

산업용 로봇 원격제어를 위한 조이스틱을 이용한 ROS기반 임베디드 시스템

권용성*, 김은수**, 신재욱***, 박범용****

ROS based Embedded System using a Joystick for Industrial Robots Remote Control

Yong-Sung Kwon*, Eun-Su Kim**, Jae-Wook Shin***, and Bum-Yong Park****

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성 (Grand ICT연구센터) 사업의 연구 결과로 수행되었음. (IITP-2021-2020-0-01612)

요 약

4차 산업혁명이 전개됨에 따라 스마트 팩토리의 필요성이 대두되고 있다. 스마트 팩토리에 필수적인 산업용 로봇의 수요가 늘 것으로 예상된다. 본 논문에서는 임베디드 환경에서 조이스틱을 활용한 산업용 로봇 제어방식을 제안한다. 조이스틱을 이용해 로봇의 말단 좌표를 설정하면, 알고리즘을 이용해 설정한 좌표가 동작 가능한 좌표인지 판별한다. 동작 불가능한 좌표라면 대체 좌표로 변환 후 역기구학 해를 이용해 해당 좌표에 대응하는 로봇의 관절 각도를 구한다. 계산된 관절 각도를 ROS(Robot Operating System)통신을 통해 산업용 로봇의 제어 박스에 전달해 로봇의 관절을 제어한다. 제안하는 시스템의 시뮬레이션 및 동작 결과 조이스틱을 이용한 산업용 로봇 실시간 원격제어가 가능한 것을 확인하였다.

Abstract

With the development of the 4th industrial revolution, the need for a smart factory is emerging. The demand for industrial robots essential for smart factory is expected to increase. In this paper, we propose an industrial robot control method using a joystick in an embedded environment. When the robot's terminal coordinates are set using a joystick, it is determined whether the coordinates are operable using an algorithm. If the coordinates are impossible, convert them to alternative coordinates and use the inverse kinematics solution to find the joint angle of the robot corresponding to the coordinates. The calculated joint angle is transmitted to the control box of the industrial robot through Robot Operating System (ROS) communication to control the robot's joints. As a result of simulation and operation of the proposed system, it was confirmed that real-time remote control of industrial robot using joystick was possible.

Keywords

embedded system, ROS, joystick, industrial robots, inverse kinematics

* 금오공과대학교 IT융복합공학과 석사과정
- ORCID: <https://orcid.org/0000-0002-4593-9260>
** 금오공과대학교 전자공학과 석사과정
- ORCID: <https://orcid.org/0000-0002-4043-0713>
*** 금오공과대학교 전자공학부 조교수
- ORCID: <https://orcid.org/0000-0002-3765-6009>
**** 금오공과대학교 IT융복합공학과 조교수(교신저자)
- ORCID: <https://orcid.org/0000-0001-9490-0365>

• Received: Jul. 30, 2021, Revised: Sep. 17, 2021, Accepted: Sep. 20, 2021
• Corresponding Author: Bum-Yong Park
Dept. of Electronic Engineering, IT Convergence Engineering, Kumoh National Institute of Technology, 61, Daehak-ro, Gumi-si, Gyeongsangbuk-do, 39177, Korea
Tel.: +82-54-478-7438, Email: bumyong.park@kumoh.ac.kr

I. 서 론

최근 정보통신 기술의 발달을 발판삼아 4차 산업혁명이 이뤄지고 있다. 4차 산업혁명의 핵심 기술 중 하나인 스마트 팩토리는 공장 내 설비와 기계에 설치된 센서가 데이터를 실시간으로 수집하여 공장 내 모든 상황이 일목요연하게 보여지고, 이를 분석해 원하는 목적에 따라 스스로 제어되는 공장을 나타낸다. 스마트 팩토리는 생산 현장에서 발생하는 문제들의 상관관계를 언어낼 수 있고, 숙련자들의 노하우를 데이터화 함으로써 누구나 쉽게 활용 가능하다는 장점이 있다[1]. 이러한 스마트 팩토리 기술을 적용하기 위해서는 산업용 로봇 사용이 필수적이다. 따라서 앞으로 제조업 분야에 산업용 로봇 사용 확대가 예상된다.

산업용 로봇은 일반적으로 티칭 펜던트라는 별도의 기기를 통해 프로그램 트리를 작성하고 로봇의 동작을 제어한다. 티칭 펜던트, 프로그램 트리에 익숙하지 않은 사용자는 로봇의 동작 제어과정에 어려움이 있다. 또한, 기존의 티칭 펜던트는 컨트롤 박스에 유선 연결되어 있어 사용 거리와 휴대성에 제약이 있다. 이러한 한계를 해결하기 위해 카메라 이미지를 통한 CNN기반 산업용 로봇 제어 연구[2], EMG신호를 이용한 손 제스처 인식기반 산업용 로봇 제어 연구[3], ECG신호를 이용한 협동 로봇 제어 연구[4] 등 산업용 로봇을 제어할 다양한 방법들이 연구되고 있으며 임베디드 보드 환경에서 ROS (Robot Operating System)를 통한 원격제어[5]-[7] 또한 연구되고 있다. 하지만 위 연구들은 아직 산업현장에서 사용하기에는 정확성이 떨어지고 여전히 작업자들이 사용하기에 직관적이지 못하다.

본 논문에서는 ROS를 기반으로 임베디드 보드 환경에서 조이스틱을 이용한 산업용 로봇 제어 방식을 제안한다. 조이스틱을 이용한 직관적인 제어 방식을 통해 작업자들의 빠른 적응이 가능하며 엡지 컴퓨팅을 통해 작업자의 공간적 제약 문제 또한 해결될 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 시스템의 구성과 개발환경, ROS, 역기구학에 대해 기술한다. 3장에서는 앞서 기술된 내용을 토대로 시뮬레이션과 실제 UR3 로봇 제어를 진행한다.

Matplotlib과 GAZEBO를 이용해 시뮬레이션을 구현하였고, PC 환경과 임베디드 보드 환경에서의 UR3 로봇 응답 지연시간을 확인하고, 마지막으로 4장에서는 본 논문의 결과에 관해 기술한다.

II. 제안하는 제어 시스템

2.1 제안하는 제어 시스템 구성도

그림 1은 제안하는 산업용 로봇 제어 시스템의 구성도이다. 조이스틱의 X, Y, Z 3축에 대한 디지털 값을 아두이노 시리얼 통신을 통해 Jetson Nano에 전달한다. Jetson Nano의 처리부에서 전달받은 디지털 값을 정규화시켜 로봇의 말단 좌표 제어에 사용할 수 있게 한다. 조이스틱을 이용해 산업용 로봇의 말단 좌표가 이동하면 이에 대응하는 각 관절들의 각도를 역기구학 해를 이용해 구한다. 관절 각도는 ROS의 action 통신을 통해 원격으로 산업용 로봇의 컨트롤 박스에 전달되어 로봇을 제어한다.

2.2 개발환경

제안하는 시스템은 3차원 공간상의 산업용 로봇 제어를 목표로 한다. 따라서 3차원 좌표 설정을 위해 3축 제어가 가능한 APEM사의 HF33R10을 사용한다. HF33R10은 상하좌우 움직임에 스틱 손잡이 회전을 추가해 그림 2와 같은 방향으로 X, Y, Z 3축 제어가 가능하다.

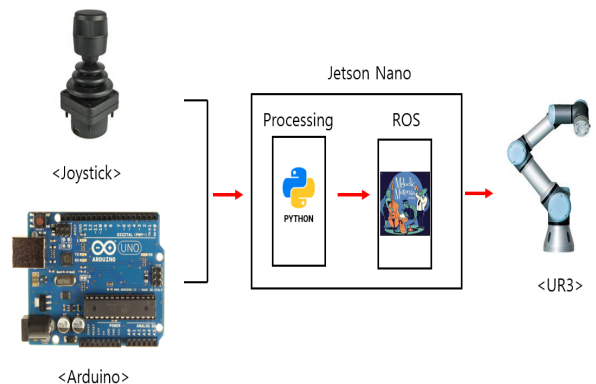


그림 1. 제안하는 산업용 로봇 제어 시스템 구성도
Fig. 1. Structure of the proposed industrial robot control system

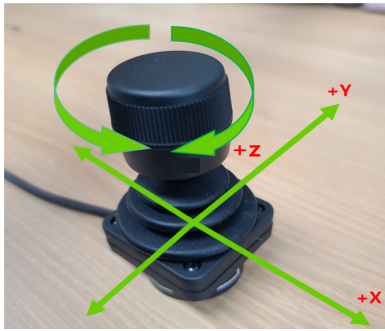


그림 2. 제안하는 시스템에서 사용된 조이스틱의 축 구성
Fig. 2. Configuring the axis of the joystick used in the proposed system

X, Y축은 중심을 기준으로 최대 18° 까지 기울어지며 원형의 동작 범위를 가진다. Z축은 중심을 기준으로 ±30° 까지 회전 가능하다. HF33R10은 각 축마다 0V ~ 5V의 값을 출력하며 중립상태일 때 2.5V를 출력한다. 이번 연구에서는 이 값을 디지털 값으로 변환 후 정규화를 진행하여 -0.64 ~ 0.64의 값으로 변환하며, 이 값은 로봇의 말단 좌표 설정에 사용된다.

실제 동작에는 Universal Robots 사의 UR3 로봇을 사용한다. 그림 3은 UR3 로봇의 관절 구성을 나타낸 그림이다. UR3 로봇은 Base, Shoulder, Elbow, Wrist1, Wrist2, Wrist3 총 6개의 관절로 이루어져 있으며 Wrist3는 무한회전, 나머지 관절은 ±360° 회전이 가능하다. 그림 4는 UR3 로봇의 축, 각도, 링크를 설명한 그림이다. 제안하는 시스템은 Base, Shoulder에 연결된 2개의 링크와 Elbow, Wrist1, Wrist2, Wrist3를 일직선으로 연결한 링크 총 3개의 링크를 이용한 3자유도 동작을 구현한다.

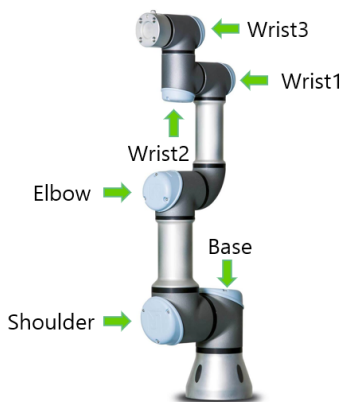


그림 3. UR3 로봇의 관절 구성
Fig. 3. Joint configuration of UR3

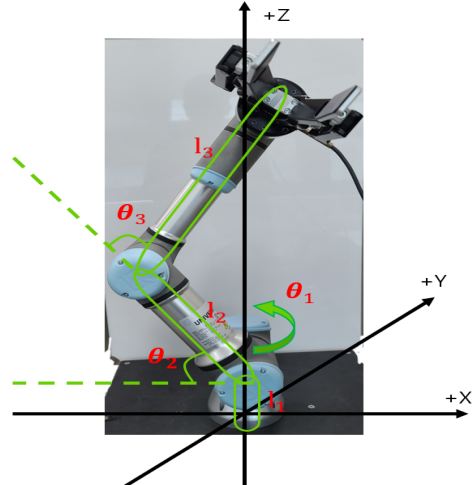


그림 4. 제안하는 시스템에서 사용된 산업용 로봇의 축, 각도, 링크 구성
Fig. 4. Configuring the axis, angles, and links of the industrial robot used in the proposed system

그림 4. Configuring the axis, angles, and links of the industrial robot used in the proposed system

제안하는 시스템에서는 Jetson Nano 임베디드 보드를 사용해 원격제어 시스템을 구현한다. Jetson Nano는 NVIDIA에서 제공하는 임베디드 보드로 128 코어 Maxwell GPU와 ARM A57 CPU를 장착한 단일보드 컴퓨터이다. 사용한 OS는 Ubuntu 18.04 LTS 이고, ROS는 Melodic Morenia 버전을 사용한다.

2.3 ROS 통신

ROS는 배포 가능한 모듈형 디자인 로봇 소프트웨어를 목표로 개발된 미들웨어로 기본적인 OS를 설치한 후 apt등의 패키징 툴을 이용하여 추가로 설치한다. ROS는 하드웨어의 가상화, TCP-IP 메시지 통신을 통한 분산 컴퓨팅 환경을 지원한다[8]. 또한 CoppeliaSim, GAZEBO, moveit 등 로봇 연구에 필요한 다양한 기능을 지원해 로봇 개발의 효율성을 향상시킨다[9].

ROS의 통신 방식에는 topic, service, action 3가지 방식이 있다. 3가지 방식 모두 노드라는 최소 단위 프로세스 간 topic을 통해 통신하는 공통점을 가진다. topic 통신은 단방향 통신, service는 양방향 동기 통신, action은 양방향 비동기 통신으로 제안하는 시스템은 action 통신 방식을 사용한다. 그림 5는 ROS action 통신을 설명하기 위한 예로 우선 server 노드에서 master에 topic을 발행한다.

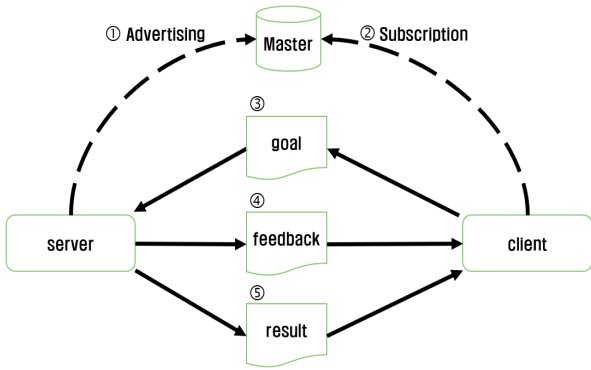


그림 5. ROS action 통신의 구조
Fig. 5. Structure of ROS action communication

Client노드에서는 master에 원하는 topic을 요청하면 master는 동일한 이름의 topic이 존재하는지 확인한 후 두 노드를 연결한다. 통신이 시작되면 client에서 goal을 요청하고 server에서는 goal에 맞는 작업을 수행한다. 작업 수행 중 client에 계속해서 feedback을 보내며, 작업이 완료되면 최종결과를 다시 client에 전송한다.

2.4 역기구학

역기구학이란 로봇의 말단 좌표와 자세가 주어지면 이에 대응하는 관절 각도를 결정하는 과정이다. 그림 6은 제안하는 시스템의 3자유도 구조체를 표현한 그림으로 직선은 링크, 원통은 관절을 나타내며 말단 좌표는 직교 좌표계 (x, y, z)로 표현한다.

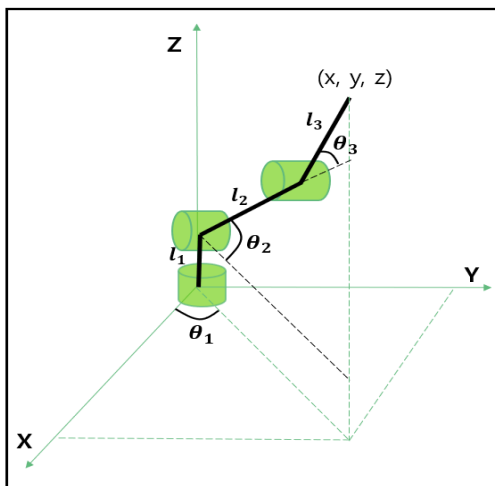


그림 6. 역기구학 해석을 위한 3자유도 구조체
Fig. 6. Three degrees of freedom structure for inverse kinematics analysis

역기구학 해를 구하기 위해서는 관절 간 상대 각도 (α), 링크의 길이(l, d), 회전 각도(θ) 4가지 요소로 구성된 DH(Denavit-Hartenberg) 매개 변수가 필요하다. DH 매개변수는 공통 법선이라는 개념을 이용하여 총 4개의 매개변수로 두 개의 링크간 관계를 표현하는 방법이다[10]. DH 매개 변수는 표 1을 통해 제공한다.

표 1. 제안 시스템의 DH 매개 변수
Table 1. DH parameters of the proposed system

Link	θ [rad]	l [cm]	d [cm]	α [rad]
1	θ_1	0	15.19	$\frac{\pi}{2}$
2	θ_2	24.365	0	0
3	θ_3	29.86	0	0

기구학은 DH 매개 변수와 관절 각도가 주어지면 로봇의 말단 좌표와 자세를 구하는 방법이다.

$$x = \{l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)\} \cos(\theta_1) \quad (1)$$

$$y = \{l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)\} \sin(\theta_1) \quad (2)$$

$$z = l_1 + l_2 \sin(\theta_2) + l_3 \sin(\theta_2 + \theta_3) \quad (3)$$

식 (1), (2), (3)은 3자유도에 해당하는 기구학 해로 Matplotlib을 활용한 시뮬레이션에서 위 식을 이용해 로봇의 동작을 계산한다.

$$\theta_3 = \tan^{-1} \left(\frac{\pm \sqrt{1 - D^2}}{D} \right) \quad (4)$$

$$\theta_2 = \tan^{-1} \left(\frac{z}{\sqrt{x^2 + y^2}} \right) - \tan^{-1} \left(\frac{l_3 \sin(\theta_3)}{l_2 + l_3 \cos(\theta_3)} \right) \quad (5)$$

$$\theta_1 = \tan^{-1} \left(\frac{y}{x} \right) \quad (6)$$

$$D = \frac{x^2 + y^2 + z^2 - l_2^2 - l_3^2}{2l_2l_3} \quad (7)$$

위 식은 3자유도 역기구학 해석을 위한 식이다[11]. 식 (4)를 통해 Elbow 관절 각도, 식 (5)를 통해 Shoulder 관절 각도, 식 (6)을 통해 Base 관절 각도를 구한다. 식 (7)의 D는 앞의 식들을 간편히 표현하기 위해 사용한 변수이다.

2.5 말단 좌표 설정 영역 제한

산업용 로봇은 각 관절의 길이에 따른 동작 범위가 제한된다. 따라서 조이스틱으로 설정한 로봇의 말단 좌표가 동작 가능 영역을 벗어나면 발생하는 문제를 막기 위한 알고리즘이 필요하다. 제안하는 시스템은 직관적인 제어를 목표로 하기에 3차원 직교 좌표계로 로봇을 제어한다. 하지만 산업용 로봇은 Base를 중심으로 구 형태의 동작 범위를 가져 구면좌표계 사용이 적합하다. 따라서 말단 좌표 영역 제한을 위해 직교 좌표계 (x, y, z)로 표현된 말단 좌표를 구 좌표계 (r, ϕ, θ)로 변환한다.

그림 7은 동작 제한 알고리즘을 설명하기 위한 예시 그림이다. 그림 7에서 보이듯 산업용 로봇은 r이 |l₂+l₃|보다 큰 경우, |l₂-l₃|보다 작은 경우 두 가지 동작 제한 범위를 갖는다. 그림 7의 A 좌표가 식 (8), B 좌표가 식 (9)와 같다고 가정한다.

$$A = (r_A, \Phi_A, \theta_A), \quad (r_A > |l_2 + l_3|) \quad (8)$$

$$B = (r_B, \Phi_B, \theta_B), \quad (r_B < |l_2 - l_3|) \quad (9)$$

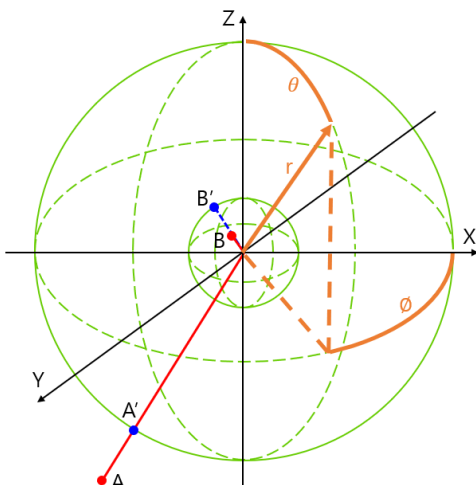


그림 7. 말단 좌표 영역 제한
Fig. 7. Limit the area of terminal coordinate

A, B 모두 동작 불가능한 좌표이다. 이 경우 좌표값의 ϕ, θ는 고정 후 r을 조정한다. 식 (10)의 A' 좌표는 r_A를 최대 동작 길이 |l₂+l₃|로, 식 (11)의 B' 좌표는 r_B를 최소 동작 길이 |l₂-l₃|로 조정된 좌표이다.

$$A' = (|l_2 + l_3|, \Phi_A, \theta_A) \quad (10)$$

$$B' = (|l_2 - l_3|, \Phi_A, \theta_A) \quad (11)$$

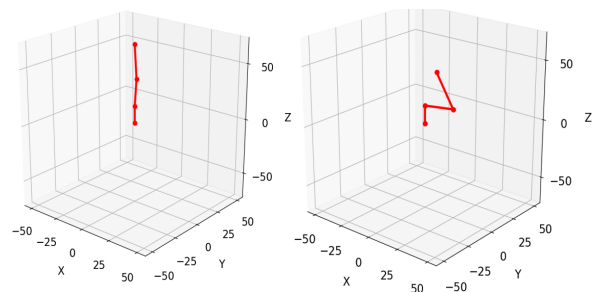
기존의 티칭 펜던트를 이용한 제어방식은 로봇이 제어 가능 범위 밖으로 벗어나게 되면 로봇의 동작을 강제로 정지시켰다. 제안하는 알고리즘을 사용함으로써 조이스틱을 이용한 산업용 로봇 제어과정에서 오류로 인한 멈춤 없이 연속적인 제어가 가능하다.

III. 시뮬레이션 및 구현

제안하는 시스템을 통해 실제 UR3 로봇을 제어하는 것에 앞서 구성한 알고리즘을 시뮬레이션해 본다. 3.1절에서는 Matplotlib을 이용한 시뮬레이션, 3.2절에서는 GAZEBO를 이용한 시뮬레이션을 진행한다. 두 차례의 시뮬레이션 진행 후 3.3절에서 PC와 Jetson Nano 환경에서 UR3 로봇을 제어하였다.

3.1 Matplotlib을 이용한 시뮬레이션

그림 8은 python의 Matplotlib 모듈을 사용해 3자유도 매니퓰레이터를 구현한 그림이다.



(a) 초기자세 (b) 동작 후 자세

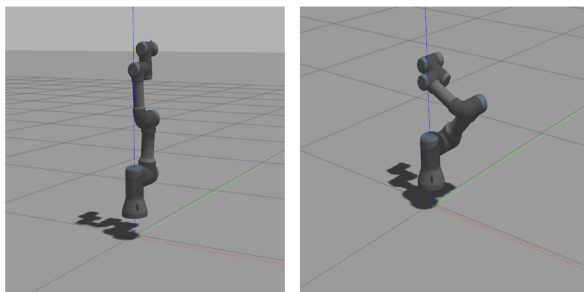
그림 8. Matplotlib을 이용한 3자유도 동작 시뮬레이션
Fig. 8. Three degrees of freedom motion simulation using Matplotlib, (a) Initial pose, (b) Post-movement pose

붉은 점은 로봇의 관절과 말단 좌표를 나타내며 붉은 선은 링크를 나타낸다. 그림 8(a)는 조이스틱을 이용해 로봇의 말단 좌표를 직교 좌표계 $(0, 0, l_1+l_2+l_3)$ 로 설정한 모습이다. 그림 8(b)는 말단 좌표를 $(10, 0, 30)$ 로 설정한 모습이다. Matplotlib 시뮬레이션을 통해 로봇의 관절이 어떻게 동작할지, 조이스틱을 이용해 제어하는 과정에서 센서값이 정확히 전달되고 있는지 확인하였다.

3.2 GAZEBO를 이용한 시뮬레이션

GAZEBO는 오픈소스 기반의 로봇 시뮬레이터로 ROS와 연동 시에는 실제 로봇에 대한 모니터링을 위해 사용 가능하며 다양한 물리엔진이 제공되어 실제 상황과 유사한 환경을 구성할 수 있다는 장점이 있다[12]. GAZEBO 시뮬레이션은 역기구학 해를 이용해 계산된 관절 각도를 ROS action 통신으로 MoveIt에 전달하여 GAZEBO의 UR3 로봇을 제어하도록 설계한다. MoveIt은 로봇 모션 플래닝에 사용되는 도구로 시뮬레이터인 GAZEBO의 동작 알고리즘을 설정할 때 사용한다.

그림 9(a)는 조이스틱을 이용해 로봇의 말단 좌표를 직교 좌표계 $(0, 0, l_1+l_2+l_3)$ 로 설정했을 때 시뮬레이션 결과이다. 그림 9(b)는 $(10, 0, 30)$ 로 좌표를 시뮬레이션한 결과이다. GAZEBO를 이용한 시뮬레이션을 통해 Matplotlib에서는 확인하기 힘들었던 Base 관절의 움직임, 동작 중 링크 간의 충돌을 확인하고 ROS action 통신의 연결을 확인하였다.



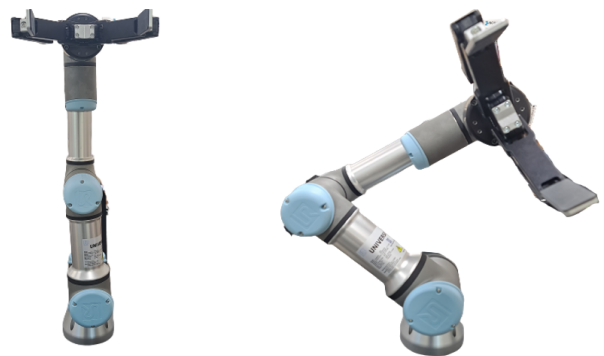
(a) 초기자세 (b) 동작 후 자세

그림 9. GAZEBO를 이용한 3자유도 동작 시뮬레이션
Fig. 9. Three degrees of freedom motion simulation using GAZEBO, (a) Initial pose, (b) Post-movement pose

3.3 PC 환경 및 임베디드 보드에서 UR3 동작

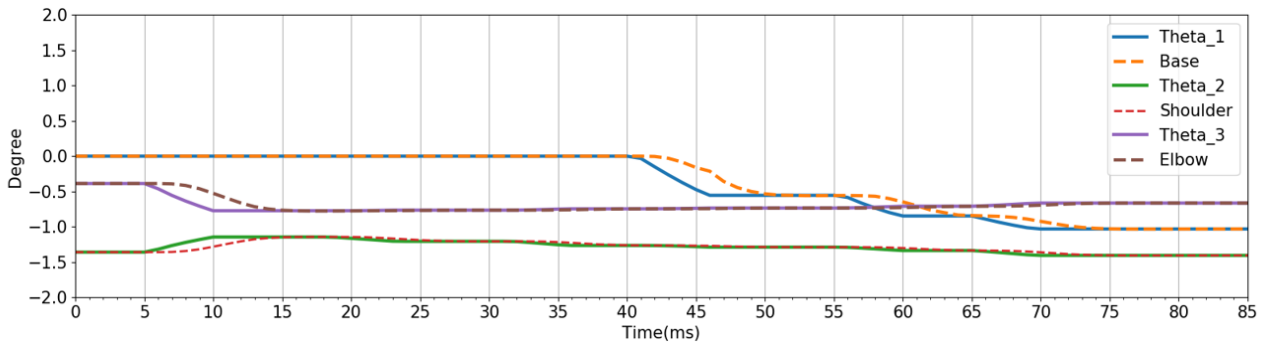
PC 환경과 임베디드 보드 환경에서의 성능 차이를 확인하기 위해 PC에서 구현한 시스템을 Jetson Nano에서 실행하는 실험을 진행한다. 두 환경 모두 Ubuntu 18.04 버전을 사용하며, 메시지 타입은 'std_msgs', 'control_msgs', 'trajectory_msgs'를 사용한다. 그림 10(a)는 조이스틱을 이용해 로봇의 말단 좌표를 직교 좌표계 $(0, 0, l_1+l_2+l_3)$ 로 설정했을 때 로봇의 동작 사진이다. 그림 10(b)는 로봇의 말단 좌표를 $(10, 0, 30)$ 로 설정했을 때 로봇의 동작 사진이다. GAZEBO 시뮬레이션 결과와 같은 동작이 실행된 것을 확인할 수 있다.

PC 환경과 임베디드 보드 환경에서의 성능 차이를 확인하기 위해 PC에서 구현한 시스템을 Jetson Nano에서 실행하는 실험을 진행한다. 그림 11은 ROS를 통해 전송하는 관절 각도와 실제 로봇의 관절 각도를 시간에 따라 나열한 그래프이다. 그래프의 점선(Base, Shoulder, Elbow)은 ROS action통신을 통해 로봇에 전달한 관절 각도이며, 실선(Theta_1, Theta_2, Theta_3)은 현재 로봇의 관절 각도를 나타낸다. 그림 11(a)의 그래프를 통해 PC 환경에서 조이스틱을 이용한 제어와 로봇의 동작 사이에 0.4초의 지연시간이 발생한 것을 확인할 수 있다. 처리부와 ROS 통신 과정에서 0.1초의 지연시간이 발생하며 나머지 0.3초는 UR3 자체의 동작에서 발생하는 지연시간이다.



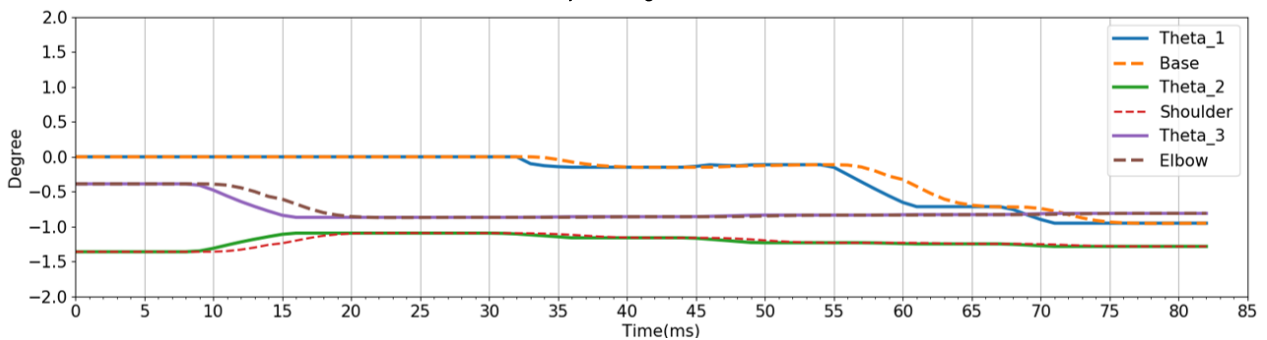
(a) 초기자세 (b) 동작 후 자세

그림 10. UR3 로봇 동작
Fig. 10. Motion of the UR3 robot, (a) Initial pose, (b) Post-movement pose



(a) PC 환경에서 시간에 따른 관절 각도의 변화

(a) Variation of joint angles over time in PC



(b) 임베디드 환경에서 시간에 따른 관절 각도의 변화

(b) Variation of joint angles over

그림 11. 환경에 따른 로봇 동작 지연시간 확인

Fig. 11. Checking the delay time of robot motion according to the environment

그림 11(b)의 그래프를 통해 임베디드 환경에서 실행 시 조이스틱을 이용한 제어와 로봇의 동작 사이에 0.5초의 지연시간이 발생한 것을 확인할 수 있다. 임베디드 보드의 성능 한계로 처리부 계산과정에서 0.1초의 추가 지연시간이 발생하였다.

IV. 결 론

이번 논문에서는 임베디드 환경에서의 ROS 기반 조이스틱을 이용한 산업용 로봇 원격 제어 시스템을 제안하였다. 이번 연구 결과 조이스틱을 이용한 직관적인 산업용 로봇 제어가 가능하며, 터치 패드 사용시 발생하던 공간적 제약 문제를 임베디드 보드 사용한 원격제어로 해결했다. 제안하는 시스템은 ROS 통신을 사용하기에 조이스틱 외에 다양한 제어 수단에도 적용 가능하며, 고성능 임베디드 보드를 사용해 지연시간을 줄일 수 있을 것이다.

Acknowledgement

"2021년도 한국정보기술학회 하계 종합학술대회에서 발표한 논문(조이스틱을 이용한 ROS 기반 산업용 로봇 원격 제어)을 확장한 것임"

References

- [1] William Pagnon, "The 4th industrial revolution-A smart factory implementation guide", International Journal of Advanced Robotics and Automation (IJARA), Vol. 2, No. 2, pp. 1-5, Jun. 2017. <http://dx.doi.org/10.15226/2473-3032/2/2/00123>.
- [2] Thomas Gulde, Dennis Ludl, and Cristóbal Curio, "RoPose: CNN-based 2D pose estimation of industrial robots", 2018 IEEE 14th International Conference on Automation Science and Engineering

(CASE), Munich, Germany, pp. 463-470, Aug. 2018. <https://doi.org/10.1109/COASE.2018.8560564>.

[3] Qujiang Lei, Hongda Zhang, Yang Yang, Yue He, Yang Bai, and Shoubin Liu, "An investigation of applications of hand gestures recognition in industrial robots", International Journal of Mechanical Engineering and Robotics Research, Vol. 8, No. 5, pp. 729-741. Sep. 2019. <http://dx.doi.org/10.18178/ijmerr.8.5.729-741>.

[4] Sangwoo Cho and Jong-Ha Lee, "A Study on the Usability Test Method of Collaborative Robot Based on ECG Measurement", International Conference on Intelligent Human Computer Interaction., Daegu, Korea, pp. 357-362, Feb. 2021.

[5] Paul Bouchier, "Embedded ros [ros topics]", IEEE Robotics & Automation Magazine, Vol. 20, No. 2, pp. 17-19, Jun. 2013. <https://doi.org/10.1109/MRA.2013.2255491>.

[6] João Pedro Carvalho, Marco Aurélio Jucá, Alexandre Menezes, Leonardo Rocha Olivi, André Luis Marques Marcato, and Alexandre Bessa dos Santos, "Autonomous UAV outdoor flight controlled by an embedded system using Odroid and ROS", CONTROLO, Portugal, pp. 423-437, Sep. 2016.

[7] M. M. Bezemer and J. F. Broenink, "Connecting ROS to a real-time control framework for embedded computing", 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, pp. 1-6, Sep. 2015. <https://doi.org/10.1109/ETFA.2015.7301648>.

[8] Ui-Sun Kim, Beom-Su Kim, and Ik-Sang Kim, "Implementation of Serving Mobile Robot Using ROS", Journal of KIIT, Vol. 17, No. 2, pp. 33-43, Feb. 2019. <http://dx.doi.org/10.14801/jkiit.2019.17.2.33>.

[9] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng, "ROS: an open-source Robot Operating System", ICRA workshop on open source software, Vol. 3, No. 2, pp. 5-10, May 2009.

[10] J Denavit and RS Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices", ASME J Applied Mechanics, Vol. 22, pp. 215-221, Jun. 1955. <https://doi.org/10.1115/1.4011045>.

[11] Md. Moin Uddin Atique and Md. Atiqur Rahman Ahad, "Inverse Kinematics solution for a 3DOF robotic structure using Denavit-Hartenberg Convention", 2014 International Conference on Informatics, Electronics & Vision (ICIEV), Dhaka, Bangladesh, pp. 1-5, May 2014. <https://doi.org/10.1109/ICIEV.2014.6850854>.

[12] Nathan Koenig and Andrew Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator", 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), Sendai, Japan, Vol. 3, pp. 2149-2154, Sep. 2004. <https://doi.org/10.1109/IROS.2004.1389727>.

저자소개

권 용 성 (Yong-Sung Kwon)



2021년 2월 : 금오공과대학교
전자공학부 전자및전파전공
(공학사)
2021년 3월 ~ 현재 :
금오공과대학교 IT융복합공학과
석사과정
관심분야 : 지능형로봇 및 제어,
딥러닝, ROS, 인간-로봇 상호작용(HRI) 기술

김 은 수 (Eun-Su Kim)



2020년 2월 : 금오공과대학교
전자공학부 정보전자전공(공학사)
2020년 3월 ~ 현재 :
금오공과대학교 전자공학과
석사과정
관심분야 : 지능형로봇 및 제어,
딥러닝, ROS, 인간-로봇

상호작용(HRI) 기술

신 재 욱 (Jae-Wook Shin)



2008년 2월 : 경북대학교
전자전기컴퓨터공학부(공학사)
2010년 2월 : 포항공과대학교
전자전기공학과(공학석사)
2014년 8월 : 포항공과대학교
전자전기공학과(공학박사)
2014년 3월 ~ 2016년 12월 :

삼성전자 무선사업부 책임연구원

2017년 2월 ~ 2021년 2월 : 순천향대학교

의용메카트로닉스공학과 조교수

2021년 3월 ~ 현재 : 금오공과대학교 전자공학부 조교수

관심분야 : Adaptive filter, 음향신호처리, 생체신호처리,
임베디드 시스템

박 범 용 (Bum-Yong Park)



2009년 2월 : 경북대학교
전자전기컴퓨터공학부(공학사)
2011년 2월 : 포항공과대학교
전자전기공학과(공학석사)
2015년 8월 : 포항공과대학교
전자전기공학과(공학박사)
2017년 8월 : 삼성전자

생산기술연구소 책임연구원

2017년 9월 ~ 현재 : 금오공과대학교 전자공학부,

IT융복합공학과 조교수

관심분야 : Robust control, robot manipulator system,
신호 처리, 임베디드 제어 시스템