

파이프 핸들링과 트리핑 공정에 사용되는 이동형 시추 장비들을 위한 충돌 방지 시스템용 시뮬레이터 설계

이재용¹ · 구남국¹ · 차주형² · 권기연^{3*}

¹동의대학교 조선해양공학과, ²동의대학교 창의소프트웨어공학부, ³금오공과대학교 산업공학부

Simulator Design for Anti-Collision System of Moving Type Drilling Machines Used in Pipe Handling and Tripping Process

Jaeyong Lee¹, Nam Kuk Koo¹, Joo Hyoung Cha², and Ki-Youn Kwon^{3*}

¹Dept. of Naval Architecture and Ocean Engineering, Dong-eui Univ.

²Division of Creative Software Engineering, Dong-eui Univ.

³School of Industrial Engineering, Kumoh National Institute of Technology

Received 25 March 2020; received in revised form 16 April 2020; accepted 4 May 2020

ABSTRACT

The Anti-Collision System to prevent accidents between moving drilling equipment is an important component of the drilling control system. The system consists of communication module, anti-collision algorithms, override functions and visualization module. To check the validity of the anti-collision algorithm, all systems associated with the system must be activated, which results in a number of inconveniences in the development process. This paper presents a simulator to facilitate the work related to the development of Anti-Collision System. The functions of the drilling equipment related to the pipe connecting and tripping process and the systems linked to the Anti-Collision System in the drilling control system are analyzed to derive the requirements of the simulator. Based on this analysis, the framework of the simulator is constructed and motion of equipment is produced using external input devices. The GUI is used to view the current status of the equipment, and the operation of the Anti-Collision System is verified using commercial visualization S/W. The validity of the developed simulator is verified with several collision scenarios.

Key Words: Anti-collision system, Drilling control network, Human-machine interface, Pipe handling process, Smart machine integrator

1. 서 론

시추 작업장의 파이프 관련 장비 운용에서 가장 기본적인 업무 단위는 스탠드 빌딩(stand building)

과 굴착 공정이라고 할 수 있다. 시추 작업에 소요되는 시간을 절약하기 위해 일반적으로 3~4개의 파이프를 연결하여 하나의 긴 파이프를 만들어 사용하는 데 이를 스탠드(stand)라고 하고, 스탠드를 만드는 공정을 스탠드 빌딩이라고 한다. 일반적으로 시추작업장에는 스탠드 빌딩을 위해 마우스 홀(mouse hole)이 설치되어 스탠드 빌딩 등의 프로

*Corresponding Author, mrkky@kumoh.ac.kr
©2020 Society for Computational Design and Engineering

세스 효율을 높이는 데 사용된다. 굴착 관련 공정에서는 특히 시추 비트를 교환하거나 다른 종류의 파이프를 설치하기 위한 트리핑(tripping) 작업에 많은 종류의 장비가 필요하다.

이러한 파이프 관련 공정들을 진행할 때 여러 대의 장비들이 한정된 공간에서 함께 운용되고, 때에 따라서는 장비들이 위치를 옮겨가며 작업하는 경우도 있다. 장비의 운용은 자동, 반자동 또는 수동의 형태로 진행되는데, 어느 경우에도 장비 사이에 충돌이 있을 수 있다. 또한 장비 대부분이 대형이며 파이프의 연결, 해체, 상승 및 하강 작업이 빠른 속력으로 진행되는 경우가 많아 장비간 충돌이 발생할 때 대형 인명사고로 연결될 수도 있다. 이를 방지하기 위하여 이동하는 장비들의 위치 정보를 바탕으로 충돌을 방지하기 위해 충돌방지시스템(ACS, anti-collision system)이 활용되고 있다^[1].

충돌방지시스템은 시추제어시스템의 일원으로 네트워크에 연결되어 있다. 각각의 장비에서 송신된 제어 관련 신호 중 충돌 관련 데이터를 검사하여 충돌 위험이 감지될 때에 정지 등과 같은 비상 신호를 장비로 보내어 충돌을 방지하는 역할을 수행한다. 국내에서도 시추장비 국산화 노력의 일환으로 장비 운용과 관련한 시뮬레이터가 개발된 바 있다^[2]. 하지만 이는 시추 시스템의 일부인 인간-기계 인터페이스(HMI, human machine interface) 시스템을 구현한 것으로, 시추장비의 운영과 관련한 경험을 절대적으로 필요한 스마트 기계통합시스템(SMI, smart machine integrator) 등과 같은 시스템을 개발하기에는 아직 기술 수준이 부족한 상황이다. 세계 제일의 해양 시추 시스템 업체인 NOV사의 경우 충돌방지시스템이 스마트 기계통합시스템을 구성하는 요소 중 하나이며, 시추제어네트워크(DCN, drilling control network)에 연결되어 사용된다. 이러한 충돌방지시스템은 그 중요성을 감안하여 시추 관련 작업의 최고 책임자 급만 접근 가능하게 되어 있으므로, 아직 국내에는 그 작동 방식이나 알고리즘에 대해 많이 알려져 있지 않은 상황이다.

최근에는 시뮬레이션을 통해 위험상황에 대한 가시화와 운영과 관련한 기술 개발이 이루어지고 있으며^[3], 시추 기반 기술의 확보를 위한 개발에도 적용되고 있다. 충돌방지시스템의 구조와 기능을 분석하고, 시추 시뮬레이터에 연결하여 충돌 감지 및 방지 기술을 구현하고 있다^[4]. 하지만 충돌방지

시스템이 전체 시스템의 일부로 연결되어 있으므로, 업그레이드된 충돌방지시스템의 성능을 검증하기 위해서는 전체 시스템이 모두 가동되어야 하며, 이는 개발 상에 큰 불편함으로 작용하고 있다.

본 논문에서는 충돌방지시스템과 연결되어 충돌 관련 모듈과 기능들을 점검하고 업그레이드하면서도, 별도의 설정 변경없이 변경된 충돌방지시스템이 원래의 시추 제어 시스템에 바로 연결되어 사용될 수 있도록 하는 시뮬레이터를 개발하였다. 장비와 충돌방지시스템 간의 통신 내용과 장비 본연의 기능을 분석하고, 이를 토대로 장비의 움직임 설정하여, 충돌방지시스템의 기능을 검증할 수 있도록 구성하였다. 이를 상용 3D 렌더링(rendering) 프로그램인 언리얼 엔진(Unreal Engine)^[5]을 통하여 시각화함으로써 유효성을 검증하였다.

2. 충돌방지시스템의 구조

2.1 시추 제어시스템의 기본 구조

충돌방지시스템용 시뮬레이터는 시추제어네트워크에 연결된 시스템 중 충돌방지시스템을 제외한 부분에 대한 기능을 수행하여야 한다. 네트워크를 포함한 시추제어시스템의 구성 예를 Fig. 1에 나타내었다. 시추 장비의 제어는 프로그램 가능 논리제어기(PLC, programmable logic controller)를 통하여 이루어지고, 통신 모듈을 통해 네트워크로 접속된다. 일반적으로는 관리의 효율성을 위해 장비 제어를 위한 프로그램 가능 논리제어기들은 같은 장소에 위치하고 있다. 시추 제어와 관련해서 시스템간 통신은 시추제어네트워크를 통해 이루

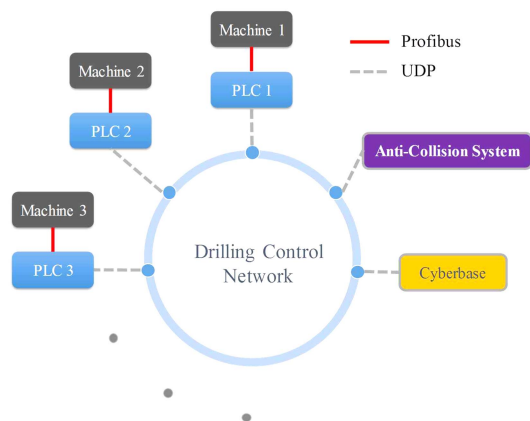


Fig. 1 Topology of drilling control system

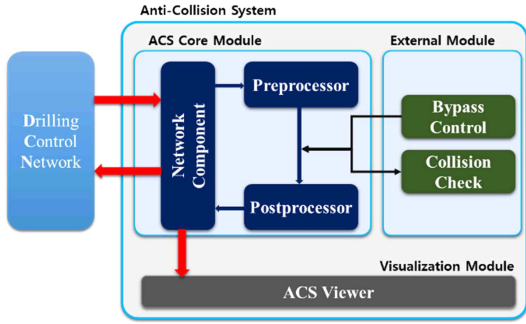


Fig. 2 Basic architecture of ACS

어진다. 장비와 프로그램 가능 논리제어기 사이에는 RS485를 기반으로 한 Profibus FDL과 Profibus DP가 주로 사용되고, 기타 시스템간 네트워크에는 이더넷 기반의 사용자 데이터그램 프로토콜(UDP, user datagram protocol) 통신이 주로 활용된다¹⁶⁾.

사이버베이스(Cyberbase)는 전체 시추 시스템의 운영을 관할하는 시스템으로 인간-기계 인터페이스기반으로 이루어진다. 충돌방지시스템 서버는 기계 시스템과 사이버베이스와의 정보 교환을 통해 충돌을 감지하고, 방지하는 역할을 수행하게 된다.

2.2 충돌방지시스템의 기본 구성

충돌방지시스템의 기본 형태는 Fig. 2와 같다. 시추제어네트워크는 시추 관련 장비와 시스템에서 생성된 데이터가 상호 공유되는 기본 통신망이다. 각각의 장비에서 송신된 데이터는 이 네트워크를 통하여 그 정보를 필요로 하는 시스템으로 전달되고, 다른 시스템에서 보낸 데이터를 수신한

다. 일반적으로 충돌방지시스템은 별도의 하드웨어에 탑재되며, 그 역할의 중대성을 고려하여 작업의 최고 책임자 급만 접근이 가능한 곳에 설치된다.

시추 관련 시스템 운영의 효율성을 위하여 충돌방지시스템은 크게 충돌방지시스템 코어(core) 모듈, 외부 모듈(external module), 그리고 가시화 모듈(visualization module)의 3가지 모듈로 구성되어 있다. 충돌방지시스템 코어 모듈은 통신 요소 위주로 간단히 구성하여 가능한 처리시간을 단축함으로써 다른 시스템이나 장비와의 통신 속도에 영향을 미치지 않도록 구성된다. 특히 상황을 처리하는 바이패스(bypass) 신호 생성 장치 및 알고리즘의 변경 가능성이 충돌 점검(collision check) 컴포넌트의 경우 코어와는 별개의 외부 모듈에 구현함으로써 필요시 해당 항목만 수정하면 되도록 구성하는 것이 필요하다. 충돌은 장비를 감싸는 경계 상자(bonding box)를 생성하여 중첩 부분을 확인하는 것으로 구현하는 방식을 적용한다^{17, 18)}. 또한 가시화 모듈은 다른 모듈과의 정보 교환없이 충돌방지시스템 코어로부터 업데이트된 신호만 수신하게 하여 전체 네트워크의 정보 처리 속도에 영향을 주지 않는 것이 중요하다.

시추 제어 네트워크에서 통신 모듈을 통해 수신된 데이터는 충돌 감지 기능을 수행할 수 있도록 정해진 규약으로 역직렬화(deserialization) 과정을 수행하는 전처리기(preprocessor)를 거쳐 외부 모듈로 보내 지고, 충돌방지 로직의 결과물은 후처리기(postprocessor)를 통하여 정해진 방식으로 직렬화(serialization) 한 후 통신 컴포넌트를 통해 네

Table 1 Scheme for simulator design

| Element | Component | Drilling Control System in the field | Realization in the simulator |
|-----------|------------------|--|---|
| Machines | HW+SW | PLC + Equipment | Emulator design |
| | Identification | IP address + Port nos. | Port nos. |
| DCN | HW | Ring type topology, Fiber optic ethernet | None (internal communication within PC) |
| | Protocol | UDP (ACS↔Cyberbase, ACS↔PLCs, etc) Profibus DP etc. (PLC↔Equipment) | Same as on the left None (combined in the emulators) |
| Cyberbase | HW | Cyber chair + touch screen | Input modules (Joystick + touch screen) |
| | SW | Machine operating SW | Manual operation only |
| ACS | HW | PLC | PC for simulator |
| | SW | ACS core | Same as on the left |
| | External Modules | ACS logic, Bypass modules | Same as on the left |
| | Viewer | 3D visualization SW | Same as on the left |

트위크로 보내 진다.

바이패스 제어(bypass control)는 충돌방지시스템에서 보낸 정지 신호에 의해 정지된 장비를 충돌방지용 로직을 적용하지 않고 이동시켜야 하는 경우에 적용된다. 정비 등의 목적으로 시추작업장의 특정 구역으로 이동하는 경우나 충돌이 일어난 상황에서 두 장비를 분리하는 경우 등이 이에 해당한다. 시추 관련 장비의 충돌방지시스템과 관련해서는 크게 무시(ignore)와 해제(release) 두 가지의 경우로 구분할 수 있다. 무시나 해제 기능은 스마트 기계통합시스템에 있는 스위치 박스의 형태로 구현되며, 해당 스위치가 대기상태에서 켜지므로 변경되면 기능이 활성화되는 방식이다.

무시 신호는 시추작업장에서 특정 장비가 ‘센서 신호 불안정’ 상태에서도 계속 운영될 수 있도록 하는 기능이다. 해당 장비가 다른 장비의 활동영역 점검에서 보이지 않게 만드는 효과가 있다. 즉 어떤 장비의 무시 스위치가 켜진 경우 충돌방지시스템은 해당 장비에서 받은 위치 정보를 무시하고 다른 장비에게 충돌 위험을 알리지 않는 것이다. 따라서 충돌 가능 영역에 해당 장비를 포함한 복수의 장비가 위치하더라도 다른 장비에는 정지 신호를 보내지 않게 된다. 해제 신호는 해당 장비가 충돌방지시스템에 의해 생성되는 다른 모든 장비와의 정지 신호에 반응하지 않도록 만드는 것이다.

2.3 시뮬레이터 구현 방식 설계

충돌방지시스템의 검증을 위한 시뮬레이터를 구현하기 위하여 시추제어시스템의 기능을 적절히 모사하는 요소들이 필요하며, 주요 구현 사항에 대해 Table 1에 나타내었다.

시추 장비들은 프로그램 가능 논리제어기와 세트 구성이 되며 아이피 주소(IP address)와 포트 번호(port no.)를 통하여 네트워크에서 식별된다. 시뮬레이터에서는 각 장비의 움직임을 모사하는 에뮬레이터를 구성하고, 동일 컴퓨터내에서 구동하기 위해 식별은 포트 번호를 사용하도록 한다.

시추제어네트워크에서 장비와 프로그램 가능 논리제어기간의 통신은 에뮬레이터내에서 이루어지므로 별도 구성은 필요하지 않다. 또한 동일 컴퓨터에서 시스템들을 함께 구동하는 것을 고려할 때, 별도의 하드웨어는 필요하지 않게 된다.

사용자와의 인터페이스가 이루어지는 사이버베이스는 각종 조작 장치가 부착된 사이버 의자(cyber





chair)와 부속 터치스크린으로 구성되나 본 시뮬레이터에서는 충돌방지시스템의 기능을 검사하기 위한 용도이므로 조이스틱(joystick)과 터치스크린으로 간략히 구성한다. 또한 조작 장치의 운용과 관련하여 별도의 소프트웨어가 필요하나, 본 연구에서는 수동조작 시나리오에 대해서만 고려하기로 한다.

충돌방지시스템은 실제 시스템과 동일하게 구성하여 시뮬레이터의 결과물이 바로 적용될 수 있도록 하였다.

2.4 시추장비 동작 분석

본 연구에서 고려하고 있는 장비들을 Schlumberger사의 용어 기준^[9]을 적용하여 Table 2에 정리하였다. 탑 드라이브(top drive)는 기존에 사용되던 로터리 테이블을 대체하여 현대의 시추선에 널리 사용되고 있는 시추 파이프 회전 장치이다. 굴착을 위해 파이프를 하강시키거나 파이프를 위로 다시 들어 올리는 트리핑 공정에서도 주 장비로 활용된다. 1~2개 정도의 전기 또는 유압 모터를 사용하고 있으며, 시추 작업을 위해 트래블링 블록(travelling block)에 연결되어 상하로 움직인다. 장비 내부의 백업 통(backup tong)을 이용하여 파이프 체결에 사용되기도 한다. 트래블링 블록을 포함하여 탑 드라이브를 상하로 이동시키는 권양기로 사용되는 것이 드로우워크(drawworks)이다. 충돌과 관련하여

Table 2 Names and basic functions of the drilling equipment in the research

| Equipment | Basic functions |
|---|---|
|  | top drive - carrying pipes up/down - connecting pipes |
|  | racker - transporting pipes - picking up pipes |
|  | iron roughneck - connecting pipes - breaking down pipes |
|  | catwalk - transporting pipes |

여 탐 드라이브, 트래블링 블록, 드로우워크가 함께 작용하는 것이므로, 본 연구에서는 탐드라이브가 다른 두 장비를 포함하는 개념으로 사용하며, 이동 시 상하 운동을 하는 것을 기본으로 한다.

래커(racker)는 수직으로 파이프를 보관하는 랙(rack)과 시추 홀 사이에서 파이프스탠드를 이동시키거나, 캣워크 장비로부터 파이프를 받아 기립하는 작업 등을 수행한다. 따라서 수평 상태의 파이프를 수직으로, 수직 상태의 파이프를 수평으로 변환시키기 위한 상하 이동 기능, 작업 시 전후 좌우 이동 기능, 전방에서 후방으로 회전하는 기능이 필요하다. NOV사에서는 하이드라래커(hydraracker) 등의 이름으로 판매하고 있다.

아이런 러프넥(iron roughneck) 장비는 통(tong)을 사용하여 파이프를 체결하는 데 사용된다. 파이프 체결하던 작업자를 러프넥이라고 부르던 것에서 유래한 이름이며, NOV사에서는 하이드라통(hydratong) 등의 이름으로 판매하고 있다. 통 장비에는 스탠드 체결용(makeup tong)과 해체용(breakout tong)이 있는데, 현대 시추설비에 활용되는 하이드라통은 두 가지 기능 모두 수행 가능하다. 준비 상태에서는 다른 장비의 운용에 방해가 되지 않도록 지정 위치에서 대기하다 필요 시 작업 위치로 이동한다. 장비가 정지하는 작업 위치는 장비 운용의 안전을 위하여 체결이 이루어지는 시추 홀(drilling hole)에서 일정 거리 떨어진 곳에 위치하므로 작업 위치에서 체결 장치를 체결이 이루어지는 위치까지 연장시키는 기능이 필요하다. 또한 체결위치에서 실제 체결이 이루어지는 높이가 때에 따라 다를 수 있으므로 높낮이를 조절할 수 있는 기능도 필요하다.

캣워크(catwalk)는 시추 파이프를 수평으로 보관하는 위치에서 시추 작업을 본격적으로 진행하기 위해 작업장으로 이송할 때 사용되는 장비이다. 장비에 설치된 테일 암(tail-in arm) 장치를 사용하여 래커 등과 함께 파이프 기립 작업을 진행하게 된다. 기본적으로 수평 이동 기능이 필요하다.

3. 충돌방지시스템 시뮬레이터 설계

3.1 장비 이동 관련 특성 분석

장비용 에뮬레이터의 구성을 위해서는 장비가 어떻게 움직이는지를 분석하여 반영하여야 한다. 본 논문에서 사용하고 있는 좌표계를 Fig. 3에 표

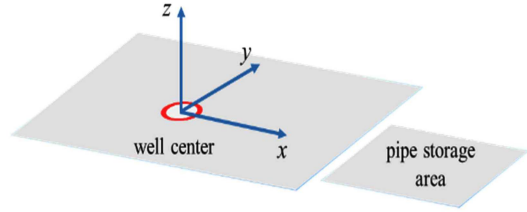


Fig. 3 Defining the coordinate system

시하였다. 일반적으로 시추 작업장에서는 시추 관련 장비의 운영을 주로 담당하는 드릴러(driller)들의 작업실에서 시추 작업장을 바라보는 방향이 y축 방향으로 설정되며, 파이프류는 x축 방향을 따라서 작업장으로 들어오게 된다. Fig. 3은 시추선에서의 일반적인 배치를 나타내는 데, 선수 방향에 파이프를 저장하는 공간이 있다. 이를 바탕으로 작업장에 배치된 장비들의 이동방향을 설정하였다. 서브시스템들의 상호 의존성에 대하여 Fig. 4에 표시하였다. 서브 시스템들은 기본 이동 장비를 바탕으로 하부 장비들이 상부 장비를 따라 움직이게 된다. 탐 드라이브의 경우 트래블링 블록이나 픽업 엘리베이터(pick up elevator) 등의 부속 장비를 통합하여 하나의 장치로 인식하도록 하였다. 이들 부속장비들이 탐 드라이브와 일체로 움직이는 경우가 대부분이기 때문이다. 통 장비의 경우 기본적으로 베이스 프레임(base frame), 툴 프

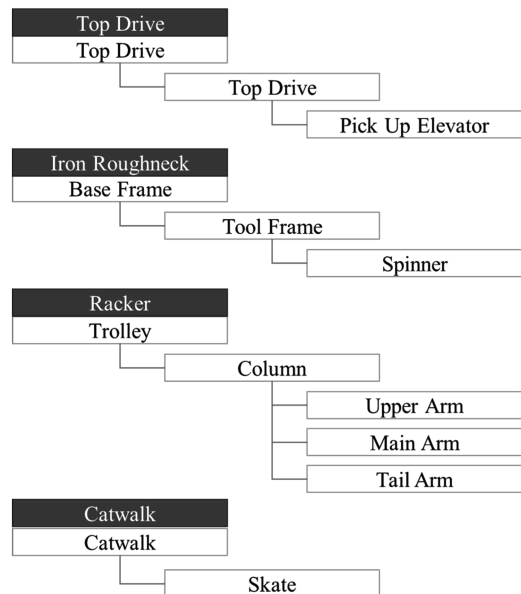


Fig. 4 Dependency between machine subsystems

레이프(tool frame), 그리고 회전력을 전달하는 스피너(spinner)의 3개 서브시스템으로 구성하였다. 실제 장비의 경우 모델에 따라 서브 시스템이 작동하는 방식이 다르므로, 본 논문에서는 툴 프레임과 스피너가 각각 독립적으로 수직방향 움직임이 가능한 것으로 설정하였다. 파이프 체결 및 해체 작업 시 러프넥은 대기상태에서 작업 위치로 움직이고 좌표계에서의 방향은 y축 방향이 된다. 러프넥 장비는 베이스 프레임이 기본 이동 장치로 툴 프레임과 스피너를 함께 이동시키게 되고, 툴 프레임을 이동시키면 스피너도 함께 움직이게 된다. 래커의 경우 트롤리를 기본 이동 장치로 해서 그 위에 컬럼과 3개의 팔(arm)들이 장착되는 것으로 설정하였다. 칼럼은 트롤리 위에서 회전이 가능하며 암들은 칼럼과 함께 회전하게 된다. 3개의 팔들은 각각 독립적으로 확장과 수축이 가능하며 이 작업은 칼럼 기준 좌표계에서 이루어진다. 캣워크는 단일 장비로 스케이트가 파이프를 파이프 저장 위치로부터 작업장까지 이송하므로 x축 방향으로 이동하게 된다. 장비들의 이동 방향과 관련하여 Table 3에 정리하였다.

3.2 충돌방지시스템 시뮬레이터의 구조

충돌방지시스템용 시뮬레이터는 Fig. 5에 나타난 바와 같이 시스템 사양을 입력 받는 사전입력(presetting), 외부 입력 장치로부터 입력을 받아 그 값을 출력하여 보여주는 GUI(graphical user interface), 그리고 기타 설정을 할 수 있는 설정(setting) 부분으로 구성하였다.

사전입력은 시스템 사양을 입력하여 저장한 후 시뮬레이터에서 호출할 수 있도록 제이슨(JSON, JavaScript object notation)파일을 생성하고, 예시 시뮬레이터를 위한 기본 설정을 하는 부분이다. 제이슨은 데이터를 가볍게 만들어 정보 처리속도가 빠르고 언어와 플랫폼에 구애 받지 않아 널리 사용된다^[10]. 사용되는 장비에 대해 필요장비입력(machine)-서브시스템입력(subsystem)-서브시스템 사양(specification)-통신/가용범위(device)의 단계로 입력 값을 받는다. 필요장비입력에서는 시뮬레이션에 사용되는 장비들의 종류를 입력할 수 있게 하였다. 본 연구에서는 Table 3에서 제시한 시추장비들을 토대로 하였으므로 이들을 기본으로 하였으며, 향후 추가할 수 있도록 구성하였다. 서브시스템입력은 서브시스템의 최대 이동 속력 및 가속

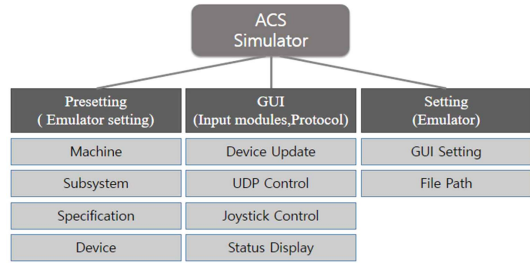


Fig. 5 Components of ACS simulator

Table 3 Summary of moving directions for subsystems

| Equipment | Subsystems | Movement (direction) |
|----------------|------------|-----------------------------|
| Top Drive | Top Drive | Vertical |
| Iron Roughneck | Spinner | Translational (y), Vertical |
| | Tool Frame | Translational(y), Vertical |
| | Base Frame | Translational (y), |
| Racker | Upper Arm | Translational (local x) |
| | Main Arm | Translational (local x) |
| | Tail Arm | Translational (local x) |
| | Column | Rotational |
| | Trolley | Translational (x) |
| Catwalk | Skate | Translational (x) |

도, 최대 회전 속력 및 가속도, 초기 위치 및 회전 상태 등을 입력 받도록 하였다. 통신 및 가용범위에서는 서브시스템별로 할당된 UDP 통신을 위한 개별 포트 번호와 허용 가동 범위를 지정할 수 있도록 하였다.

GUI는 외부 입력장치와의 인터페이스와 관련한 사항을 관할하고, 시스템 작동 상태를 모니터링하는 기능이다. 장치업데이트(device update)에서는 장비 제어를 위한 외부 입력 장치의 변화를 반영하는 부분으로 기존 장비가 아닌 다른 장비로 교체하였을 때 사용된다. UDP 제어(control)에서는 충돌방지시스템과의 통신 방식인 UDP 통신을 시작하고 종료하는 기능을 구현한다. 조이스틱 제어(Joystick control)에서는 조이스틱의 이상 발생 시 GUI 화면을 통해 장비 제어를 직접 수행할 수 있도록 설정하였으며 기타 장비들의 현재 입력 상황을 화면(display)에서 보이도록 하였다.

설정에서는 기본적인 GUI의 환경을 설정하고, 관련 파일의 저장 위치를 지정할 수 있도록 하였으며, 시뮬레이터를 포함한 전체 시스템의 구성은

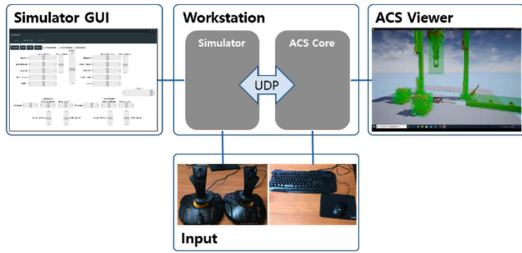
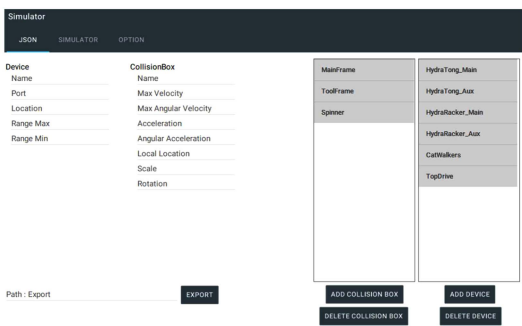


Fig. 6 Structure of the developed simulator and ACS

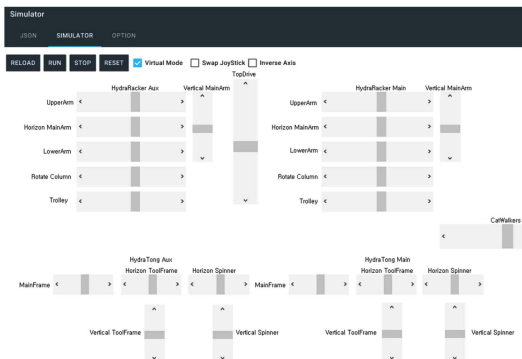
Fig. 6과 같다. 한 대의 워크스테이션에 시뮬레이터와 충돌방지시스템 코어를 함께 구현하였다. 시뮬레이터와 충돌방지시스템 코어 간에는 실제 장비와 충돌방지시스템과의 통신에 사용되는 UDP 방식과 동일한 데이터 형태를 가지도록 구성하여 실제 장비와의 연결 시 추가적인 설정 변경 없이 사용할 수 있도록 구성하였다. 시뮬레이터 상에서 장비의 움직임 입력하기 위한 조이스틱과 뷰어에서 화면을 조정하기 위한 키보드 및 마우스로 입력을 받도록 하였다. 조이스틱을 사용하여 장비에 내리는 명령을 확인하고 오른쪽 화면에서 장비의

운용 상황을 볼 수 있도록 하였다. 충돌방지시스템 코어와 외부 모듈 중 바이패스 제어 컴포넌트는 C#으로 구현하였고, 시뮬레이터는 비주얼 스튜디오 2019커뮤니티를 사용하여 코딩하였다. 시뮬레이터의 사전 입력 컴포넌트와 GUI 컴포넌트는 Fig. 7과 같다.

시뮬레이터와 충돌방지시스템 코어 호환되는 데이터의 형태는 Table 4와 같이 구성하였다. 시뮬레이터에서 충돌방지시스템으로 보내는 데이터는 실제 장비 운영 시 송수신하는 메시지로 구성하였다. 장비 별로 2개의 불(bool)형 데이터는 장비의 센서 시스템에 문제가 있음을 나타내는 ‘Not Healthy’ 신호와 장비의 해제 바이패스 신호를 나타내도록 구성하였다. 충돌과 관련한 기본 데이터의 위치와 자세 관련 정보는 실수형으로 처리하였다. 다른 하나의 불형은 해당 장비의 바이패스 여부를 나타낸다. 탑 드라이브의 경우 수직 운동만 진행하므로 하나의 높이 관련 실수 값으로 표현하였고 러프백은 3개의 세부장치의 위치 데이터를 기반으로 구성하였다. 세부 장치는 직선 운동, 일부 세부 장치는 회전운동을 수행하므로 이를 반영한 데이터 구성을 하였다. 트롤리는 1개의 축 방향 직선 운동만 수행하므로 1개의 위치 값으로 표현하고, 3개의 팔들은 각각 3차원 위치데이터를 포함하도록 하였으며, 회전이 일어나는 칼럼(column)을 위해 회전 값을 포함하였다. 킷워크의 경우 마우스 홀까지의 수평 운동만 고려하는 점을 반영하였다. 장비 에뮬레이터에서 충돌방지시스템으로 보내지는 데이터는 중립 상태에 있는 조이스틱을 움직이면 새로운 값을 생성하게 된다. 조이스틱에 있는 버튼을 사용하여 장비 및 서브시스템을 선택하고, 스틱을 움직이면 해당 변위만큼 비례하여 이동/회전 속력이 생기도록 하여 새로운 위치 및 회전각을 계산한다. 이렇게 생성된 데이터는 장비의 아이피, 포트 번호와 함께 충돌방지시스템으로 보내지고, 충돌 검사가 진행된다. 충돌검사 컴포넌트



(a) Presetting



(b) GUI

Fig. 7 Presetting and GUI components of the simulator

Table 4 Data types and quantities for drilling equipment generated by the simulator and sent to ACS core

| Equipment | Data Types (quantities) |
|----------------|-------------------------|
| Top Drive | Bool (2), Float (1) |
| Iron Roughneck | Bool (2), Float (9) |
| Racker | Bool (2), Float (11) |
| Catwalk | Bool (2), Float (1) |

에서는 장비의 크기에 맞게 경계상자의 크기를 설정하고 경계상자의 위치 및 검사결과를 언리얼로 구현된 뷰어로 전송하여 장비의 상태를 확인할 수 있게 구성하였다.

4. 시뮬레이션

개발된 시뮬레이터의 기능을 점검하기 위한 작업 시나리오를 Table 5에 정리하였다. 에뮬레이터 세팅을 위한 사전 입력의 결과로 생성된 설정 값을 에뮬레이터에서 읽어 들이게 된다. 조이스틱 연결을 통해 운용장비 연결을 확인하고, 통신 연결을 점검하여 충돌방지시스템의 코어와 사용자 데이터그램 프로토콜이 정상 작동하는 것을 확인하였다.

Table 5 Test procedure

| No. | Operation | Simulator check point |
|-----|--|--------------------------|
| 1 | Generating JSON file | Machine emulators |
| 2 | Joystick connection | Operation modules |
| 3 | Check communication | ACS Core, UDP |
| 4 | M1 moves to M2 | Emulators, Input modules |
| 5 | Check collision | ACS logic, Viewer |
| 6 | M2 Ignore switch ON | ACS Bypass module |
| 7 | M2 moves to M1 | Emulators, Input modules |
| 8 | Check collision | ACS logic, Viewer |
| 9 | M2 Release switch ON | ACS Bypass module |
| 10 | Change M1 and M2 and repeat No. 4 to 9 | |

충돌 검사를 위해 임의의 장비 두 대(M1, M2)를 선정하여 충돌이 발생하는 상황을 연출하였으며, 경계상자의 겹침이 발생할 때 정지하는 것을 확인하였다. 바이패스 제어 상황(무시, 해제)에서 충돌방지시스템의 정지 신호에도 불구하고 장비가 움직이는 것을 확인하였으며, 이러한 과정을 충돌 가능성이 있는 장비 조합에 대하여 실시하였다.

시뮬레이션에 사용된 워크스테이션과 기타 장비는 Table 6에 나타내었다. 조이스틱을 조작하여 장비를 이동시키며 작업 시나리오 상에서 충돌이 발생할 수 있는 상황에 대해 테스트를 진행하였다. 조이스틱으로부터 신호를 받아 내부 처리를 하고, 다시 내보내는 작업이 10 ms 이하로 이루어지는 것을 자체 테스트로 확인을 하였으며, 전체 시스템의 통신 요구 속도가 100 ms여서 시간 지연 문제는 발생하지 않았다.

충돌 여부는 각각의 세부 장치 외부를 싸고 있는 경계 상자의 색상이 변하는 것으로 확인할 수 있다. 충돌방지시스템 용 시뮬레이터의 기능을 검사하기 위한 목적이므로 단순한 형태의 경계 상자인 AABB(axis aligned bounding box)와 OBB

Table 6 Specification of simulator H/W and viewer

| | |
|----------|-----------------------------|
| CPU | Intel Xeon E5-2670 @ 2.6GHz |
| RAM | Samsung 32G |
| GPU | NVIDIA Quadro K4000 |
| Joystick | Thrustermaster T 16000M |
| Viewer | Unreal 4.23.3 |

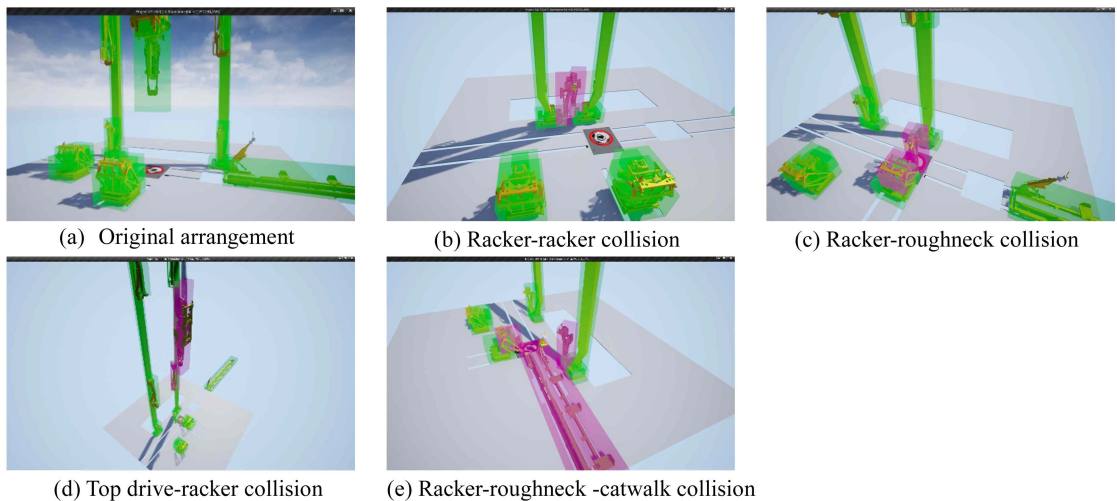


Fig. 8 Collision check simulation

(object oriented bounding box) 타입으로 설정하였다. 회전운동을 래커의 칼럼과 3개의 팔들에 OBB를 설정하고, 나머지 장치들에는 AABB 타입의 경계 상자를 지정하였다.

왼쪽 조이스틱에서 탑 드라이브, 러프넥 1대, 래커 1대 등 좌측 3개 장비를, 오른쪽 조이스틱에서 래커 1대, 러프넥 1대, 캣워크 등 우측 3개 장비를 제어하도록 하였고, 각 장비별로 서브시스템도 선택하여 작동 가능하도록 하였는데, 이는 실제 시스템의 작동 방식을 고려한 것이다.

충돌 검사는 래커-래커 충돌, 래커-통 충돌, 탑 드라이브-래커 충돌 상황을 비롯하여 3개 장비의 동시 접촉 상황을 위해 통-래커-캣워크 충돌 상황에 대해 진행하였다. 시뮬레이션 결과는 Fig. 8에 나타내었다. Fig. 8(a)에서와 같이 초기에 경계 상자를 모두 녹색으로 설정한 상태에서 시뮬레이터를 통해 장비를 조작하여 충돌 감지 시 색상을 변화시키고, 장비를 멈추게 하는 충돌방지시스템의 동작이 정상적으로 작동하는 것을 확인하였다.

5. 결 론

본 연구에서는 시추 작업에 사용되는 충돌 방지 시스템의 개발을 위한 시뮬레이터를 구성하였다. 자동차나 기타 이동형 시스템에 사용되는 시스템과는 달리 본 연구에서의 충돌방지시스템은 시추 작업에서의 스탠드 빌딩과 트리핑 공정에 사용되는 이동형 장비들의 상호 충돌을 감지하고 방지하는 것을 목표로 하는 시스템을 고려하였다.

통신모듈, 데이터 처리 모듈을 비롯하여 충돌감지로직과 바이패스 등의 외부 모듈로 구성되는 충돌방지시스템을 실제 시스템에 연동하기 전 개발 및 업그레이드 단계에서 각 모듈들의 기능을 검사하기 위한 목적으로 시뮬레이터를 개발하였다.

각 장비들의 기본 사양을 입력하고, 조이스틱을 사용하여 각 장비들의 이동시키는 기능을 구현하였다. 충돌방지시스템 코어와의 UDP 통신을 실제 하드웨어 연결과 동일하게 설정하여, 시뮬레이터로 검증된 충돌방지시스템에 실제 장비 연결 시 별도의 추가적인 세팅이 없이 바로 동작하도록 구성하였다. 복수의 장비를 동시에 작동하여 충돌 상황을 연출하였으며 각 상황에 대하여 충돌 감지가 정상적으로 이루어지고, 그 결과가 뷰어에 나타나는 것을 확인하였다.

시추시스템의 국산화가 진행됨에 따라 운영시스템 등에 대한 기술개발이 이루어져 왔으나, 안전과 관련한 시스템은 경험부족으로 진전이 더딘 상황이다. 기자재에 설계 기술 등을 확보할 경우 보다 정확한 시스템의 기능을 파악하고, 시뮬레이터에 추가할 수 있어 HILS(hardware in the loop simulation) 기술로 확대할 수 있는 기반 기술을 확보하였으며, 본 연구 결과를 바탕으로 충돌 방지 시스템 등 시추 지원 기술 개발을 활성화시킬 수 있을 것으로 기대된다.

감사의 글

이 논문은 산업통상자원부 산업기술혁신사업(과제번호: 10070163) “해양시추장비 통합 운용제어 시스템 및 HILS 기반 검증시스템 개발” 과제의 지원으로 수행된 연구임.

References

- Jiang, Z.Q., 2014, Anti-Collision Control Research on Ocean Drilling Platform Drill String Automated Processing, *Advanced Materials Research*, 945, pp.2665-2669.
- Park, G.P., Lee, J.B., Ham, S.H. and Jo, A.R., 2012, Development of Simulator for Drilling and Well Control, *Proceedings for the SNAK Annual Autumn Meeting*, Changwon, KOREA, pp.884-885.
- Koo, M.W., Ha, S., Cha, J.H. and Bae, J.H., 2019, Operator Training Simulation for Fire Incident in Ocean Environment Based on Virtual Reality, *Korean Journal of Computational Design and Engineering*, 24(4), pp.431-441.
- Moon, G.H. and Lee, J., 2018, Analysis of the Design Requirements for Anti-Collision Systems in Offshore Drilling Units, *Proceedings KSME Spring Conference*, Jeju, Korea, pp.59-60.
- Epic Games, Inc., <https://www.unrealengine.com/>, accessed Mar. 24, 2020.
- IADC Advanced Rig Technology Committee, *Rig Communication Protocol List Rev. 601*, <https://www.iadc.org/wp-content/uploads/Rig-Communication-Protocol-List-Rev-601.pdf>, accessed Mar. 24, 2020.
- Sagardia, M., Stouraitis, T. and Lopes e Silva, J., 2014, A New Fast and Robust Collision Detection and Force Computation Algorithm Applied to the Physics Engine Bullet: Method,

- Integration, and Evaluation, *Proceedings Conference and Exhibition of the European Association of Virtual and Augmented Reality*, Bremen, Germany, pp.59-60.
8. Schneider, D., Schömer, E. and Wolpert, N., 2017, Collision Detection for 3D Rigid Body Motion Planning with Narrow Passages, *Proceedings IEEE International Conference on Robotics and Automation*, Singapore, Singapore, pp.4365-4370.
 9. Schlumberger Limited, Schlumberger Oilfield Glossary, <https://www.glossary.oilfield.slb.com/>, accessed Mar. 24, 2020.
 10. Yun, G.Y. and Lee, Y.G., 2016, Service-oriented Web for Transformable 4D Printing Components, *Korean Journal of Computational Design and Engineering*, 21(2), pp.122-129.



이 재 용

1992년~2000년 부산대학교 기계공학부 학사
 2000년~2002년 Texas A&M University 기계공학과 석사
 2002년~2007년 Texas A&M University 기계공학과 박사
 2007년~2012년 삼성중공업 대덕 연구센터 책임연구원
 2012년~현재 동의대학교 조선해양공학과 교수
 관심분야: Ocean system dynamics and control, Offshore drilling system simulation, Dynamic positioning of offshore vessels
 이메일: jlee@deu.ac.kr



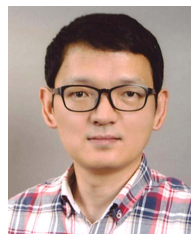
구 남 국

1999년~2004년 서울대학교 조선해양공학과 학사
 2004년~2012년 서울대학교 조선해양공학과 박사
 2014년~현재 동의대학교 조선해양공학과 교수
 관심분야: Multibody dynamics, Machine Learning, Ship Stability
 이메일: knk80@deu.ac.kr



차 주 형

2018년~현재 동의대학교 응용소프트웨어공학과 학부과정
 관심분야: AI, 유전자 알고리즘, 딥러닝, 3D 그래픽
 이메일: aoikazto@naver.com



권 기 연

1995년~1999년 고려대학교 기계공학과 학사
 1999년~2001년 고려대학교 기계공학과 석사
 2001년~2006년 한국과학기술원 기계공학과 박사
 2006년~2016년 삼성중공업 중앙연구소 수석연구원
 2016년~현재 금오공과대학교 산업공학부 교수
 관심분야: CAD/CAE, Digital manufacturing, Mesh generation, Dimensional quality management, 3D measurement system
 이메일: mrkky@kumoh.ac.kr