# 로봇 제어 플랫폼의 통신 시스템 설계 및 구현

남승우, ,박재원*, 유경민*, 백의준*, 최정우*, 김명섭*
고려대학교

{nam131119, 2018270614, *rudals2710, *pb1069, *choigoya97, *tmskim}@korea.ac.kr

# Design of Communication System for Robot Control Platform

Nam Seung Woo, Park Jae Won, Yu Gyeong Min*, Baek Ui Jun*,
Choi Jeong Woo*, Kim Myung Sup*
Korea Univ.

## 요 약

현대 사회에서 기술 혁신의 중심에 있는 자율주행 기술의 발전은 자율주행 로봇 분야에도 중요한 영향을 미치고 있다. 이러한 로봇들은 다양한 환경에서 물류 운송부터 위험한 임무 수행까지 다양한 역할을 수행하며, 일상생활에서도 점점 중요한 역할을 차지하고 있다. 로봇 사용량의 증가에 따라, 효과적인 로봇 관제를 위한 플랫폼의 필요성이 대두되었다. 로봇 관제 플랫폼은 확장성과 유연성을 통해 새로운 로봇의 기능을 쉽게 추가할 수 있어야 하며, 원격 제어를 통해 로봇을 원격으로 연결하고 제어할 수 있어야 한다. 또한 실시간 모니터링 기능을 통해 주변 환경과 로봇의 상태를 실시간으로 확인할 수 있어야 한다. 본 논문에서는 이러한 요구 사항을 충족하고 각 요구 사항의 통신 체계를 구축하여 효율적인 로봇 제어 플랫폼을 제안한다.

## Ⅰ. 서 론

자율 주행 기술의 발전은 현대 사회에서 기술 혁신의 핵심 요소로 자리 잡고 있으며, 이러한 기술의 발전은 자율 주행 로봇 분야에도 영향을 미치고 있다. 자율 주행 로봇은 물류 운송, 위험한 임무 수행에서부터 일상생활의 보조에 이르기까지, 여러 환경에서 중요한 기능을 하고 있다. 자율 주행 로봇은 지상뿐만이 아니라 상공에서도 다양한 연구가 진행되고 있다. 드론 배달 시스템, 군사 목적을 가진 드론 등 다양한 분야에서 사용되고 있다. 이런 중요한 기능들을 맡으며 로봇의 사용량 증가를 가져왔으며, 결과적으로 효율적인 관리와 제어를 위한 로봇 관제 플랫폼[1]의 필요성이 증가했다. 로봇 관제 플랫폼을 설계하여 로봇은 더욱 복잡한 작업을 수행하고, 다양한 환경에서 안정적으로 작동할 수 있게 된다. 이러한 효율적인 활용과 관리할 수 있는 로봇 관제 플랫폼이 필요하며 필요성은 다음과 같다.

- 확장성 및 유연성: 새로운 로봇 기능이 추가되면 쉽게 추가할 수 있게 한다.
- 실시간 모니터링: 실시간으로 주변 환경 모니터링과 로봇의 상태 또한 확인할 수 있어야 한다.
- 원격 제어: 로봇을 원격으로 연결해 제어할 수 있게 한다.

로봇 관제 플랫폼은 위와 같은 필요성을 가져야 하며 플랫폼의 유지 및 관리가 용이해야 한다. 본 논문은 자율 주행 로봇의 이러한 요구 사항을 충족시키며 안정적이고 효과적인 관리를 위하여 다양한 통신 방식 시스템 기반 로봇 제어 플랫폼을 설계한다.

본 논문은 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력 기반 지역혁신 사업의 결과이다. (2021RIS-004)

## Ⅱ. ROS(Robot Operation System)

본 논문에서의 중앙 관리 관제 플랫폼 구축의 기반이 될 로봇 체계는 ROS2[2] 기반으로 구축하였다. ROS(Robot Operating System)는 로봇 소프트웨어 개발을 위한 유연하고 강력한 프레임워크이며 유연성과 강력한 기능들을 제공한다. ROS2는 기존 ROS1의 후속 버전으로 ROS1은 ROS Master를 사용하여 마스터 노드가 전체 시스템을 관리했지만, ROS2는 분산형 컴퓨팅 아키텍처를 사용하며, 실시간 및 분산 시스템을 위한 데이터 통신 서비스인 DDS(Data Distribution Service)를 사용하여 마스터 노드를 사용하지 않고 분산 시스템과의 호환성을 강화했다. ROS 환경에서는 기본적으로 토픽(Topic), 서비스(Service), 액션(Action) 등의 통신 방법을 사용해 로봇 간의 효율적인 데이터 교환과 상호작용을 지원한다.

토픽은 퍼블리셔(Publisher)와 서브스크라이버(Subscriber) 사이의 통신 스트림으로, 퍼블리셔가 브로드캐스트로 전송한 토픽을 서브스크라이버가 수신하는 단방향 통신 방식이다. 서비스와 액션은 양방향 통신 방식으로, 서비스는 클라이언트(Client)가 서버(Server)에 특정 요청을 보내고, 서버로부터 응답받는 방식이며 액션은 클라이언트가 장기 실행 명령을 서버에게 보내면 서버로부터 명령 진행 상황의 피드백과 결과를 받는 방식이다. ROS는 이러한 방식을 사용하여 다양한 데이터들을 송수신하고 처리한다.

ROS 환경 안에서 로봇은 SLAM(Simultaneous Localization and Mapping)과 Navigation을 사용하여 구역을 주행한다. SLAM은 동시적 위치추정 및 지도작성 기술로 로봇이 주어진 구역을 주행하기 위해 맵을 만들고 자신의 위치를 추정하는 기술이다. Navigation은 그려진 맵을 바탕으로 목적지를 설정하여 로봇이 자율 주행을 할 수 있도록 하는 기술이다.

## Ⅲ. 로봇 제어 플랫폼 구축

로봇 제어 플랫폼을 구축하기 위하여 프론트엔드, 백엔드, 데이터베이스를 구축하였다. 각 서버는 도커(Docker)의 컨테이너를 활용하여 각 서버를 구축하였다. 도커는 컨테이너 기반의 가상 환경 구축 프로그램으로, 서버를 컨테이너라는 격리된 환경에서 구축 및 실행할 수 있게 해주는 프로그램이다. 도커를 사용하여 하나의 서버에서 프론트엔드, 백엔드, 데이터베이스를 각각 독립적으로 구축하면서 각 컨테이너는 한 부분에서 문제가 발생하면 전체에 영향을 끼치지 않도록 개발했다.

각 서버를 구축한 후, 로봇과의 통신을 위하여 HTTP[3] 방식의 GET, POST 방식과 Socket[4] 방식의 통신 체계를 구축하였다. GET, POST 통신은 HTTP(Hypertext Transfer Protocol) 방식의 통신 방식이다. 클라이언트와 서버가 통신하는 방식이며 각각마다 주요 목적이 다르다. GET 통신은 일반적으로 데이터를 검색하기 위한 통신 방식으로 서버에 URL의 포트로 데이터를 전송하면 서버는 사전에 구축한 API의 URL에 따라 처리를 진행한다. POST 통신은 일반적으로 서버의 상태나 데이터를 변경하기 위해 사용하는 방식으로 HTTP 메시지의 본문(body)에 데이터를 담아 전송하면 서버가 사전에 구축된 API에 따라 이를 처리한다. Socket 방식은 서버가 클라이언트와 연결한 다음에 통신하는 방식으로 지속적인 데이터 통신을 진행할 때 유용한 통신 방식이다.

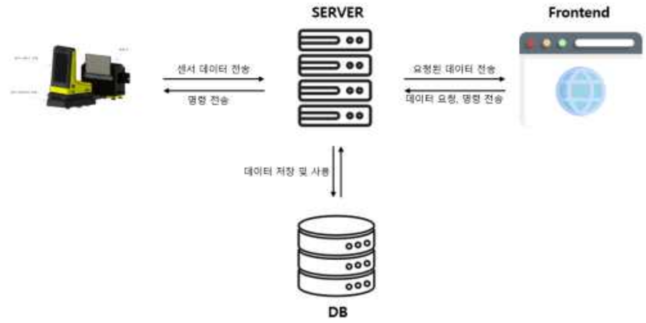


그림 1. 로봇 제어 플랫폼 구조

위에서 언급한 통신 방식을 사용하여 로봇 제어 플랫폼을 구축하였고 플랫폼 전체 구조도는 그림 1과 같다. 로봇은 SLAM에서 발행하는 로봇의 위치 정보와 로봇의 각종 센서 데이터들을 구독하고 이를 서버에 POST 통신을 사용하여 주기적으로 전송한다. 서버는 로봇의 센서 정보들을 데이터베이스에 저장하고 프론트엔드는 저장된 로봇의 센서 정보들을 GET 요청을 통하여 가져온다. 그림 2는 구축한 프론트엔드 화면이다.

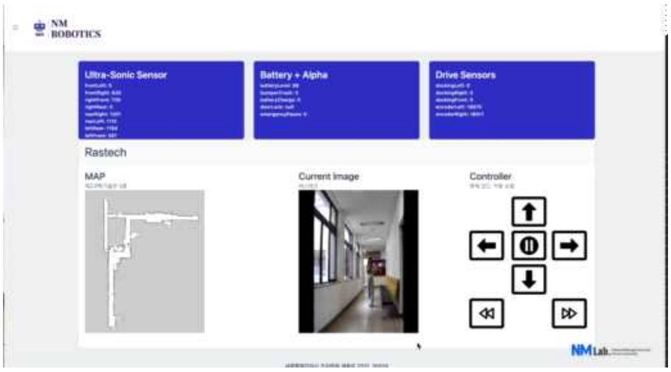


그림 2. 플랫폼 프론트엔드 화면

프론트엔드에서는 로봇의 초음파 센서, 배터리 잔량, 주행 관련 센서, 로봇의 위치 정보를 표시한다. 하단 하얀 배경에서 지도 같은 경우, 로봇의 ROS2의 절대 좌표 정보를 가지고 있는 토픽을 서브스크라이브하여 해당 정보만 추출 후, 백엔드로 post 해준다. 백엔드는 수신한 정보를 db에 저장하며 프론트엔드가 해당 정보를 get 요청하게 되면 db에서 찾아 응답해준다. 프론트엔드는 지도 이미지 기준으로 로봇의 절대 좌표를 이미지 기준 상대 좌표로 변환해 지도 사진에 로봇의 위치를 시각적으로 보일 수 있게 하였다.

중앙의 Current Image는 관리자의 지속적인 실시간 확인을 위해 로봇에 장착된 카메라를 통해 이미지를 저장하고 이를 Socket 방식을 사용하여 백엔드와 연결해 이미지를 전송한다. 백엔드는 실시간으로 프론트엔드에 출력할 수 있도록 post 방법을 활용해 전송한다.

오른쪽의 Controller는 지속적인 원격 제어를 위해 로봇과 서버 간의 소켓 방식의 통신을 통해 백엔드와 연결한다. 프론트엔드에서 컨트롤러 버튼이나 키보드의 화살표를 누를 시, 백엔드에 해당 제어 명령을 보내게 되며 백엔드는 로봇과 연결된 소켓을 통해 해당 제어를 전달한다. 로봇의 소켓은 제어값을 받을 시, 로봇을 직접 제어할 수 있는 '/cmd_vel' 토픽을 발행해 로봇을 움직이도록 하였다.

## Ⅳ. 결론

자율 주행 로봇을 안정적이고 효과적으로 관리하기 위해 로봇 제어 플랫폼의 통신 시스템을 성공적으로 설계 및 구현하였다. 확장성과 유연성을 고려한 통신 프로토콜, 실시간 모니터링, 원격 제어 기능을 통합하여 로봇의 안정적인 운영과 효율적인 제어가 가능해졌다. 그에 따라 로봇을 효율적으로 관리할 수 있게 되었으며 ROS2 기반의 관리 시스템을 통해 데이터 통신의 신뢰성과 효율성을 향상시키는 데에 기여하였다. 그에 반면, 인터넷 신호가 취약한 구역에서는 정상적인 통신이 이루어지지 않았으며 장시간의 자율 주행 시 오차 누적에 따른 로봇이 자신의 위치를 제대로 판단하지 못하는 kidnap 현상에 대처하지 못했다.

추후 연구에서는 본 논문의 플랫폼의 세부적인 기능과 위에서 언급한 문제점들을 개선하고 다양한 기능을 추가하여 더욱 효율성 있는 플랫폼을 구축할 예정이다.

## 참 고 문 헌

[1] Bong Keun Kim, M. Miyazaki, K. Ohba, S. Hirai and K. Tanie, "Web Services Based Robot Control Platform for Ubiquitous Functions," Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 691-696, doi: 10.1109/ROBOT.2005.1570198.

[2] MARTÍN, Francisco, et al. Plansys2: A planning system framework for ros2. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021. p. 9742-9749.

[3] Q. Liu and X. Sun, "Research of Web Real-Time Communication Based on Web Socket," International Journal of Communications, Network and System Sciences, Vol. 5 No. 12, 2012, pp. 797-801. doi: 10.4236/ijcns.2012.512083.

[4] Q. Liu and X. Sun, "Research of Web Real-Time Communication Based on Web Socket," International Journal of Communications, Network and System Sciences, Vol. 5 No. 12, 2012, pp. 797-801. doi: 10.4236/ijcns.2012.512083.