent, integrated development environment for a simulation system, as compared with existing studies and commercial simulation systems.

Key words : Modeling and simulation, Simulation framework, Discrete event simulation, Discrete time simulation, Process planning, Shipbuilding

요 약

본 연구에서는 조선 공정 계획의 수립 완성도 향상을 위한 다양한 시뮬레이션 시스템의 개발을 지원하는 시뮬레이션 프레임워크(simulation framework)를 제안하였다. 그리고 제안된 시뮬레이션 프레임워크의 핵심 부분인 시뮬레이션 커널을 구현하였다. 구현된 시뮬레이션 커널은 이산 사건(discrete event)과 이산 시간(discrete time) 시뮬레이션뿐만 아니라 이들이 혼합된 형태의 시뮬레이션도 처리 가능하도록 하였다. 제안된 시뮬레이션 프레임워크의 효율성과 효용성을 평가하기 위해, 이를 조선 분야의 블록 탑재 공정(block erection process)에 적용하였다. 그 결과, 제안된 시뮬레이션 프레임워크가 조선 공정 계획을 위한 다양한 시뮬레이션 시스템을 개발하는데 효과적으로 사용될 수 있음을 확인하였다.

주요어 : 모델링 및 시뮬레이션, 시뮬레이션 프레임워크, 이산 사건 시뮬레이션, 이산 시간 시뮬레이션, 공정 계획, 조선

---

* 본 연구는 a) 서울대학교 해양시스템공학 연구소 및 BK 21 해양기술인력양성사업단 b) 울산대학교 c) 국방과학연구소 수중운동체기술특화센터 SM-11과제 “수중 운동체의 체계/부체계 기능 및 성능 시뮬레이션을 위한 네트워크 기반의 가상(Virtual) 복합 시스템 모델 구조(Architecture) 연구”의 지원과 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2008-314-D00494)을 밝히며, 이에 감사드립니다.

2008년 10월 1일 접수, 2008년 11월 29일 채택

1) 서울대학교 조선해양공학과 대학원
2) 울산대학교 조선해양공학부
3) 대우조선해양(주) 산업기술연구소 생산시스템연구팀
4) 서울대학교 조선해양공학과 및 해양시스템공학연구소
주 저 자 : 차주환
교신저자 : 노명일
E-mail; miroh@ulsan.ac.kr

1. 서 론

1.1 연구 배경 및 필요성

최근 조선 공정 계획(process planning) 분야에서도 사전에 여러 가지 대안을 모의 실험("시뮬레이션(simulation)") 하여 보다 빠른 시간 안에 많은 대안을 검토해 정확한 공정 계획을 수립하고자 노력하고 있다. 이러한 노력의 일환으로, 현업에서는 공정 계획 시뮬레이션을 위한 시스템을 자체적으로 개발하거나 상용 시뮬레이션 시스템을 이용하여 공정 계획을 수립하고 있다. 그러나 이러한 방법은 적용 분야가 달라질 때마다 새로운 시스템의 개발에 따른 시간과 인력의 소비, 기존 시스템의 재사용의 어려움, 상용 시스템의 이용에 따른 현업의 다양한 요구 반영의 어려움 등의 측면에서 효율적이지 못하다. 또한 기존의 설계 및 생산 정보(CAD 정보, 공정 및 일정 계획 정보 등)를 가공 없이 그대로 시뮬레이션에 활용하기가 어렵다. 만약 적용 분야에 구애 받지 않고, 기존의 설계 및 생산 정보를 그대로 활용할 수 있는, 일관되고 통합된 시뮬레이션 시스템의 개발 환경("시뮬레이션 프레임워크(simulation framework")이 개발된다면 조기에 정확한 공정 계획을 수립할 수 있을 것이다.

1.2 관련 연구 현황

최근 시뮬레이션 프레임워크와 관련된 연구가 국내외에서 진행된 바 있다. Praehofer(1991, 2000, 2001)는 이산 사건(discrete event: 특정 사건(event)의 발생에 따라 대상의 상태가 변화하는 것으로 공정 및 물류 시뮬레이션 시스템 등이 이에 해당함)과 이산 시간(discrete time: 일정한 단위 시간의 변화에 따라 대상의 상태가 변화하는 것으로 동역학 시뮬레이션 등이 이에 해당함)이 혼합된 시뮬레이션 대상을 처리할 수 있는 모델링 및 시뮬레이션 방법론을 제안하고, 이를 기반으로 시뮬레이션 프레임워크를 개발하였다. 하지만 개발된 시뮬레이션 프레임워크는 일반적인 공장의 물류 시뮬레이션에 초점을 두어 개발하였기에 조선 공정에 그대로 적용하기가 어려우며, 특히 조선의 기존 설계 및 생산 정보를 시뮬레이션에 활용하기가 어렵다는 단점이 있다.

우종훈(2005)은 상용 시뮬레이션 툴인 QUEST(QUeuing Event Simulation Tool)를 기반으로 한 시뮬레이션 프레임워크를 개발하고 이를 선박 건조 공정에 적용하였다. 이 연구에서는 선박 건조 공정을 제품(production), 공정(process), 설비(resource) 관점으로 분석한 후, 선박 건조 공정을 위한 시뮬레이션 모델을 IDEF(ICAM DEFinition)와 UML (Unified Modeling Language)을 이용하여 정의하였다. 또한 조선소의 기존의 일정 계획 정보를 시뮬레이션에 활용하려는 시도를 하였다. 하지만 개발된 시뮬레이션 프레임워크는 상용 시뮬레이션 툴을 기반으로 하기 때문에 상용 시뮬레이션 툴이 변경될 경우 시뮬레이션 프레임워크를 새로 개발해야 하고, 조선소의 기존의 설계 및 생산 정보 중 하나인 CAD 정보를 시뮬레이션에 활용하기가 어렵다는 단점이 있다.

본 연구에서는 Praehofer가 개발한 시뮬레이션 프레임워크와 유사하게 이산 사건과 이산 시간이 혼합된 시뮬레이션 대상을 처리할 수 있는 조선 공정용 시뮬레이션 프레임워크를 제안하였고, 이를 조선 블록 탑재(block erection) 시뮬레이션에 적용해 보았다. 즉, 제안한 시뮬레이션 프레임워크를 활용해 블록 탑재 공정 시 발생할 수 있는 문제점을 미연에 방지하고자 하는 것이 본 연구의 최종 목표이다.

2. 이산 모델링 및 시뮬레이션

2.1 모델링 및 시뮬레이션의 개념

다음 물음을 통하여 모델링 및 시뮬레이션의 개념을 살펴보자. "A 은행"은 "대기 마을"에 새로운 지점을 개설하려고 한다. "대기 마을 지점"을 방문하는 고객의 평균 대기 시간을 2분 이내로 유지하기 위한 적정한 직원의 수를 결정하고 한다. 가장 간단한 방법은 실제로 지점을 개설한 후, 고객의 평균 대기 시간을 관찰하여 적절한 수의 직원을 배치하는 것이다. 이러한 방법을 통해 도출된 결과는 정확성과 신뢰성이 높으나 결과를 얻기까지 많은 시간과 비용이 소요된다. 또 다른 방법은 개설될 지점과 유사한 가상적인 지점을 만들고 이 가상의 지점을 이용하여 고객의 평균 대기 시간을 관찰하여 적절한 직원의 수를 결정하는 것이다. 후자의 방법은 전자의 방법보다 시간과 비용을 대폭 절감할 수 있다. 그러나 가상의 지점을 실제 지점과 유사하게 만들어야만 현실적인 결과를 얻을 수 있다. 이와 같이 실제의 대상과 유사하게 행동하는 가상의 대상을 "모델(model)"이라 하고, 이러한 모델을 만드는 과정을 "모델링(modeling)"이라고 한다.

모델은 실제 대상을 잘 표현하는 "상태 변수(state variable)", 그 상태 변수를 변경시키는 "외적 및 내적 요인", 그리고 그 요인들에 의해 상태 변수가 변화하는 방법을 정의한 "상태 변경 함수(state transition function)"로 구성된다. 예를 들어 은행의 고객 대기 모델은 고객의 대기 상태를 정의한 상태 변수("대기 고객의 수", "현재 업무 처리 중

인 직원의 수", "현재 직원의 업무 처리 가능 여부"), 고객의 대기 상태를 변화시키는 외적 및 내적 요인("고객 도착", "업무 처리 완료", "대기"), 외적 및 내적 요인에 의해 상태 변수가 변화하는 방법을 정의한 상태 변경 함수로 구성된다.

여기서 모델의 상태 변수를 변경시키는 외적 및 내적 요인을 각각 외부 사건(external event), 내부 사건(internal event)이라고 한다. 이러한 외부 및 내부 사건이 단속적으로 발생하는 것을 이산 사건(discrete event)라고 한다. 그리고 외부 및 내부 사건에 의해 모델의 상태 변수가 변화하는 과정을 도출하는 것, 즉 시간의 변화에 따라 상태 변수의 변화를 계산하는 것을 "시뮬레이션(simulation)"이라고 한다.

2.2 이산 사건 시뮬레이션(discrete event simulation)

그림 1은 앞서 살펴본 예에서, 상태 변수인 "대기 고객의 수"가 외부 및 내부 사건에 따라 변화하는 모습을 나타내고 있다. 대기 고객의 수가 1명일 때 t_0 시간에 새로운 고객이 도착(외부 사건)하면 대기 고객의 수가 2명으로 변경된다. 또한 t_1 시간에 새로운 고객이 도착(외부 사건)하면 대기 고객의 수가 3명으로 변경되고, t_2시간에 고객의 업무 처리가 완료(내부 사건)되면 대기 고객 중 한 명이 업무를 처리하게 되므로 대기 고객의 수는 2명으로 감소한다. 이와 같이 특정한 사건(고객 도착, 업무 처리 완료 등)에 따라 모델의 상태 변수를 변경시키며 시간을 진행하는 특징을 갖는 시뮬레이션을 이산 사건 시뮬레이션이라고 한다(Zeigler, 1990, 2000). 이산 사건 시뮬레이션은 모델의 상태 변수를 변경시키는 사건들을 그 발생 시간 순서대로 나열하여 순차적으로 처리한다. 즉, 사건이 처리될 때 모델의 상태 변수가 변경되고 시뮬레이션 시간이 진행한다.

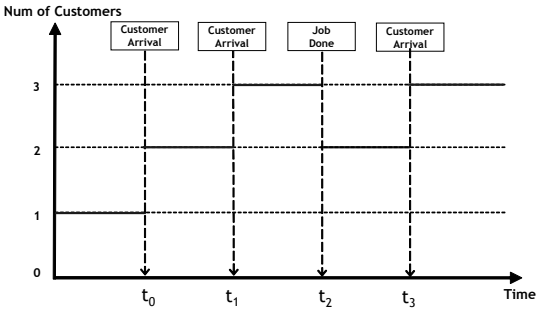

그림 1. Example of the discrete event simulation: bank customer queuing system

이산 사건 시뮬레이션은 실제의 대상과 유사하게 만든 시뮬레이션 모델(simulation model)(간단히 모델이라고도 함)과 사건의 발생 시간 순서에 따라 그 모델의 상태를 변경시켜가며 시뮬레이션을 진행시키는 시뮬레이션 엔진(simulation engine)으로 구성된다.

2.3 이산 시간 시뮬레이션(discrete time simulation)

"A 놀이 동산"은 번지 점프(bungee jump) 놀이 기구를 설치하려고 한다. 번지 점프는 마치 용수철의 진자 운동과 유사하게 작동한다(그림 2 참조). 이 놀이 기구의 운행 시간은 3분으로 한 번에 2명의 고객이 이용할 수 있다. 이용 요금이 고가인 대신 15초에 한 번씩 고객의 표정을 찍어 주는 서비스 전략을 세웠다. 따라서 사진을 잘 찍기 위해서는 15초 마다 놀이 기구의 정확한 위치를 파악하여 고객의 얼굴을 선명하게 찍을 수 있게 카메라를 설치하는 것이 중요하였다. 따라서 "A 놀이 동산"은 매 단위 시간마다 놀이 기구의 정확한 위치를 계산하여 카메라를 설치할 수 있도록 이 문제를 시뮬레이션 하려고 한다. 이처럼 일정한 시간 간격(15초) 마다 대상의 상태(놀이 기구의 위치)를 계산하는 시뮬레이션을 이산 시간 시뮬레이션(discrete time simulation)이라고 한다(Zeigler 등, 2000). 이산 시간 시뮬레이션은 시스템의 상태 변수를 일정한 시간 간격마다 계산하기 때문에 역학 시스템 분석용 시뮬레이션에서 주로 사용하는 방법이다.

2.3 이산 사건과 이산 시간 혼합형 시뮬레이션(combined discrete event and discrete time simulation)

"혼합 음료사"는 하나의 제조 라인에서 하루 1만병의 음료수를 생산하기 위해 적절한 음료 주입기(bottle filling station)를 선정하려고 한다. 즉, 초당 몇 리터(liter)를 주입할 수 있는 음료 주입기를 선택해야 하는가가 주어진 문제이다. 이를 해결하기 위해 "혼합 음료사"는 시뮬레이

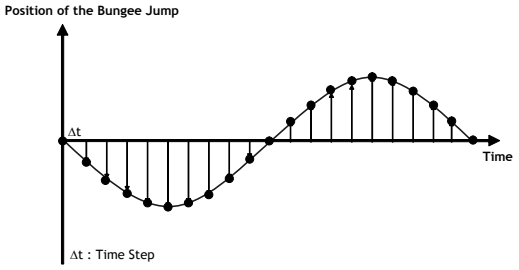

그림 2. Example of the discrete time simulation: bungee jump simulation

션을 이용하려고 한다. 음료수 제조 라인의 작동 과정은 다음과 같다. 음료 주입기는 빈 음료수 병이 들어 왔을 때 (“Bottle Input”), 주입 중인 음료수 병이 없는 상태(“Stop”)이면 들어 온 음료수 병을 채우는 상태(“Filling”)로 변경한다. 그리고 주입 중인 음료수 병이 가득 차게 되면 (“Bottle Full”) 주입 중인 음료수 병이 없는 상태(Stop)로 변경하고 빈 음료수 병이 도착할 때까지 그 상태를 유지한다. 이때 주입 중인 음료수 병이 가득 찼는지의 여부를 확인하는 방법은 매 단위 시간 마다 음료수 병 내의 음료 양이 1.5 리터가 넘는지를 확인하는 것이다. 그림 3은 시간에 따른 음료 주입기의 상태를 표현한 것이다. 그림 3에 나타나 있듯이, 음료 주입기는 “Filling”과 “Stop” 상태를 가지고 있으며, 이 상태는 “Bottle Input” 사건과 “Bottle Full” 사건에 의해 변경된다는 것을 알 수 있다. 여기서 “Bottle Full” 사건은 음료수 병 내의 음료 양을 단위 시간마다 검사하다가 1.5 리터가 되었을 때 발생한다. 이상과 같이 음료 주입기 시뮬레이션은 사건에 따라(빈 음료수 병 입력, 음료수 주입 완료 등) 모델(음료 주입기)의 상태를 변경시키며 시뮬레이션을 진행하며, 모델이 특정 상태(음료 주입기가 음료수 병을 채우는 상태)일 때는 단위 시간 마다 모델의 상태 변수를 계산하여 시뮬레이션을 진행한다. 따라서 이 시뮬레이션은 이산 사건 시뮬레이션과 이산 시간 시뮬레이션이 혼합된 시뮬레이션이라는 것을 알 수 있다. 이러한 시뮬레이션을 이산 사건과 이산 시간 혼합형 시뮬레이션(combined discrete event and discrete time)이라고 한다(Praehofer, 1991, 2000, 2001; Zeigler, 2006).

블록 탑재와 같이 조선 분야의 생산 공정은 특정 사건들을 일련의 순서대로 실행하는 것이다. 예컨대, 블록 탑재를 위해서는 (1) 블록 운송 수단인 트랜스포터(transporter)를 이용하여 해당 블록을 골리앗 크레인(goliath crane) 근처에 옮겨 놓고, (2) 골리앗 크레인을 조종하여 놓여진 블록으로 이동하고, (3) 블록을 골리앗 크레인과 연결하여 들어 올린 후, (4) 블록의 탑재 위치까지 운반하여 내려놓아야 한다. 한편, 골리앗 크레인을 이용하여 블록을 들어 올리고, 탑재 위치까지 이동하여 블록을 내려놓을 때에는 블록 낙하 사고, 블록 간섭(interference) 등의 문제가 빈번히 발생되기 때문에 골리앗 크레인과 블록 사이의 와이어(wire) 장력, 블록의 간섭 여부를 매시간 확인해야 한다. 따라서 블록 탑재 공정과 같이 조선 분야의 공정을 효과적으로 시뮬레이션 하기 위해서는 이산 사건과 이산 시간이 혼합된 시뮬레이션이 필요하다.

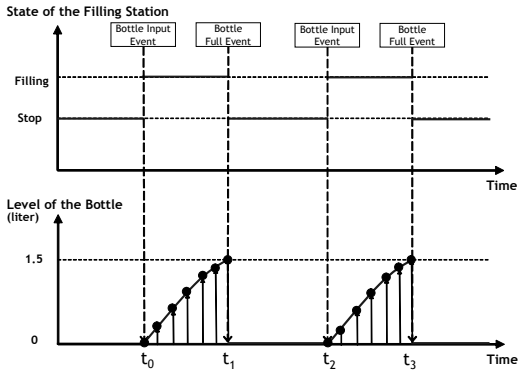

그림 3. the combined discrete event and discrete time simulation: bottle filling station system

3. 이산 사건 시스템과 연속 시스템을 위한 혼합형 시뮬레이션 프레임워크

3.1 시뮬레이션 프레임워크의 구성도

그림 4는 본 연구에서 제안한 이산 사건 시스템과 연속 시스템을 위한 혼합형 시뮬레이션 프레임워크(그림 4의 ①, ②)와 이를 기반으로 하는 시뮬레이션 시스템(그림 4의 ①, ②, ③, ④)의 구성을 나타낸다. 그림 4에 나타나 있듯이, 본 연구에서 제안한 시뮬레이션 프레임워크는 시뮬레이션의 모델링 및 진행 방법을 정의한 시뮬레이션 커널(simulation kernel, 그림 4의 ①)과 이를 이용하여 조

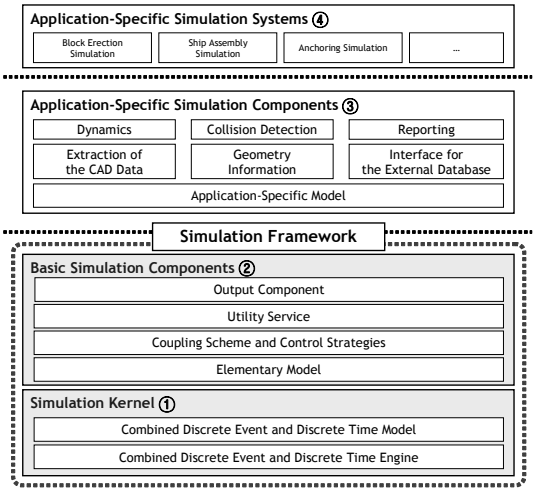

그림 4. Configuration of the combined discrete event and discrete time simulation framework proposed in this study

선 공정 계획용 시뮬레이션 시스템에서 활용 가능한 단위 시뮬레이션 모델을 생성하는 기본 시뮬레이션 요소(basic simulation components, 그림 4의 ②)로 구성된다. 이러한 시뮬레이션 프레임워크를 이용하여 현업에서 필요로 하는 다양한 응용 시뮬레이션 프로그램(application-specific simulation systems, 그림 4의 ④)을 구현할 수 있다. 이러한 응용 시뮬레이션 프로그램은 시뮬레이션에 필요한 응용 시뮬레이션 모델(application-specific model)을 생성하고, 필요에 따라 데이터베이스와 연결하거나 동역학 계산 및 충돌 검사를 할 수 있는 외부 모듈을 추가하는 과정을 통해 구현될 수 있다(application-specific simulation components, 그림 4의 ③). 이들 각각에 대해 보다 상세하게 살펴보면 다음과 같다.

3.1.1 시뮬레이션 커널(simulation kernel)

시뮬레이션 커널은 체계적인 시뮬레이션 모델링 및 진행 방법을 제공하는 것으로 시뮬레이션 모델과 시뮬레이션 엔진으로 구성된다. 시뮬레이션 모델은 계층적 모델 구조를 정의하는 역할하며, 시뮬레이션 엔진은 모델간의 사건들을 관리(event scheduling)하고 처리(event handling)하는 역할을 한다. 본 연구에서는 이산 사건, 이산 시간 및 이들이 혼합된 시뮬레이션에 대해 위와 같은 기능을 수행할 수 있는 시뮬레이션 커널을 구현하였다.

3.1.2 기본 시뮬레이션 요소 (basic simulation components)

기본 시뮬레이션 요소는 시뮬레이션 커널을 이용하여 생성된 기초 시뮬레이션 모델(elementary model), 그 모델을 조합하여 응용 시뮬레이션 모델을 만드는 방법(coupling scheme and control strategies), 시뮬레이션에 필요한 다양한 계산을 할 수 있는 유틸리티 모듈(utility service), 그리고 기본적인 시뮬레이션 결과를 출력해주는 출력 모듈(output component)로 구성된다.

기초 시뮬레이션 모델은 하나의 독립적인 기능을 할 수 있는 단위 시뮬레이션 모델을 말한다. 예컨대, 조선 공정 계획 중 하나인 블록 탑재 공정을 시뮬레이션 한다고 가정하자. 제일 먼저 블록 탑재 공정에서 중요한 역할을 하는 골리앗 크레인을 시뮬레이션 모델로 정의해야 한다. 골리앗 크레인을 모델링 하는 방법은 전체를 하나의 모델로 모델링 하는 방법과 다수의 기초 시뮬레이션 모델로 나누어 모델링 하는 방법이 존재한다. 여기서 후자의 방법이 기초 시뮬레이션 모델을 다른 응용 분야에도 재사용할 수 있다는 점에서 더욱 효율적인 방법이다. 예컨대, 골리앗 크레인의 기능을 블록이 주어졌을 때 이를 들기 위해 해당 블록이 있는 위치로 이동하는 것과 해당 블록을 들어 지정된 위치로 이동시켜 탑재하는 것으로 나눌 수 있다. 따라서 골리앗 크레인에 해당하는 시뮬레이션 모델을 특정 입력에 대해 처리를 위해 준비하는 이동 모델(moving model)과 특정 입력에 의해 정해진 일을 처리하는 처리 모델(process model)의 결합으로 생성할 수 있다. 이때 이동 모델과 처리 모델을 기초 시뮬레이션 모델이라고 한다. 이러한 기초 시뮬레이션 모델을 응용 분야에 적합하게 정의를 한다면, 응용 시뮬레이션 시스템의 구현을 위한 복잡한 응용 시뮬레이션 모델(application-specific model)도 이들 기초 시뮬레이션 모델을 조합하여 생성할 수 있다. 따라서 효과적인 시뮬레이션 시스템의 구현을 위해 기초 시뮬레이션 모델을 조합하여 응용 시뮬레이션 모델을 만드는 방법 역시 필요하다. 또한 시뮬레이션을 위해 기본적인 계산 기능(함수의 적분 기능, 난수(random number)의 생성 방법 등)과 시뮬레이션의 진행 상황 및 결과 등을 사용자에게 보여주기 위해서는 출력 기능이 필요하다.

3.1.3 응용 시뮬레이션 요소 (application-specific simulation components)

최종 결과물인 응용 시뮬레이션 시스템은 시뮬레이션에 필요한 응용 모델을 생성하고, 필요에 따라 데이터베이스와 연결하거나 동역학 계산 및 충돌 검사를 할 수 있는 외부 모듈을 추가하는 과정을 통해 구현될 수 있다. 예컨대, 블록 탑재 시 골리앗 크레인이 블록을 들어 올릴 때 시간에 따른 블록의 움직임을 알고 싶을 경우, 동역학 계산을 할 수 있는 모듈이 필요하다. 또한 골리앗 크레인이 들어 올리는 블록의 무게, 무게 중심 등의 정보를 알기 위해서는 이에 대한 정보가 저장되어 있는 현업의 데이터베이스와 연결하는 모듈이 필요하다. 이 외에도 블록의 형상 정보를 이용하여 충돌 검사를 할 때 형상 정보의 추출을 위한 CAD 모델의 추출 모듈, 형상 정보를 저장할 수 있는 모듈, 형상 정보를 이용하여 충돌 검사를 할 수 있는 모듈 등도 필요하다. 응용 시뮬레이션 요소는 이러한 기능을 제공하고, 응용 시뮬레이션 모델을 이용하여 최종적으로 응용 시뮬레이션 프로그램을 구현할 수 있도록 한다.

3.1.4 응용 시뮬레이션 시스템 (application-specific simulation systems)

시뮬레이션 프레임워크와 응용 시뮬레이션 요소를 이용하여 구현한 시뮬레이션 시스템으로서 실제 사용자들

이 시뮬레이션을 위해 사용할 시스템을 말하며, 블록 탑재 시뮬레이션 시스템 등이 이에 해당한다.

3.2 시뮬레이션 프레임워크의 구현

본 연구에서는 C# 언어를 이용하여 앞서 제안한 시뮬레이션 프레임워크의 프로토타입을 구현하였다(그림 5 참조). 이산 사건과 이산 시간 혼합형 시뮬레이션 커널(그림 5의 ①)을 자체적으로 구현하였고, 기본 시뮬레이션 요소 중 출력 모듈(그림 5의 ②) 역시 구현하였다. 그리고 구현된 시뮬레이션 커널의 성능을 검증하기 위해 조선 블록 탑재 블록 탑재 시뮬레이션 시스템(그림 5의 ⑤)를 구현하였다. 이때 응용 시뮬레이션 요소 중 역학 모듈(그림 5의 ③)과 충돌 검사 모듈(그림 5의 ④)은 공개용 프로그램인 ODE(Open Dynamic Engine)(2008)을 이용하였다.

4. 혼합형 시뮬레이션 프레임워크를 이용한 조선 블록 탑재 시뮬레이션

4.1 블록 탑재 공정

선박과 같은 거대 구조물(재화 중량 300,000톤 대형 유조선의 경우, 길이는 320m, 폭은 60m, 깊이는 30m 정도임)은 자동차와 같이 한 번에 제작할 수 없다. 따라서 다수의 조립 블록(building block, 선체 블록 또는 간단히 블록이라고도 말하며 선박 건조를 위한 요소 단위라고 생각할 수 있음)들로 먼저 분할하여 각각 조립하고(재화 중량 300,000톤 대형 유조선의 경우 약 200개의 블록들로 분할됨), 이 블록들을 다시 여러 개씩 합쳐 보다 큰 탑재 블록(erection block)들을 만들게 된다(재화 중량 300,000톤 대형 유조선의 경우 약 90개의 탑재 블록들로 구성됨). 그 후 도크(dock) 내에서 이 탑재 블록들을 다시 일정한 순서로 탑재하여 하나의 선박을 완성하게 된다. 여기서 탑재란 PE장(Pre-Erection area: 두 개 이상의 조립 블록들을 합쳐 하나의 큰 탑재 블록을 만드는 도크 옆의 공간)에 놓인 탑재 블록들을 골리앗 크레인을 이용, 일정한 순서로 하나씩 들어서 도크 내에서 쌓아 올리는 것을 말한다(그림 6 참조).

이와 같이 블록 탑재 공정은 골리앗 크레인과 도크를 이용하여 최종적으로 선박을 완성한다는 측면에서 선박 건조 공정에서 중요한 의미를 가진다. 하지만 실제 블록 탑재 시에는 건조 공기의 지연을 초래하는 많은 문제가 발생하고 있다. 예컨대 골리앗 크레인이 블록을 들고 이동할 때(골리앗 크레인과 블록은 와이어, 샤클(shackle), 러그(lug)로 연결됨, 그림 7 참조) 블록의 과도한 움직임에 의해 블록이 추락하거나(그림 8의 ①) 후행 탑재되는 블록이 도크 내에서 선행 탑재된 블록과 간섭이 발생(그

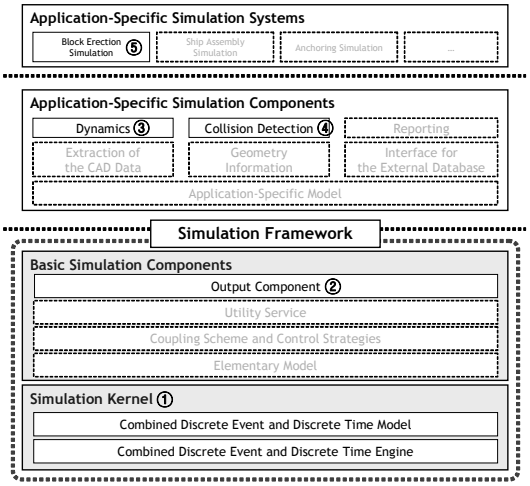

그림 5. Implementation of the prototype of the combined discrete event and discrete time simulation framework

그림 6. Block erection process in shipyards

그림 7. Wire, shackle, and lug for the connection between the goliath crane and the erection block

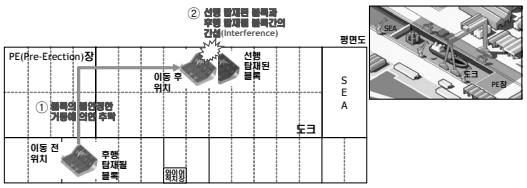

그림 8. Examples of problems during the block erection process

림 8의 ②)하는 등의 문제가 발생하고 있다. 본 연구에서는 전자의 문제를 사전에 방지하기 위한 블록 탑재 시뮬레이션 프로그램을 개발하였다. 즉, 이 프로그램은 골리앗 크레인이 블록을 들고 이동할 때 와이어와 러그 사이의 장력과 각도(angle)를 계속 계산하면서 이 값이 각각 허용 장력(사용되는 와이어의 종류에 따라 다음)과 허용 각도(10도)를 초과하는지를 검사하게 된다.

4.2 블록 탑재 시뮬레이션의 모델링

블록 탑재 시뮬레이션을 위한 시뮬레이션 모델은 그림 9와 같다. “Block Controller” 모델은 블록 탑재 일정을 관리하는 역할을 한다. 즉, 블록 탑재 일정에 따라 블록을 “Crane” 모델에 전달하는 역할을 하고, “Crane” 모델에 의해 탑재가 완료된 블록을 입력 받으며, 다음 탑재 일정에 따라 해당 블록을 “Crane” 모델에 전달하는 역할을 한다. 본 연구에서는 블록 탑재 일정 즉, 블록 탑재 순서가 주어진 것으로 가정하였다. “Crane” 모델은 “Block Controller” 모델로부터 입력된 블록을 탑재하기 위해 해당 블록의 현재 지번(area code, 블록의 이동 전 위치)으로 이동한 후 블록과 연결을 한다. 블록과 연결을 완료한 후 블록의 탑재 지번(블록의 이동 후 위치)까지 이동하고 블록을 탑재 완료한 후 완료된 블록을 “Block Controller” 모델에 보낸다. 또한 블록 탑재 과정 중 와이어와 러그 사이에 작용하는 장력과 와이어와 러그 사이의 각도를 계산하여 이 값이 각각 허용 장력과 허용 각도를 초과했을 때 골리앗 크레인의 이동 가속도를 감소시키는 역할을 한다. 또한 “Block Erection Simulation” 모델은 “Block Controller” 모델과 “Crane” 모델을 결합하여 만든 모델(coupled model)이다. 즉, “Block Controller” 모델의 출력인 “block”을 “Crane” 모델의 입력으로, “Crane” 모델의 출력인 “block”을 “Block Controller” 모델의 입력으로 설정하였다.

아래에서 “Block Controller” 모델과 “Crane” 모델에 대해 보다 상세히 살펴보자.

4.2.1 “Block Controller” 모델

앞서 설명하였듯이, “Block Controller” 모델은 블록 탑재 일정에 따라 블록을 출력하고 탑재가 완료된 블록, 즉 입력된 블록을 처리한다. “Block Controller” 모델은 블록을 출력하는 상태(“Work”)와 탑재가 완료된 블록을 기다리는 상태(“Ready”)로 표현할 수 있다. “Block Controller” 모델은 이산 사건 상태 변수만을 포함하고 있으므로 이처럼 이산 사건 시뮬레이션 모델로 정의되어야 한다. 먼저 “Work”와 “Ready”의 값을 갖는 이산 사건 상태 변수를 정의한다(그림 10의 ①). 그 후, 외부에서 입력이 들어 왔을 때 이산 사건 상태 변수를 변경시키는 외부 변경 함수를 정의한다(그림 10의 ②). “Block Controller” 모델의 외부 변경 함수는 탑재가 완료된 블록이 입력되면, 현재 상태가 블록을 기다리는 상태(“Ready”)일 경우 입력된 블록을 처리하고 현재 상태를 다음에 탑재할 블록을 출력하는 상태(“Work”)로 변경시킨다.

그 후, 시간 진행 함수(time advance function: 다음의 내부 사건이 발생하는 시간을 결정하는 함수)를 정의한다(그림 10의 ③). 즉, 이 함수는 블록 탑재 일정을 검사하며 다음에 탑재할 블록을 입력할 시간으로 시뮬레이션 시간을 진행시키는 역할을 한다. 그 후, 블록 탑재 시간이 되면 블록을 출력할 수 있는 출력 함수를 정의한다(그림 10의 ④). 탑재 블록을 출력한 후 “Block Controller” 모델의 상태를, 탑재가 완료된 블록을 기다리는 상태(“Ready”)로 변경시키는 내부 변경 함수를 정의한다(그림 10의 ⑤). 이 두 함수를 이용하여 탑재할 블록을 “Crane” 모델에 보내고 “Block Controller” 모델은 탑재가 완료되기를 기다린다.

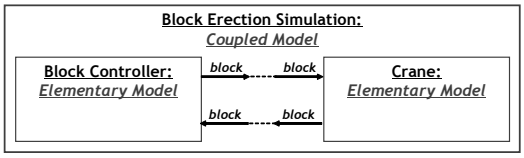

그림 9. Simulation model for the block erection simulation system

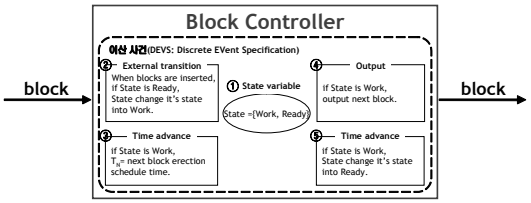

그림 10. Configuration of the “Block Controller” model

4.2.2 “Crane” 모델

“Crane” 모델은 탑재 블록의 입력에 따라 일련의 탑재 과정을 수행(이산 사건)하며 탑재 과정 중 일정 단위 시간마다 와이어와 러그 사이에 작용하는 장력과 와이어와 러그 사이의 각도를 계산(이산 시간)한다. 따라서 이 시뮬레이션 모델은 이산 사건과 이산 시간 혼합형 시뮬레이션 모델로 정의되어야 한다. 먼저 “Crane” 모델이 가질 수 있는 상태로부터 이산 사건 상태 변수와 이산 시간 상태 변수를 결정한다. “Crane” 모델은 그림 11과 같이 탑재할 블록의 입력 대기(“Ready”) 상태, 탑재할 블록과 연결하기 위해 이동하는(“Move to Lifting”) 상태, 블록을 연결한 후 블록을 들어 올리는(“Hoisting Up”) 상태, 블록의 탑재 지번까지 이동하는(“Move with Block”) 상태, 탑재 지번에 도착한 후 블록을 내리는(“Hoisting Down”) 상태, 와이어와 러그 사이의 장력 또는 와이어와 러그 사이의 각도가 허용 값을 초과할 경우 골리앗 크레인의 이동 가속도를 변경하는(“Emergency”) 상태의 6가지의 상태를 갖는다. 그림 11과 같은 이산 사건 상태 변수를 “Crane” 모델에 정의하고(그림 12의 ①), 일정 단위 시간마다 와이어와 러그 사이의 장력, 와이어와 러그 사이의 각도를 나타내는 이산 시간 상태 변수(“Tension”과 “Angle”)를 정의한다(그림 12의 ②). “Crane” 모델은 “Block Controller” 모델로부터 탑재할 블록을 입력 받으면 “Crane” 모델의 상태를 탑재 블록의 입력 대기(“Ready”) 상태에서 탑재 블록과 연결하기 위해 이동하는(“Move to Lifting”) 상태로 변경시킨다. 이러한 과정은 외부 변경 함수에서 처리된다(그림 12의 ③). 탑재 블록과 연결하기 위해 이동하는(“Move to Lifting”) 상태에서 블록을 연결한 후 블록을 들어 올리는(“Hoisting Up”) 상태로 진행되기 위해서는 탑재 블록의 현재 지번까지 골리앗 크레인이 이동하는데 소요되는 시간을 계산해야 한다. 즉, 이동하는데 소요되는 시간 이후에 “Crane” 모델 내부적으로 이동 완료 사건을 처리하여 현재 상태를 변경시킨다. 이는 내부 변경 함수(그림 12의 ④)와 시간 진행 함수(그림 12의 ⑤)를 정의하여 처리할 수 있다. 또한 블록 탑재가 완료되었을 때 탑재가 완료된 블록을 출력하는 출력 함수를 정의한다(그림 12의 ⑥). “Crane” 모델의 6가지 상태 중 탑재할 블록의 입력 대기(“Ready”) 상태, 탑재할 블록과 연결하기 위해 이동하는(“Move to Lifting”) 상태, 블록을 연결한 후 블록을 들어 올리는(“Hoisting Up”) 상태, 블록의 탑재 지번까지 이동하는(“Move with Block”) 상태, 탑재 지번에 도착한 후 블록을 내리는(“Hoisting Down”) 상태는 외부 변경 함수, 내부 변경 함수, 시간 진행 함수를 이용하여 처리할 수 있다. 그러나 와이어와 러그 사이의 장력 또는 와이어와 러그 사이의 각도가 허용 값을 초과할 경우 골리앗 크레인의 이동 가속도를 변경하는(“Emergency”) 상태는 이산 시간 상태 변수의 계산 결과에 영향을 받으므로 이산 시간 상태 변경 함수를 이용하여 처리해야 한다. 따라서 단위 시간마다 이산 시간 상태 변수(“Tension”과 “Angle”)을 계산하고 새로운 사건이 발생하였을 때 이를 처리하는 함수를 정의하였다(그림 12의 ⑦, ⑧, ⑨). 여기서 단위 시간마다 “Tension”과 “Angle”은 동역학 모듈을 이용하여 계산한다.

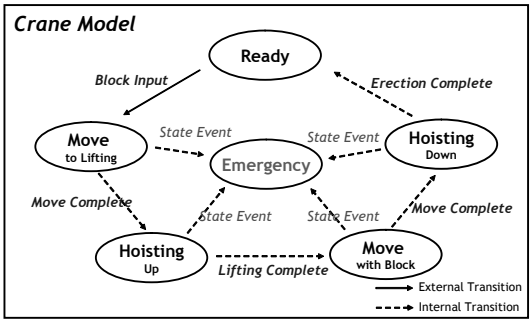

그림 11. State diagram of the “Crane” model

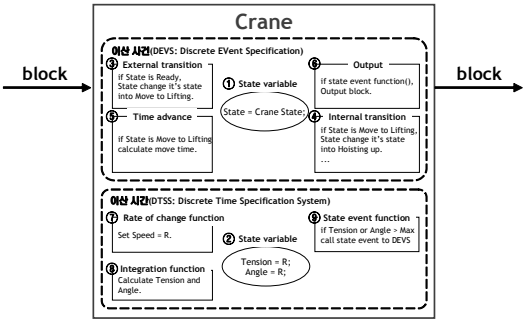

그림 12. Configuration of the “Crane” model

4.3 블록 탑재 시뮬레이션의 결과

그림 13은 본 연구에서 제안한 시뮬레이션 프레임워크를 이용하여 개발한 블록 탑재 시뮬레이션 시스템의 실행 화면을 나타낸다. 그리고 그림 14는 이 시스템을 이용하여 블록 탑재 시뮬레이션을 수행한 결과를 나타낸다. 즉, 블록 탑재가 진행됨에 따라 와이어와 러그 사이에 걸리는 장력과 와이어와 러그 사이의 각도 변화를 각각 나타낸 것이다. 블록 탑재 도중 와이어와 러그 사이에 걸리는 장력이 허용 값을 초과하여 작업자가 골리앗 크레인의 이동

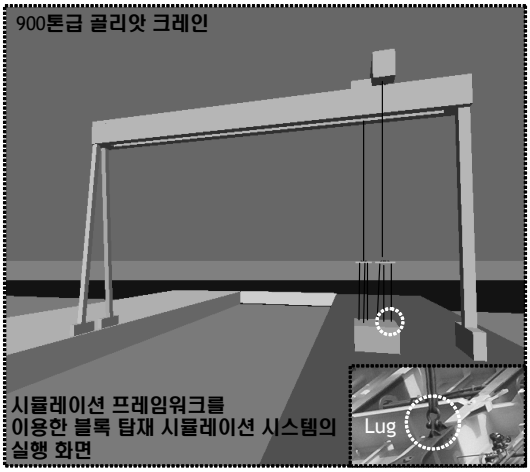

그림 13. Screenshot of the block erection simulation system developed in this study

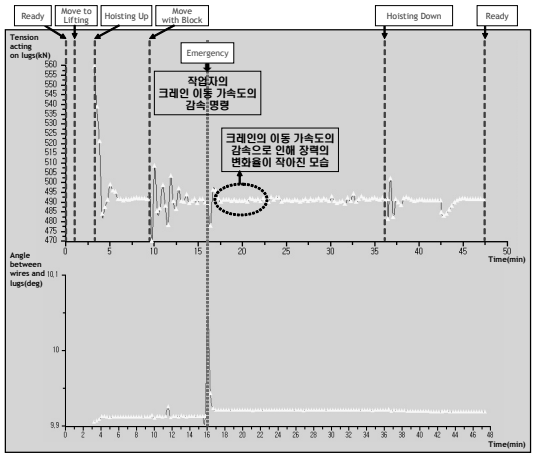

그림 14. Simulation result of the block erection simulation system developed in this study

가속도를 감소시켰고, 그 결과 장력의 변화율이 감소되었음을 알 수 있다.

5. 결론 및 향후 연구 계획

본 연구에서는 조선 공정 계획의 수립 완성도 향상을 위한 다양한 시뮬레이션 요구에 대응하기 위해 시뮬레이션 시스템을 효과적으로 개발할 수 있는 통합 개발 환경인 시뮬레이션 프레임워크를 제안하였고 이의 프로토타입을 구현하였다. 이를 위해 먼저 이산 사건과 이산 시간이 혼합된 시뮬레이션을 처리할 수 있는 시뮬레이션 커널을 구현하였다. 개발된 시뮬레이션 커널의 효용성을 검증하기 위해 기존에 개발된 이산 사건 시뮬레이션을 위한 시뮬레이션 커널과의 성능 비교를 수행하였다. 그 결과 본 연구에서 개발된 시뮬레이션 커널의 성능이 보다 뛰어남을 확인하였다. 마지막으로 본 연구에서 제안한 시뮬레이션 커널의 효용성을 검증하기 위해 이를 조선 블록 탑재 공정에 적용하였다. 그 결과 블록 탑재 공정을 포함한 다양한 조선 공정에 활용 가능함을 확인하였다.

향후에는 본 연구에서 제안한 시뮬레이션 프레임워크의 모든 구성 요소(그림 4 참조)를 구현하고 다양한 조선 공정 분야에 적용하여 시뮬레이션 프레임워크의 효용성을 향상시키고자 한다.

참 고 문 헌

1. 우종훈 (2005), *제품, 공정, 설비와 일정 정보를 통합한 선박 건조 내업 시스템의 모델링 및 시뮬레이션*, 박사학위논문, 서울대학교.
2. Open Dynamic Engine Homepage (2008), http://ode.org
3. Praehofer, H. (1991), *System Theoretic Foundations for Combined Discrete-Continuous System Simulation*, Ph.D Thesis, Johannes Kepler University, Linz, Austria.
4. Praehofer, H., Sametinger, J. and Stritzinger, A. (2001), “Concepts and Architecture of a Simulation Framework based on the JavaBeans Component Model”, *Journal of Future Generation on Computer Systems*, Vol. 17, No. 5, pp. 539-559.
5. Zeigler, B.P. (1990), *Object Oriented Simulation with Modular, Hierarchical Models*, Academic Press.
6. Zeigler, B.P., Praehofer, H. and Kim, T.G. (2000), *Theory of Modeling and Simulation*, 2nd Edition, Academic Press.
7. Zeigler, B.P. (2006), “Embedding DEV&DESS in DEVS: Characteristic Behaviors of Hybrid Models”, *Simulation Series*, Vol. 38, No. 1, pp. 125-132.

차 주 환 (jhcha79@snu.ac.kr)

2002 서울대학교 조선해양공학과 학사
2004 서울대학교 조선해양공학과 석사
2008 서울대학교 조선해양공학과 공학박사
2008～현재 서울대학교 공학연구소 연구원

관심분야 : 모델링&시뮬레이션, CAD, CAM

노 명 일 (miroh@ulsan.ac.kr)

1998 서울대학교 조선해양공학과 학사
2000 서울대학교 조선해양공학과 석사
2005 서울대학교 조선해양공학과 공학박사
2005～2007 서울대학교 공학연구소/해양시스템공학연구소 선임 연구원
2007～현재 울산대학교 조선해양공학부 전임강사

관심분야 : 모델링&시뮬레이션, 선박 설계, CAD, CAGD, CAE, Optimization, CAPP

방 경 운 (kwbang@dsme.co.kr)

2004 서울대학교 조선해양공학과 학사
2006 서울대학교 조선해양공학과 석사
2006～현재 대우조선해양(주) 산업기술연구소 생산시스템연구팀 연구원

관심분야 : 모델링&시뮬레이션, 생산 관리, 생산 계획, CAD, CAM

이 규 열 (kylee@snu.ac.kr)

1975 독일 하노버공과대학교 조선공학 석사
1982 독일 하노버공과대학교 조선공학 박사
1983～1994 한국기계연구원 선박해양공학연구센터, 선박설계, 생산자동화 연구사업(CSDP)단장
1994～현재 서울대학교 조선해양공학과 교수

관심분야 : 모델링&시뮬레이션, 선박 설계, CAD, CAGD, Optimization, CALS