

Design and Implementation of Multi-HILS based Robot Testbed to Support Software Validation of Biomimetic Robots

Hanjin Kim[†] · Kwanhyeok Kim^{††} · Beomsu Ha^{†††} · Joo Young Kim^{††††} ·
Sung Jun Shim^{†††††} · Jee Hoon Koo^{††††††} · Won-Tae Kim^{†††††††}

ABSTRACT

Biomimetic robots, which emulate characteristics of biological entities such as birds or insects, have the potential to offer a tactical advantage in surveillance and reconnaissance in future battlefields. To effectively utilize these robots, it is essential to develop technologies that emulate the wing flapping of birds or the movements of cockroaches. However, this effort is complicated by the challenges associated with securing the necessary hardware and the complexities involved in software development and validation processes. In this paper, we presents the design and implementation of a multi-HILS based biomimic robot software validation testbed using modeling and simulation (M&S). By employing this testbed, developers can overcome the absence of hardware, simulate future battlefield scenarios, and conduct software development and testing. However, the multi-HILS based testbed may experience inter-device communication delays as the number of test robots increases, significantly affecting the reliability of simulation results. To address this issue, we propose the data distribution service priority (DDSP), a priority-based middleware. DDSP demonstrates an average delay reduction of 1.95 ms compared to the existing DDS, ensuring the required data transmission quality for the testbed.

Keywords : Biomimetic Robot, M&S, HILS, Robot Testbed

생체모방로봇 소프트웨어 검증 지원 다중 HILS 기반 로봇 테스트베드 설계 및 구현

김 한 진[†] · 김 관 혁^{††} · 하 범 수^{†††} · 김 주 영^{††††} · 심 성 준^{†††††} · 구 지 훈^{††††††} · 김 원 태^{†††††††}

요 약

생체모방로봇은 조류나 곤충과 같은 생체의 특성을 모방하여 미래 전장에서 은밀한 감시와 정찰 작업에 큰 역할을 할 것으로 기대된다. 이 로봇들의 효과적인 활용을 위해서는 새의 날갯짓이나 바퀴벌레의 움직임 등을 모방하는 기술이 중요하지만, 이를 지원하는 하드웨어 확보와 소프트웨어 개발 및 검증 과정의 복잡성으로 인해 어려움이 따른다. 본 논문에서는 모델링 및 시뮬레이션(M&S) 기술을 적용한 다중 HILS 기반 생체모방로봇 소프트웨어 검증 테스트베드를 설계하고 구현한 결과를 소개한다. 테스트베드를 사용함으로써 개발자들은 하드웨어 부재 문제를 극복하고, 미래 전장 시나리오를 시뮬레이션하며 소프트웨어 개발과 테스트를 수행할 수 있다. 그러나, 다중 HILS 기반 테스트베드는 테스트 대상 로봇 수의 증가에 따른 장치 간 연동 지연 문제를 경험할 수 있으며, 이는 시뮬레이션 결과의 신뢰도에 크게 영향을 미칠 수 있다. 이를 해결하기 위해, 우리는 우선순위 기반 미들웨어인 data distribution service priority (DDSP)를 추가로 제안한다. DDSP는 기존 DDS 대비 1.95 ms의 평균 지연 감소 효과를 보이며, 테스트베드에서 요구되는 데이터 전송 품질을 보장할 수 있음을 입증하였다.

키워드 : 생체모방로봇, 모델링 및 시뮬레이션, HILS, 로봇 테스트베드

※ 이 논문은 국방신속획득기술연구원(방위산업기술지원센터)의 지원 (사업명: 초소형 생체모방로봇용SW프레임워크 기술개발, 계약번호: UC2000010D)의 지원하에 수행되었음.

† 준 회원 : 한국기술교육대학교 컴퓨터공학과 박사과정

†† 준 회원 : 한국기술교육대학교 컴퓨터공학과 석사

††† 비 회원 : 한국기술교육대학교 컴퓨터공학과 석사

†††† 정 회원 : LIG넥스원무인체계연구소 수석연구원

††††† 비 회원 : LIG넥스원 무인체계연구소 수석연구원

†††††† 비 회원 : LIG넥스원 무인체계연구소 선임연구원

††††††† 정 회원 : 한국기술교육대학교 컴퓨터공학부 교수

Manuscript Received : April 3, 2024

Accepted : May 14, 2024

* Corresponding Author : Won-Tae Kim(wtkim@koreatech.ac.kr)

1. 서 론

다가올 미래, 국방 과학기술의 발전에 따라 전장의 양상은 현재와 매우 다를 것으로 예상된다. 사람들 간의 총격전이 주를 이루던 기존의 전장과는 달리, 사이버전, 신무기전 등으로 전쟁 수단이 다양화되고 고도화될 전망이다[1]. 이미 우크라이나 전쟁과 이·팔 전쟁에서 드론 같은 새로운 형태의 무기체계가 사용된 바 있다 [2]. 이처럼 변화하는 전장 환경에서 생체모방로봇은 미래전의 판도를 변화시킬 중요한 기술로 부상

하고 있다. 조류, 곤충 등 생체의 특성을 모방한 이 로봇들은 은밀한 기동을 통해 감시 및 정찰에 우위를 점할 수 있는 잠재력을 지니고 있다.

생체모방로봇의 효과적 활용을 위해선, 그들의 특성을 적절히 이용하는 것이 중요하다. 예를 들어, 조류의 날갯짓[3]이나 바퀴벌레의 움직임[4] 같은 특징을 활용하면 은밀한 비행이나 좁은 공간 탐색과 같은 임무를 수행할 때 감시 및 정찰에서 우위를 점할 수 있다. 하지만 이러한 로봇의 새로운 기능과 특성은 기술 개발과 검증을 어렵게 만든다.

바퀴벌레의 움직임이나 새의 날갯짓과 같은 생체의 특성을 모방한 로봇의 하드웨어를 구현하는 것은 비용과 시간이 많이 들며, 바퀴가 달린 로봇이나 로봇팔과 달리 레퍼런스 로봇을 구하기가 어렵다. 소프트웨어 개발자 관점에서 생체모방로봇은 임무 달성을 위한 임무 소프트웨어와 더불어 생체모방로봇의 구동을 위한 특화된 소프트웨어의 개발이 요구된다[5, 6]. 이런 상황에서 개발 대상 하드웨어의 부재는 관련 소프트웨어 개발의 진입장벽임과 동시에 개발된 소프트웨어의 품질에도 문제가 될 수 있다. 게다가, 임무 수행의 시너지를 위해 단독 운용뿐만 아니라 다양한 유형의 로봇의 복합적인 운용도 고려한다면 로봇 소프트웨어의 개발과 검증 과정은 더욱 복잡해질 수 있다.

언급한 문제의 상황의 해결책 중 하나로 모델링 및 시뮬레이션(M&S) 기술을 고려해 볼 수 있다[7, 8]. 다양한 유형의 생체모방로봇을 설계 프로그램을 통해 모델링하고, 시뮬레이션 소프트웨어를 활용하여 이들의 특징적 기능을 구현함으로써, 실제 로봇 하드웨어의 확보 문제를 해소할 수 있다. 또한, 다양한 미래 전장 상황을 시뮬레이션하여 임무 소프트웨어 개발과 테스트를 지원할 수 있다. 게다가, hardware-in-the-loop simulation (HILS)를 활용하면 타겟 임베디드 보드를 기반한 시뮬레이션 테스트베드를 구축할 수 있어, 생체모방로봇의 구동에 관련된 소프트웨어의 기능을 함께 개발, 테스트할 수 있다[9]. 단, HILS 기술을 테스트베드에 적용하면서 시뮬레이션 환경과 검증 대상 소프트웨어가 구동되는 장치 간의 물리적 거리로 인해 지연 문제가 발생할 수 있으므로, 이에 대한 대응 방안을 테스트베드 설계 및 구현 시 고려해야 한다.

본 논문에서는 M&S 기술의 장점을 바탕으로 다중 HILS 기반 생체모방로봇 운용 검증 지원 테스트베드를 설계, 구현하였다. 논문의 2절에서는 설계한 테스트베드를 살펴보고 다중 HILS 환경에서 생체모방로봇을 검증하기 위한 연동 요구사항을 논의한다. 3절에서는 설계를 바탕으로 구현된 테스트베드를 살펴보고, 4절에서는 2절에서 논의된 연동 요구사항을 충족시키기 위해 개발된 우선순위 기반의 연동 미들웨어와 실험 결과를 소개한다.

2. 테스트베드 설계

본 논문에서 설계한 다중 HILS 기반 테스트베드는 Fig. 1과

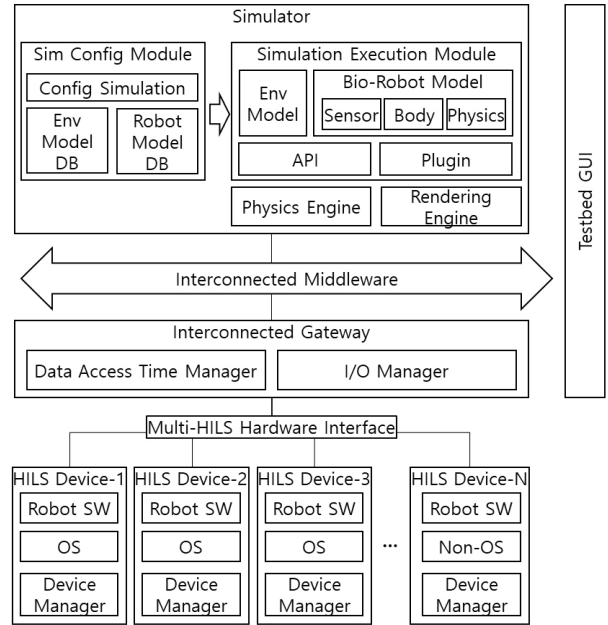


Fig. 1. Design of Multi-HILS based Biomimetic Robot Testbed

같다. 테스트베드는 시뮬레이터, 테스트베드 graphical user interface (GUI), 연동 게이트웨이, 다중 HILS 장치로 구성된다. 가상 시뮬레이션 환경에서 생체모방로봇 모델은 임무 시나리오에 따라 환경과 상호작용하며 센서 데이터를 생성한다. 이 데이터는 HILS 장치로 전송되며, HILS 장치에 탑재된 검증 대상 로봇 소프트웨어(SW)는 센서 데이터를 받아 제어 신호를 생성한다. 이 제어 신호는 다시 시뮬레이터의 가상 로봇 모델로 피드백되며 이러한 과정을 바탕으로 소프트웨어의 검증을 수행한다.

2.1 테스트베드 소프트웨어 구조 설계

앞서 언급한 바와 같이 테스트베드는 크게 시뮬레이터, 게이트웨이, 다수의 HILS 장치로 구성된다. 시뮬레이터는 별도로 개발하거나 상용과 오픈소스를 고려해 볼 수 있다. 본 논문에서는 Gazebo 시뮬레이터[10]를 고려하였다. 따라서 테스트베드 소프트웨어의 기본 구조는 Gazebo의 소프트웨어 구조를 따른다.

시뮬레이터에는 시뮬레이션 실행을 위한 시뮬레이션 실행 모듈과 시뮬레이션 구성 모듈이 있다. 시뮬레이션 실행 모듈에서는 application programming interface (API)를 통해 물리 엔진과 렌더링 엔진의 도움을 받아 로봇 운용 검증을 위한 시뮬레이션을 실행한다. 플러그인은 시뮬레이터(Gazebo) 기능에 접근하기 위한 코드로서 환경 모델 및 생체모방로봇 모델이 시뮬레이터 기능에 접근하기 위해 사용한다. 시뮬레이션 구성 모듈에서는 환경 모델 DB와 로봇 모델 DB에 존재하는 환경 모델과 로봇 모델을 load하여 시뮬레이션 시나리오를 설정한다. 설정된 시나리오에 따라 시뮬레이션 실행 모듈에서 시뮬레이션을 실행한다.

게이트웨이는 시뮬레이터와 다수의 HILS 장치 간 연결될 때 데이터 Access를 관리한다. 가상 환경의 로봇 모델은 각각 구성된 가상 센서 모델이 서로 다르며, 각 로봇이 시나리오에 따라 운용되며 생성하는 센서 데이터는 대응되는 HILS 장치로 전달되어야 한다. 데이터는 적시에 전달되어야 검증의 신뢰성이 유지될 수 있다. 이 과정의 효율적인 관리를 위해 데이터 Access 시간 관리기는 데이터의 접근 시간을 관리하며, I/O 관리기는 각 가상 로봇 모델과 대응되는 HILS 장치 간의 연결을 담당한다.


HILS 장치는 검증 타겟 로봇 SW가 탑재되어 검증을 위한 임무 시나리오 환경에서 해당 SW의 검증이 수행된다. 로봇 SW는 operating system (OS)가 없는 환경에서 구동될 수 있어 테스트베드 설계 시 non-OS 장치도 고려하였다.

2.2 생체모방로봇 모델 설계

본 논문에서 고려하고 있는 생체모방로봇은 Fig. 2와 같이 바퀴벌레 로봇, 장수풍뎅이 로봇, 및 벌새 로봇 총 3종이다. 바퀴벌레 로봇은 바퀴벌레의 생체적 특징을 바탕으로 장애물을 넘어가며 수색이 어려운 장소에서 활약할 수 있다. 장수풍뎅이 로봇은 빠른 이동을 통해 특정 공간을 빠르게 감시, 정찰 임무를 수행할 수 있다. 벌새 로봇은 비행을 통해 정찰 감시 반경을 극대화할 수 있고 회전의 또는 고정의 대비 소음이 적어 은밀한 탐지가 가능하다. 본 논문에서 검증 대상으로 고려하는 소프트웨어는 다양한 종류 및 기능을 가진 생체모방로봇에 공통적으로 적용할 수 있는 모델 기반 소프트웨어이다. HW 플랫폼에 독립적인 소프트웨어 기능만을 테스트하기 위해 로봇의 센서부와 제어부는 테스트베드를 통해 추상화된다. 테스트베드는 로봇 공통 기능들과(이동, 감시, 선회 등) 로봇 특화 기능들(점프, 날갯짓 등)의 테스트를 지원한다. Gazebo 시뮬레이터가 지원하는 기능을 바탕으로 구현한 상세한 로봇 모델 구성은 3.2절에서 살펴보겠다.

2.3 임무 검증 지원 테스트베드 GUI 설계

본 논문에서는 생체모방로봇의 검증을 위해 별도의 테스트



Robot	Roach	Beetle	Hummingbird
Drive Method	Control of 6 legs	Control of 6 legs	Control of both wings
Sensors	Camera IMU	GPS Camera IMU	GPS Camera IMU
Features	Overcoming rough terrain through jumping	reconnaissance/surveillance through rapid movement	stealth reconnaissance with less noise compared to rotary/fixed-wing

Fig. 2. Design of Biomimetic Robots

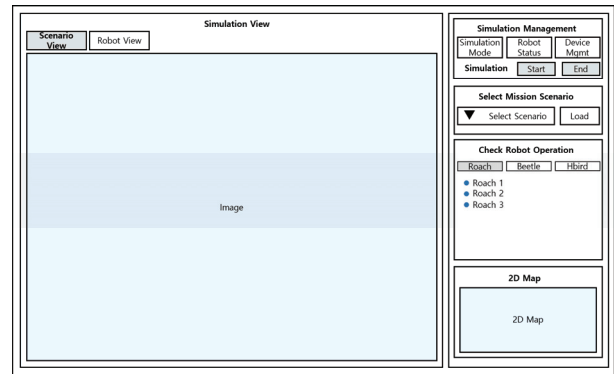


Fig. 3. Design of Testbed GUI

Table 1. Requirements of the GUI

Feature	Purpose
Mission Scenario Selection	Selection of a mission scenarios for robot software validation
Simulator Management	Management of the simulator, including its initiation and termination, along with monitoring the operations of HILS devices
Simulation View	Key view points of the mission scenarios
Robot Operation Validation	Validation of the current operational status for each type of robot
Robot Status Check	Assessment of the specific status of a robot using sensor data (camera, imu, etc.)
Simulation Map	Monitoring the movement of robots and the location of enemies/allies in the operation fields

베드 GUI를 설계하였다. 설계된 테스트베드 GUI는 Fig. 3과 같다. 설계에 있어 고려한 기능들은 Table 1과 같다.

2.4 HILS 기반 테스트베드 연동 요구사항

HILS 기반 연동 테스트베드에서는 시뮬레이션 상의 가상 로봇 모델을 제어하는 검증 대상 SW가 HILS 장치에서 실행된다. 이는 closed loop system인 로봇 내부의 센싱과 제어가 시뮬레이터와 HILS 장치 사이에서 이루어져야 함을 말한다. 즉, 가상 로봇 모델이 가상의 환경과 상호작용하며 생성하는 센서 데이터는 HILS 장치로 전송되고, 이 데이터를 기반으로 HILS 장치 내 검증 대상 SW가 제어한 데이터는 다시 시뮬레이터로 피드백되어야 한다. 생체모방로봇 또한 일반 로봇과 마찬가지로 안정적인 제어를 위해 요구되는 제어 주기가 있다. [11]에서 ‘roboBee’라는 로봇은 안정적인 비행 제어를 위해 날개를 날개 비트당 120 Hz로 날갯짓을 하며, 이는 8.3ms마다 제어되어야 함을 의미한다. 정리하면, HILS 기반 테스트베드에서는 시뮬레이터에서 생성된 센싱 데이터로부터 HILS 장치를 거쳐 다시 가상 로봇 모델로 제어 데이터가 피드백되기

까지의 총 시간이 8.3 ms (별세 로봇 기준) 이내여야 한다. 본 논문에서는 해당 값을 HILS 기반 테스트베드의 연동 요구사항으로 선정하였고, 이를 만족하기 위한 우선순위 기반 연동 미들웨어를 4절에서 자세히 살펴본다.

3. 테스트베드 구현

본 절에서는 2절에서 설계한 테스트베드 설계안을 바탕으로 구현한 테스트베드를 살펴본다.

3.1 테스트베드 구성 및 구현

구현된 테스트베드는 Fig. 4와 같다. 테스트베드는 앞서 언급한 바와 같이 테스트베드를 실행시키는 시뮬레이터 PC와 게이트웨이, 그리고 다수의 HILS 장치로 구성된다. 시뮬레이터와 게이트웨이는 robot operation system 2 (ROS2) [12]의 data distribution service (DDS) [13]를 통해 연동된다. HILS 장치에서는 검증을 요구하는 생체모방로봇 SW가 실행된다. 시뮬레이터에서 생성된 가상의 센서 데이터는 HILS 장치에 전송되어 각 장치의 관리를 통해 로봇 SW로 전송된다. 로봇 SW는 해당 센서 데이터를 기반으로 제어 명령 데이터를 생성, 해당 데이터가 시뮬레이터로 전송됨으로서 가상 로봇 모델을 제어한다. 게이트웨이는 HILS 장치와 이더넷으로 연결되며 로봇을 위한 센서 및 제어 신호를 이더넷 프레임에 encapsulation 하여 전송한다. 본 논문에서 구현된 게이트웨이 및 HILS 장치(보드)는 Fig. 5에서 살펴볼 수 있다.

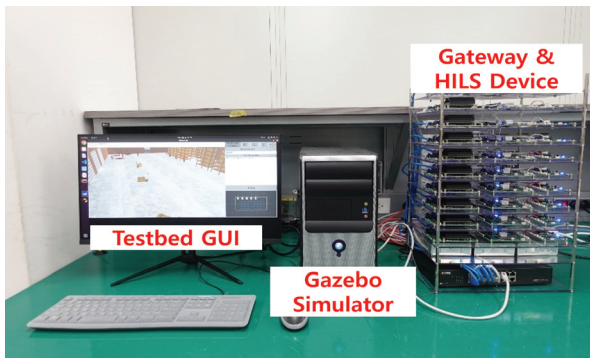


Fig. 4. Implementation of Testbed

3.2 생체모방로봇과 임무 검증 환경 모델 구현

Gazebo 시뮬레이터를 기반으로 개발한 로봇 모델의 구조는 Fig. 6A, Fig. 6B, 및 Fig. 6C에서 살펴볼 수 있다. 가상의 로봇 모델은 HILS 장치의 로봇 SW가 생성한 제어 데이터가 포함된 control topic을 통해 제어된다. 이때, Control topic 내 제어 데이터는 검증 타겟 로봇 SW와 HILS 장치 구성에 따라 하드웨어 수준의 하위 레벨 제어 신호(예. pulse width modulation)일 수도 있고, 상위 레벨 제어 신호(로봇의 선속도와 각속도)일 수 있다. Converter는 가상 로봇 모델의 제어 수준과 맞춰주는 역할을 한다. 본 논문에서 구현된 가상 로봇 모델은 선속도와 각속도(cmd_vel)를 바탕으로 제어되기에, 상위 레벨 제어 수준에 맞췄다. 가상의 환경 모델과 상호작용하며 제어 데이터에 따른 가상 로봇 모델의 구동은 구동 플러그인에서 계산된다.

로봇 SW로부터 가상 로봇 모델이 제어됨에 따라 가상 환경 모델과 상호작용하며 센서 데이터가 생성된다. 카메라 플러그인, IMU 플러그인, 배터리 플러그인 등은 Gazebo 시뮬레이터의 기능들을 활용하여 적합한 센서 데이터를 생성한다. 로봇의 위치와 이동 경로를 정확하게 계산할 수 있다는 시뮬레이션의 장점으로 odometry와 같은 센서 데이터에서 추정해야 하는 값들도 추출하여 활용할 수 있다. 이는 생체모방로봇 소프트웨어 개발자들이 개발 대상을 더 명확히 정의하고, 개발 시간 및 비용을 절약하는 데 큰 도움이 된다.

제시된 로봇 모델 외에도 바퀴벌레, 장수풍뎅이 등 다양한 생체모방로봇을 개발, 테스트베드 내에서 활용할 수 있다. 본

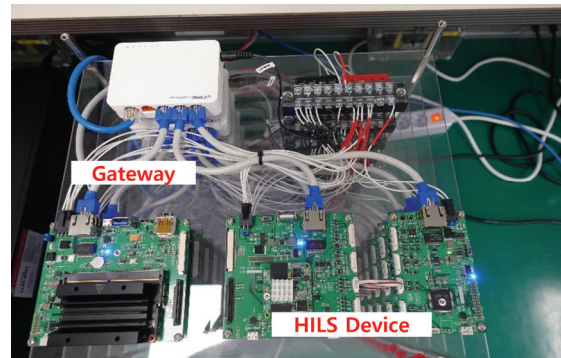


Fig. 5. Implementation of Gateway and HILS Device

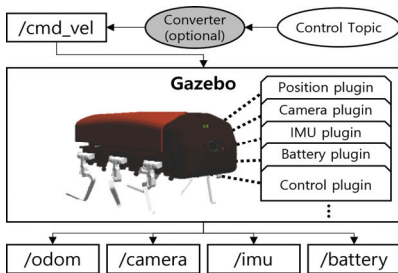


Fig. 6A. Roach Robot

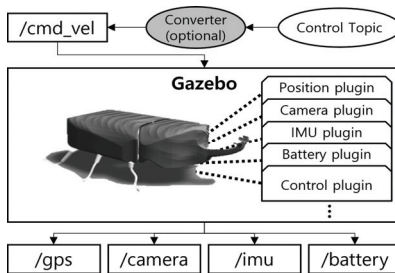


Fig. 6B. Beetle Robot

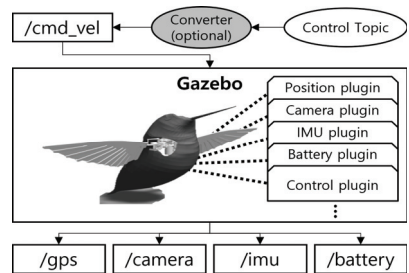


Fig. 6C. Bird Robot

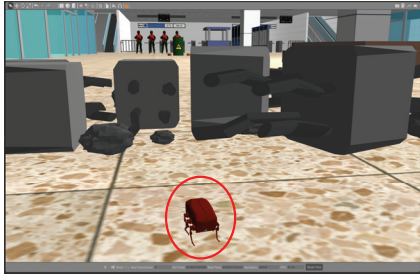


Fig. 7A. Roach Robot in Subway

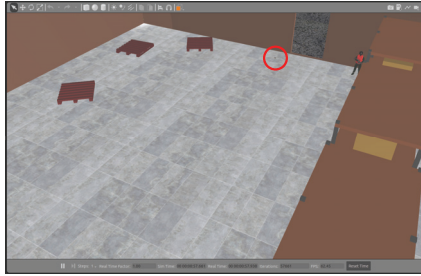


Fig. 7B. Beetle Robot in Warehouse

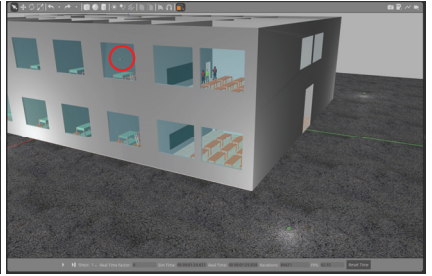


Fig. 7C. Bird Robot Around Enemy Building

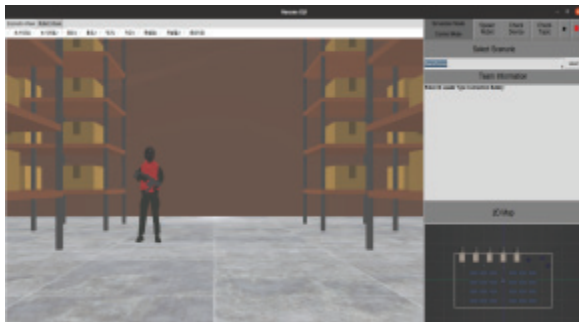


Fig. 8. Implementation of Testbed GUI

논문과 연관된 과제에서 해당 로봇 모델들의 설계와 구현을 진행한 바 있으나 보안상의 이유로 설명에서 제외하였다.

생체모방로봇의 개발과 더불어 생체모방로봇을 검증하기 위해 이들의 기능 테스트에 적합한 임무 환경 모델을 구현하였다. 각 로봇의 특성에 따라 바퀴벌레 로봇은 지하철의 수색 및 탐색, 장수풍뎠이 로봇은 적 점령 시설의 정찰, 벌새 로봇은 적 건물 외곽에서의 은밀 정찰 시나리오를 고려, 이에 따라 구현된 환경 모델은 Fig. 7A, Fig. 7B, 및 Fig. 7C에서 살펴볼 수 있다. 일련의 임무 환경 모델을 바탕으로 테스트베드 사용자는 생체모방로봇 모델의 생체를 모방한 기능들이 올바르게 동작하는지 검증할 수 있다.

3.3 임무 검증 지원 테스트베드 GUI 구현

구현한 임무 검증 지원 테스트베드 GUI는 Fig. 8과 같다. 앞서 설계 단계에서 고려한 GUI의 기능을 바탕으로 각 기능별 GUI layout을 구성하였다. 해당 GUI를 바탕으로 사용자는 임무 시나리오 수행 상황과 더불어 생체모방로봇의 기능 검증과 상태를 확인할 수 있다.

본 절에서는 우리가 제안한 다중 HILS 기반 생체모방로봇 운용 검증 지원 테스트베드를 살펴보았다. 시스템의 운용과 검증의 핵심 요소는 시뮬레이터와 다수의 HILS 장치 간 효과적인 연동이다. 다수의 생체모방 로봇을 검증할 시 검증 로봇 대수의 증가에 따라 송수신이 요구되는 데이터의 양이 증가하면서 연동 과정에서 지연이 발생할 수 있다. 이러한 지연을 방지하기 위해 우선순위 기반 연동 방법을 고려하였으며, 이와 관련된 제안과 실험 결과는 이어지는 4절에서 자세히 다룰 예정이다.

4. 우선순위 기반 테스트베드 장치 연동 및 실험

본 논문에서 설계 및 구현한 테스트베드는 2.4절에서 언급한 바와 같이 HILS 기반의 임무 검증을 수행하기 때문에 시뮬레이터와 HILS 장치 간의 데이터 연동이 무엇보다 중요하다. 데이터의 손실 또는 지연으로 전송이 제대로 이루어지지 않는다면 검증 대상 로봇 SW에서 도출한 결과값이 달라질 수 있다. 우리는 8.3 ms를 시뮬레이터와 HILS 장치 간 연동 요구사항으로 선정하였고, 시뮬레이션 수행에 따른 processing delay를 고려하여 단방향 통신 평균 지연 4 ms 이내를 목표로 하는 우선순위 기반 연동 미들웨어인 DDS-Priority (DDSP)를 구현, 기존 미들웨어 기술과의 비교 실험을 진행하였다. 비교 대상 미들웨어는 DDS와 TCPRODA.

다중 HILS 기반 테스트베드는 검증 대상 SW의 증가로 HILS 장치의 수가 많아지거나, 송/수신 데이터의 양이 많아짐으로 인하여 데이터의 손실 또는 지연이 발생할 수 있다. 데이터 전송의 품질 관리를 위해 DDS의 QoS를 사용할 수 있으나, 이는 설정된 QoS 값을 만족하는지에 대한 여부만을 알려주기 때문에, QoS를 만족하지 못할 상황의 대응 방안은 별도로 구성해야 한다. 본 논문에서는 이에 대한 해결책으로 전송해야 할 데이터에 우선순위를 부여하여 미들웨어 단에서 QoS를 만족하지 못한 event를 받으면 우선순위가 낮은 데이터의 전송 순서를 조절하여 높은 우선순위를 갖는 데이터의 전송을 보장하는 DDSP를 제안한다. DDSP 데이터 처리 순서도는 Fig. 9와 같다.

테스트베드 상황에서 DDS를 기반으로 주기적인 pub-sub 통신을 수행하는 n개의 플로우가 존재한다고 하였을 때, 각 subscriber 노드는 publisher 노드로부터 수신한 주기적 데이터가 QoS(Deadline)를 만족하는지 지속적으로 확인한다. 만약 데이터 스트림이 m번 이상 Deadline을 만족하지 못할 경우(실험에서는 m을 5로 설정함), subscriber 노드는 publisher 노드로 이벤트를 발생시킨다.

$$\sum_{k=1}^m (L_k > Deadline_k) \geq m \quad (1)$$

이벤트를 수신한 각 publisher 노드는 데이터 스트림의 우선순위를 비교하여 높은 우선 순위를 갖는 데이터 스트림이 먼저 전송될 수 있도록 전송 순서를 조절한다.

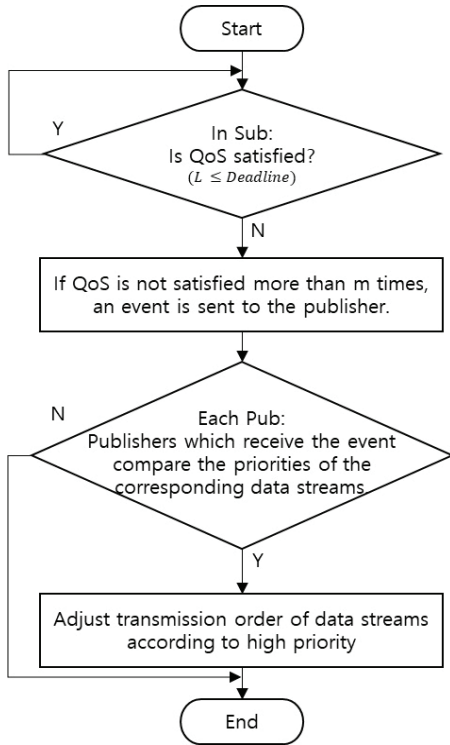


Fig. 9. Flow Chart of DDSP

if $P(S_i) > P(S_j)$ then S_i is transmitted before S_j (2)

그 후 모든 데이터 스트림에 대해 Equation (3)에 따라 우선순위를 기반으로 전송 순서를 결정한다.

$$O = \{S_k \mid P(S_k) \geq P(S_{k+1}), \forall k\} \quad (3)$$

데이터 전송의 우선순위를 고려한 테스트베드 연동 실험 환경은 Table 2와 같다. 본 실험에서는 시뮬레이터로부터 5대의 HILS 장치에 로봇 제어를 위한 자세 데이터(Topic Type A: 52 Byte)와 카메라 데이터(Topic Type B: 120K Byte)가 전송되어야 하는 상황을 가정한다. TCPROS는 Topic 마다 별도의 QoS를 설정하지 않았다.

DDSP, DDS 및 TCPROS의 평균 지연 결과는 Fig. 10과 같다. 평균 지연 결과 그래프에서 살펴볼 수 있듯이 DDSP는 우선 순위 기반의 전송 순서 처리를 수행하기 때문에 단순 event만 발생시키는 DDS와 달리 우선순위에 따른 데이터 전송 순서를 조절함으로써 높은 우선순위를 갖는 Topic Type A의 4 ms 이내 평균 지연 요구사항을 만족시킴을 보였다.

DDSP의 Type B의 평균 지연을 살펴보면 DDS의 Topic Type B 대비 상대적으로 지연이 더 길어진 것을 볼 수 있다. 이는 Topic Type A의 QoS 요구사항을 지원하기 위해 발생한 지연으로, 만약 연동이 요구되는 HILS 장치의 수가 더 늘어난다면 Topic Type B의 QoS를 만족할 수 없기에 우선순위를 기반한 네트워크 자원할당은 시뮬레이션 시나리오에 영향을 미치지 않는 수준으로 설정해야 한다. 이에 대한 방안 중 하나

Table 2. Experimental Setting

	QoS Setting	Data Priority
DDSP	Type A - Deadline, 4 ms Type B - Deadline, 20 ms	Type A - Priority high Type B - Priority low
DDS	Type A - Deadline, 4 ms Type B - Deadline, 20 ms	-
TCPROS	-	-

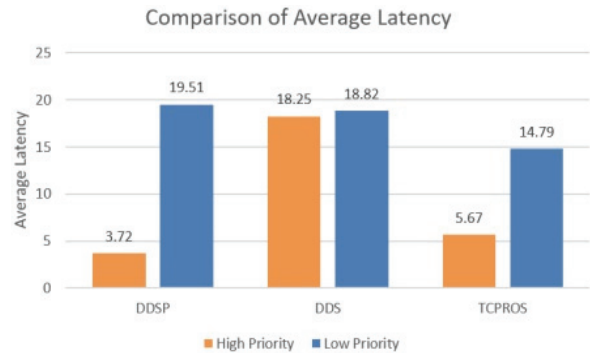


Fig. 10. Comparison of Average Latency

로 본 논문 저자들이 [14]에서 제안한 시뮬레이션 시나리오에서 시간대별 중요 우선순위를 기반으로 네트워크 자원할당을 조절하는 방법을 고려해볼 수 있다.

결과 그래프에서 특이한 점은, TCPROS는 Topic Type B 지연이 DDSP 및 DDS보다 작게 나오는 것을 볼 수 있다. 이는 TCPROS의 메시지 전송 프로세스가 DDS와 다른 점에 의한 차이이다. TCPROS는 일련의 메시지를 하나의 대형 버퍼에 채운 뒤, 해당 버퍼에서 TCP의 send를 call한다. 반면, DDS는 메시지를 여러 UDP 패킷으로 나누고, UDP 전송을 여러 번 호출하는 방식이기 때문에 TCPROS가 전송 지연이 짧을 수 있다. 하지만, Topic 마다의 QoS 요구사항을 고려하지 않고 일관된 메시지로서 버퍼에 채워 보내기 때문에, TCPROS의 Topic Type A처럼 지연 요구사항을 만족하지 못하는 상황이 발생할 수 있다. 게다가 검증 대상 HILS 장치의 node 수가 증가한다면, 신뢰성 보장을 위한 TCP의 연결형 서비스로 인해 통신 지연이 지속적으로 증가할 수 있다[15, 16]. 따라서 소프트웨어 검증을 위해 생성되는 센서 또는 제어 데이터들의 지연 요구사항이 다르고 검증 대상 HILS 장치 수가 다수인 본 테스트베드에서는 DDSP를 적용하는 것이 더 바람직하다.

5. 결 론

본 논문에서는 생체모방로봇 소프트웨어를 검증하기 위한 다중 HILS 기반 로봇 테스트베드를 설계, 구현하였다. 실제 로봇 확보의 어려움으로 인한 소프트웨어 기술 개발과 검증의 어려움을 해소하고자 M&S 기반의 테스트베드를 구축하였고, 생체모방로봇의 구동부를 포함한 소프트웨어 검증을 위해 HILS 기술을 적용하였다. 또한 개발된 로봇 소프트웨어를 검

증할 수 있는 테스트베드 GUI를 별도로 개발하였다. 우리는 테스트베드를 활용하여 가상의 로봇 모델(바퀴벌레 로봇, 장수풍뎅이 로봇, 벌새 로봇)과 각 로봇의 운용 시나리오에 적합한 환경 모델을 구현, 시뮬레이션을 수행하였다. 검증 대상 로봇 소프트웨어의 명령에 따라 가상 로봇 모델이 제대로 구동됨을 테스트베드 GUI를 통해 확인함으로써 생체모방로봇용 소프트웨어 검증 테스트베드로서 활용될 수 있음을 확인하였다. 테스트베드 내 장치 간 연동을 고려한 별도의 실험에서는 우선순위 기반의 DDS와 일반 DDS, TCPROS와 비교하였으며, 우선순위 기반 DDS가 높은 우선순위를 갖는 데이터의 전송 QoS를 보장할 수 있음을 확인하였다. 우리는 제한한 테스트베드가 생체모방로봇 소프트웨어의 검증을 지원함으로써 국내 생체모방로봇 기술 확보와 경쟁력 강화에 기여할 수 있는 것으로 기대한다.

References

[1] National Science and Technology Advisory Council Steering Committee, "Plan for Basic Source R&D Linkage for Future Defense Technology Acquisition," 2021.

[2] D. Kunertova, "Drones have boots: Learning from Russia's war in Ukraine," in *Contemporary Security Policy*, Vol.44, No.4, pp.576-591, 2023.

[3] E. Pan, X. Liang, and W. Xu, "Development of vision stabilizing system for a large-scale flapping-wing robotic bird," in *IEEE Sensors Journal*, Vol.20, No.14, pp.8017-8028, 2020.

[4] K. Jayaram and J. F. Robert, "Cockroaches traverse crevices, crawl rapidly in confined spaces, and inspire a soft, legged robot," in *Proceedings of the National Academy of Sciences*, Vol.113, No.8, pp.950-957, 2016.

[5] W. Jiankun, et al., "A survey of the development of biomimetic intelligence and robotics," in *Biomimetic Intelligence and Robotics*, Vol.1, 2021.

[6] N. Nedjah and S. J. Luneque, "Review of methodologies and tasks in swarm robotics towards standardization," in *Swarm and Evolutionary Computation*, Vol.50, 2019.

[7] H. S. Choi, et al., "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward," in *Proceedings of the National Academy of Sciences*, Vol.118, No.1, 2021.

[8] J. Collins, et al., "A review of physics simulators for robotic applications," in *IEEE Access*, Vol.9, pp.51416-51431, 2021.

[9] E. Moréac, E. M. Abdali, F. Berry, D. Heller, and J. -P. Diguët, "Hardware-in-the-loop simulation with dynamic partial FPGA reconfiguration applied to computer vision in ROS-based UAV," International Workshop on Rapid System Prototyping (RSP), Hamburg, Germany, 2020.

[10] N. Koenig and H. Andrew, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol.3, 2004.

[11] T. S. Clawson, et al., "A blade element approach to modeling aerodynamic flight of an insect-scale robot," *American Control Conference (ACC) IEEE*, 2017.

[12] Y. Maruyama, K. Shinpei, and A. Takuya, "Exploring the performance of ROS2," in *Proceedings of the 13th International Conference on Embedded Software*, 2016.

[13] G. Pardo-Castellote, "Omg data-distribution service: Architectural overview," in *23rd International Conference on Distributed Computing Systems Workshops*, 2003.

[14] G. Kim, et al., "Digital twin-based communication optimization method for mission validation of swarm robot," in *KIPS Transactions on Computer and Communication Systems*, Vol.12, No.1, pp.9-16, 2023.

[15] D. Madhuri and P. C. Reddy, "Performance comparison of TCP, UDP and SCTP in a wired network," *International Conference on Communication and Electronics Systems (ICES)*, Coimbatore, India, pp.1-6, 2016.

[16] K. Gatimu, A. Dhamodaran, T. Johnson, and B. Lee, "Experimental study of low-latency HD VoD streaming using flexible dual TCP-UDP streaming protocol," *15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp.1-6, 2018.



김 한 진

<https://orcid.org/0000-0002-2436-3784>

e-mail : gks359@koreatech.ac.kr

2017년 한국기술교육대학교

메카트로닉스공학부(학사)

2019년 한국기술교육대학교 컴퓨터공학과
(석사)

2019년 ~ 현재 한국기술교육대학교 컴퓨터공학과 박사과정
관심분야 : Cyber-physical Systems, Wireless Time-sensitive
Networking, Digital Twin



김 관 혁

<https://orcid.org/0000-0002-5046-9195>

e-mail : friend2281@koreatech.ac.kr

2021년 한국기술교육대학교 컴퓨터공학부

(학사)

2024년 한국기술교육대학교 컴퓨터공학과
(석사)

관심분야 : Digital Twin, Data Distribution Service, Time-sensitive
Networking



하 범 수

<https://orcid.org/0000-0003-0711-7108>
e-mail : habemsu@koreatech.ac.kr
2021년 한국기술교육대학교 전자공학부
(학사)
2024년 한국기술교육대학교 컴퓨터공학과
(석사)

관심분야: Digital Twin, Simulation, 로봇 제어



구 지 훈

<https://orcid.org/0000-0003-0472-0681>
e-mail : jeehoon.koo@lignex1.com
2014년 광운대학교 전자공학과(학사)
2014년~현 재 LIG넥스원
무인체계연구소 선임연구원
관심분야: 소형 로봇, Digital Twin, RF
Communication



김 주 영

<https://orcid.org/0009-0001-1811-2903>
e-mail : jooyoung.kim@lignex1.com
2007년 이화여자대학교 컴퓨터공학(학사)
2009년 이화여자대학교
컴퓨터정보통신공학(석사)
2009년~현 재 LIG넥스원무인체계연구소
수석연구원

관심분야: 군집 임무계획/할당, 이기종 운용 통제, 기계학습



김 원 태

<https://orcid.org/0000-0003-3426-3792>
e-mail : wtkim@koreatech.ac.kr
1994년 한양대학교 전자공학과(학사)
1996년 한양대학교 전자공학과(석사)
2000년 한양대학교 전자공학과(박사)
2001년~2005년 (주)로스텍테크놀로지
기술이사

2005년~2010년 한국전자통신연구원 임베디드SW연구부
선임연구원

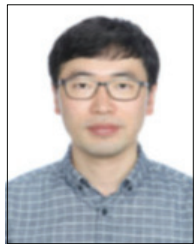
2010년~2015년 한국전자통신연구원 CPS연구실 실장/책임연구원

2015년~2019년 한국기술교육대학교 컴퓨터공학부 조교수

2019년~2023년 한국기술교육대학교 컴퓨터공학부 부교수

2023년~현 재 한국기술교육대학교 컴퓨터공학부 교수

관심분야: Cyber-physical Systems, Digital Twin, Internet
of Things, Dependable Computing,
Neuromorphic Computing, System of Systems
Engineering



심 성 준

<https://orcid.org/0009-0002-8473-1654>
e-mail : sungjun.shim@lignex1.com
2002년 경희대학교 기계공학과(학사)
2004년 경희대학교 기계공학과(석사)
2013년~현 재 LIG넥스원
무인체계연구소 수석연구원

관심분야: Swarm robot, 유무인복합체계, 초소형 로봇