

# ROS 기반 비전 시스템을 이용한 마커 위치 추정 및 로봇 매니플레이터 제어

김영빈\*<sup>1</sup>, 최낙원\*<sup>2</sup>, 박범용\*\*

## Marker Position Estimation and Robot Manipulator Control using ROS-based Vision System

Yeon-Bin Kim\*<sup>1</sup>, Nak-Won Choi\*<sup>2</sup>, and Bum-Yong Park\*\*

이 연구는 금오공과대학교 대학 연구과제비로 지원되었음(과제지원년도, 2022년~2023년)

### 요약

최근 로봇을 활용한 사업의 지속적인 성장에 따라 로봇 관련 산업의 영역이 확대되고 있으며, 스마트 공장에서 산업용 로봇이 핵심적인 역할을 수행하고 있다. 그러나 최근 산업 현장에서 요구하는 일반적인 산업 형태뿐만 아닌 고기능화된 산업 현장의 자동화를 위해서 로봇과 주변 환경에 대한 영상 데이터를 이용하여 다양한 연구가 이루어지고 있다. 본 논문에서는 ROS(Robot Operating system) 플랫폼을 통해 식별 마커 위치 추정 정보에 대한 비전 카메라 데이터를 획득하고 이를 기반으로한 로봇 매니플레이터 제어 시스템을 제안한다. 로봇 매니플레이터를 시뮬레이션과 실제 환경을 통해 마커 위치 도달 및 정확도를 측정하였다. 실험 결과 마커까지 도달하는 평균 시간은 6.61초, 마커 좌표와 실제 엔드 이펙터 좌표 간의 RMSE(Root Mean Squared Error)는 0.0125로 나타났다.

### Abstract

Recently, with the continued growth of businesses and sales using robots, the scope of robot-related industries is expanding, and industrial robots are playing a key role in smart factories. However, recently, various studies are being conducted using video data about robots and the surrounding environment to automate not only general industrial types required in industrial sites but also highly intelligent industrial sites. In this paper, we acquire vision camera data for identification marker position estimation information through the Robot Operating System(ROS) platform and propose a robot manipulator control system based on this. The marker position arrival and accuracy of the robot manipulator were measured through simulation and real environments. As a result of the experiment, the average time to reach the marker was 6.61 seconds, and the Root Mean Squared Error(RMSE) between the marker coordinates and the actual end effector coordinates was 0.0125.

### Keywords

robot manipulator, ROS, ArUco marker, vision camera, industrial robot

\* 국립금오공과대학교 IT융복합공학과 석사과정  
- ORCID<sup>1</sup>: <https://orcid.org/0009-0000-2477-8667>  
- ORCID<sup>2</sup>: <https://orcid.org/0009-0003-9164-0635>  
\*\* 국립금오공과대학교 IT융복합공학과 부교수(교신저자)  
- ORCID: <https://orcid.org/0000-0001-9490-0365>

• Received: Jan. 15, 2024, Revised: Feb. 27, 2024, Accepted: Mar. 01, 2024  
• Corresponding Author : Bum-Yong Park  
Dept of IT Convergence Engineering, Kumoh National Institute of Technology, 61, Daehak-ro, Gumi-si, Gyeonngsangbuk-do, 39177 Korea  
Tel.: +82-54-7438, Email: bumyong.park@kumoh.ac.kr

## 1. 서 론

한국로봇산업진흥원과 한국로봇산업협회가 최근 실시한 조사[1]에 따르면, 로봇을 활용한 사업체 및 그에 따른 매출이 지속해서 성장하고 있음을 알 수 있다. 이러한 증가 추세는 제조 및 서비스 분야를 비롯하여 부품 생산과 소프트웨어 개발 등 로봇 사용을 위한 산업 부문에서도 이루어지고 있다. 이는 다양한 영역에서 로봇 활용이 확대되고 있음을 의미한다.

특히, 산업용 로봇은 4차 산업혁명의 핵심 요소로 부각되어 스마트 공장 형태의 산업 현장에서 핵심적인 역할을 수행하고 있다[2]. 일반적으로 산업용 로봇은 사용자의 입력, 조작을 통해 로봇 제어 프로그램을 작성할 수 있는 티칭 펜던트[3]를 사용하여 로봇 부품을 제어한다. 이러한 방식은 단순하고 고정된 형태의 생산, 조립 작업에 적합하다. 하지만 최근 산업 현장에서 요구하는 방식은 IoT, 인공지능 등 첨단 기술을 적용한 스마트 공장 형태의 작업을 추구한다[4]. 이를 위해 산업용 로봇의 동작 방식을 고정된 위치의 물체를 옮기고 조작하는 단순한 형태에서 나아가 로봇이 스스로 주변 환경을 인식하고 그에 따른 작업 방식에 변화를 줄 수 있는 지능을 갖춘 방식으로 변화할 필요가 있다[5].

지능을 갖춘 산업용 로봇은 물리적인 세계에서 얻는 힘/토크 센서나 충돌 감지 센서와 같은 데이터뿐만 아니라 소리나 영상과 같은 정보가 필요하다[6]. 특히 영상 데이터를 활용하면 로봇이 외부 요인과 충돌하기 전에 미리 대비할 수 있거나, 물체의 색상, 크기, 위치 등을 파악할 수 있다[7]. 이러한 정보를 활용하면 산업 현장에서 로봇 동작을 더 다양하게 조정할 수 있다. 그러므로 산업용 로봇에게 영상 데이터를 통해 얻은 정보를 적용하는 연구가 필요하다.

영상 데이터를 통해 얻을 수 있는 정보 중 많은 연구가 이루어지는 분야에는 식별 마커를 통한 위치 추정 방법이 있다. 식별 마커는 카메라를 통해 획득한 주변 환경 데이터에서 감지 알고리즘을 이용하여 자동으로 인식될 수 있는 패턴으로 구성되어 있다[8]. 해당 패턴 이미지는 카메라와 마커 간

매개 변수를 통해 상대적인 3차원 위치 추정이 가능하다[9].

식별 마커 위치 추정 방법은 모바일 매니퓰레이터를 이용한 쌍안 비전 기반 엔드 이펙터 자세 추정 연구[10], EMG 및 식별 마커 위치 추정 기반 원격 조작 프레임워크 개발연구[11], 마커 위치 정보 기반 실시간 연속 로봇 형상 감지 연구[12], 로봇 매니퓰레이터 관절 각도와 위치 추정을 통한 시각적 서브모터 제어 연구[13], 영상 설정을 통한 식별 마커 위치 정확도 측정 실험[14] 등 로봇과 영상 데이터가 상호작용하는 다양한 연구에 사용되고 있다. 하지만 이러한 연구는 실제 산업용 로봇을 기반으로 하여 높은 초기 비용과 유지 및 보수를 필요로 하거나 시뮬레이션을 통해 이루어져 실제 환경에서 획득한 데이터 정보가 부족하다. 또한 카메라를 통해 획득한 위치 정보 데이터를 로봇에 적용하는 과정에 있어서 데이터 파일을 따로 저장하고 불러오는 경우가 발생한다.

본 논문에서는 사용자가 일일이 영상 데이터를 입력하지 않으며, 적은 비용으로 기존 산업용 로봇에도 적용할 수 있는 시스템을 목표로 하였다. 해당 시스템은 저가형 로봇 매니퓰레이터 모델과 컴퓨터 비전 카메라를 통해 얻은 식별 마커 위치 정보 데이터를 통신하여 로봇 엔드 이펙터(End effector) 제어 기반 Pick-and-Place 동작 수행을 통해 구현하였다. 이를 구현하기 위한 다양한 시스템 개발 방법 중 로봇 응용 프로그램 개발에 필요한 하드웨어 제어, 프로세스 간 연결, 패키지 관리 등의 기능을 제공하는 오픈 소스 운영체제인 ROS(Robot Operating System)를 사용하였다. 이를 통해 로봇 매니퓰레이터, 비전 카메라를 PC 환경의 ROS 시스템상에서 연결하고 식별 마커와 상호작용할 수 있게 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 시스템 구성, 로봇 매니퓰레이터 모델, 비전 카메라를 이용한 위치 추정 시스템, ROS 시스템에 관해 기술한다. 3장에서는 앞서 기술된 내용을 바탕으로 시뮬레이션과 실제 로봇 매니퓰레이터 동작 제어에 대한 내용을 다룬다. 4장에서는 결론에 관해 기술한다.

## II. 시스템 구성

시스템 구성은 그림 1과 같이 크게 매니플레이션 동작, 비전, ROS 부분으로 나눌 수 있다. 동작 순서는 먼저, 비전 카메라를 통해 식별 마커의 위치 정보를 획득하고 이를 PC 환경으로 통신한다. 해당 데이터는 ROS 시스템을 통해 로봇 매니플레이터에 전달되고, 로봇 제어 명령에 따라 정해진 동작을 수행한다.

### 2.1 로봇 매니플레이터

로봇 매니플레이터는 ROBOTIS사의 Open-Manipulator-X 모델을 선정하였다. Open-Manipulator-X는 표 1과 같이 5개의 로봇 전용 액추에이터 XM430-W350-T를 사용하여 로봇 자유도를 구성한다.

표 1. Open-Manipulator-X 사양[15]  
Table 1. Specifications of Open-Manipulator-X[15]

Actuator	DYNAMIXEL XM430-W350-T
Input voltage	12V
DOF	5 (4 DOF + 1 DOF Gripper)
Payload	0.5kg
Repeatability	< 0.2mm
Speed(Joint)	46RPM
Weight	0.70kg
Reach	380mm
Gripper stroke	20~75mm

그 중 그리퍼를 제외하면 관절 역할을 하는 액추에이터는 4개이며 이로 인해 Open-Manipulator-X의 엔드 이펙터 제어를 위해서는 4 자유도 로봇 매니플레이터에 대한 좌표계 변환이 필요하다.

그림 2는 Open-Manipulator-X의 자유도 구조체로 직선, 원통을 통해 각각 링크와 관절 표현을 나타내었다. 해당 구조체는 로봇 베이스에 대한 좌표계 ( $X_0, Y_0, Z_0$ )부터 엔드 이펙터의 좌표계 ( $X_e, Y_e, Z_e$ )로 이루어져있다. 이에 대한 좌표계 변환을 위해 D-H(Denavit-Hartenberg) 파라미터가 필요하다. D-H 파라미터는 링크의 길이( $a$ ), 관절 축 비틀림각( $\alpha$ ), 관절 축 회전각( $\theta$ ), 관절 축 오프셋( $d$ )으로 이루어져있다.

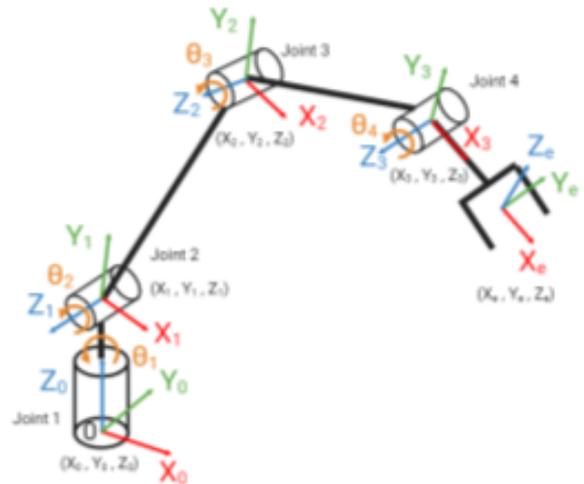


그림 2. Open-Manipulator-X 자유도 구조체  
Fig. 2. Degree of freedom structure of Open-Manipulator-X

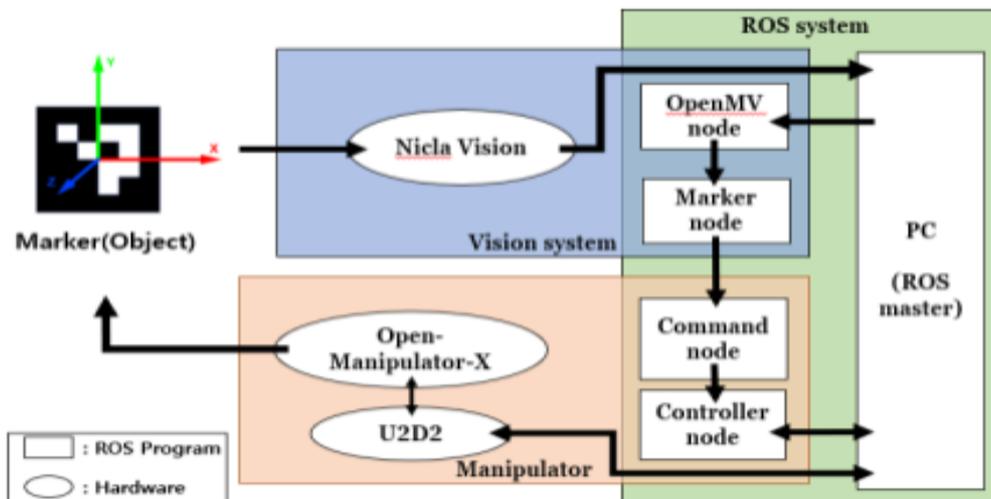


그림 1. 시스템 구성  
Fig. 1. System configuration

표 2. Open-Manipulator-X의 D-H 파라미터

Table 2. D-H Parameter of Open-Manipulator-X

Joint	a (m)	$\alpha$ (°)	$\theta$ (°)	d (m)
1	0	90	$\theta_1$	0.077
2	0.130	0	$\theta_2 + \theta_0$	0
3	0.124	0	$\theta_3 - \theta_0$	0
4	0.126	0	$\theta_4$	0

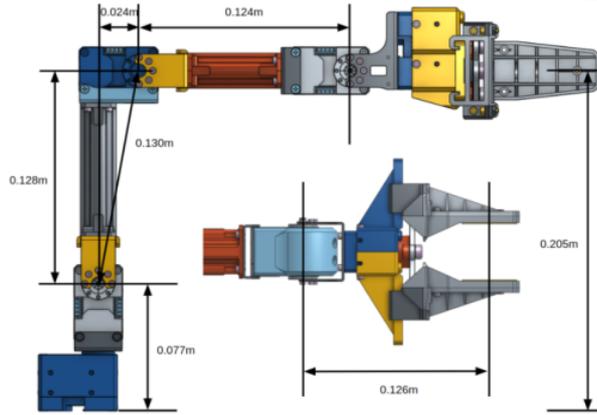


그림 3. Open-Manipulator-X 구성 [15]

Fig. 3. Configuration of Open-Manipulator-X [15]

Open-Manipulator-X의 구성은 그림 3과 같이 이루어졌다. 이를 통해 표 2와 같이 D-H 파라미터를 구할 수 있다. 이때 오프셋 각도  $\theta_0$ 는 식 (1)[16]과 같이 관절 2와 3 사이에 발생하는 관절 축 비틀림을 고려하여 계산한다.

D-H 파라미터를 기반으로 하는 좌표계 변환 행렬은 식 (2)[16]과 같이  $i-1$  프레임에 대한  $i$  프레임의 변환을 통해 구할 수 있다.  $T_1^0, T_2^1, T_3^2, T_e^3$  변환 행렬을 구한 후 각 변환 행렬을 결합하면 로봇 베이스에 대한 엔드 이펙터 좌표계 변환  $T_e^0$ 를 구할 수 있다.

$$\theta_0 = \tan^{-1}\left(\frac{0.024}{0.128}\right) \approx 11^\circ \quad (1)$$

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

## 2.2 비전 시스템

비전 카메라는 시각적 정보를 촬영하고 처리하기 위한 장치로, 이미지나 영상을 촬영하여 이를 디지털 데이터로 변환한다. 본 연구에 사용한 비전 카메라는 Nicla Vision 모델이다[17]. 해당 카메라는 컴퓨터 및 머신 비전 프로그래밍 통합 개발 환경 OpenMV IDE를 통해 내부 메인 스크립트를 수정할 수 있다. 이를 통해 해상도를 640×480으로 설정하고 촬영된 카메라 이미지는 USB 케이블을 통해 시리얼 데이터 형태로 변경하여 PC에 전달된다.

비전 카메라를 이용한 시스템을 구현하기 위해 카메라 캘리브레이션 데이터가 필요하다. 카메라 캘리브레이션이란 3차원 현실을 2차원 이미지 영상으로 투사할 때 필요한 영상 처리 파라미터를 구하는 과정을 뜻한다. 해당 과정을 통해 월드 좌표계와 카메라 좌표계, 이미지 좌표계, 픽셀 좌표계 간의 변환 관계, 카메라 왜곡 계수를 구할 수 있다. 월드 좌표계는 현실 세계의 기준 좌표계로 카메라 좌표계와의 변환 관계는 식 (3)과 같이 구성되어 있다.  $(X_c, Y_c, Z_c, 1)^T$ 는 카메라 좌표계를,  $(X_w, Y_w, Z_w, 1)^T$ 는 월드 좌표계를 나타내며 R과 t는 각각 3×3 회전 행렬과 3×1 이동 변환 행렬을 나타낸다. 이미지 좌표계와 픽셀 좌표계의 변환 관계는 식 (4)와 같이 구성되어 있다.  $(u, v, 1)^T$ 은 픽셀 좌표계를,  $(x, y, 1)^T$ 는 이미지 좌표계를 나타낸다. A 행렬은 카메라 내부 파라미터로  $f_x, f_y$ 는 이미지 좌표계에 대한 x, y 방향의 초점 거리를 나타내고  $\gamma$ 는 비대칭 계수,  $C_x, C_y$ 는 이미지 좌표계의 원점 좌표를 픽셀 좌표계 상에서 나타낸 주점을 나타낸다. 해당 비전 시스템에서 사용한 Nicla Vision 카메라 내부 파라미터는 식 (5)이며 카메라 왜곡 계수 k는 식 (6)과 같다.

비전 카메라 시스템에 사용한 식별 마커는 ArUco 마커이다. ArUco 마커는 실시간 비전 데이터 처리 라이브러리 OpenCV(Open Source Computer Vision) 패키지[18]에서 제공하는 OpenCV ArUco Marker Detection 기능과 앞서 구한 카메라 파라미터를 통해 카메라와 마커 간의 좌표계 변환 관계를 얻을 수 있다.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}, \quad \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = [X_c \ Y_c \ Z_c \ 1]^T \quad (3)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad A = \begin{bmatrix} f_x \ \gamma \ C_x \\ 0 \ f_y \ C_y \\ 0 \ 0 \ 1 \end{bmatrix} \quad (4)$$

$$A = \begin{bmatrix} 1624.547 & 0 & 697.938 \\ 0 & 1458.114 & 235.805 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$k = (0.0823, 0.1427, -0.0052, 0.1201, 0) \quad (6)$$

$$T_{conv} = R(Z, -90^\circ) * R(X, -90^\circ) * T_m^c \quad (7)$$

Nicla Vision 카메라는 그림 4와 같이 Open-Manipulator-X의 Link5를 기준으로 X축 7cm, Z축 6cm 떨어진 곳에 위치하며 이를 Camera Link로 설정하였다. Camera Link는 로봇 매니플레이터 베이스의 좌표계 방향이 일치한 좌표계로 설정하였다.

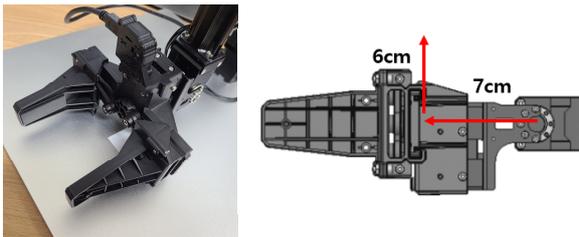


그림 4. Nicla Vision 카메라 부착 위치  
Fig. 4. Attachment location of Nicla Vision camera

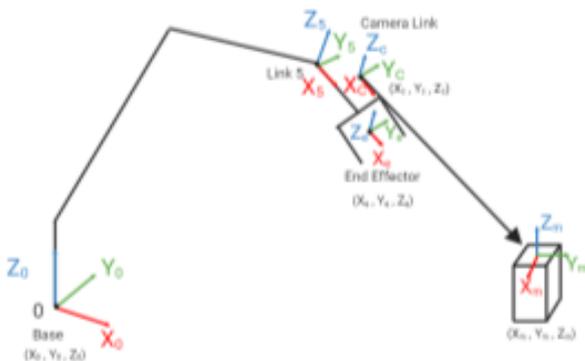


그림 5. 좌표계 변환  
Fig. 5. Coordinate system transformation

이를 통해 그림 5와 같이 로봇 매니플레이터 베이스에 대한 카메라 좌표계 변환  $T_c^0$ 와 앞서 구한 카메라와 마커 간의 좌표계 변환 관계  $T_m^c$ 를 이용하면 로봇 매니플레이터 베이스 좌표계에 대한 ArUco 마커의 상대적인 위치를 알 수 있다. 이때 OpenCV에서 제공하는 기능을 통해 구한 카메라와 마커 간의 좌표계 변환 관계에서 설정된 카메라 좌표계는 로봇 매니플레이터 베이스를 기준으로 설정한 Camera Link 좌표계와 일치하지 않다는 문제가 있다. 그러므로 카메라 좌표계를 일치시키기 위해 식 (7)과 같이 좌표계를 Z축 기준  $-90^\circ$ , X축 기준  $-90^\circ$ 만큼 회전하였다.

### 2.3 ROS 시스템

로봇 운영 체제 ROS는 오픈 소스형 로봇 소프트웨어 플랫폼으로, 로봇 시스템 작성, 제어, 시뮬레이션 등에 사용된다. ROS는 최소 단위 프로그램 Node를 사용하여 Master를 통해 필요한 메시지 데이터를 통신하며 작동한다. ROS의 통신은 비동기적 데이터 교환 형태의 토픽(Topic), 요청과 응답을 기반으로 하여 클라이언트와 서버 간의 상호작용을 하는 서비스(Service), 목표 설정과 중간 상태를 보고하는 액션(Action) 형태로 나뉜다.

본 시스템의 ROS 시스템 구조는 그림 1과 같이 크게 Master 역할의 PC와 4개의 Node로 나눌 수 있다.

OpenMV Node에서는 Nicla Vision으로부터 받은 ArUco 마커 이미지 시리얼 데이터를 ROS에서 사용할 수 있는 메시지 타입으로 처리한다. 해당 데이터는 Marker Node로 전달되어 OpenCV ArUco Marker Detection 기능을 통해 좌표 변환 관계 획득이 이루어진다. 해당 변환 관계와 ROS 시스템 상에서 로봇 매니플레이터의 Link 간 변환, 로봇 상태 정보를 통해 로봇 매니플레이터 베이스에 대한 ArUco 마커의 위치 정보를 획득한다.

Command Node는 로봇 매니플레이터 동작 명령을 담당한다. 해당 Node에는 Motion Planning, 역 기구학 해 풀이, 로봇 제어 기능을 제공하는 MoveIt! 패키지를 통해 Open-Manipulator-X 모델에 대한 엔드 이펙터를 설정하고 로봇 상태 정보를 불러오고 ArUco 마커

가 잘 인식되는 범위에 카메라가 위치하는 “detect” 자세를 설정해준다. 이를 기반으로 하여 앞서 획득한 마커 위치 정보를 통해 시스템에 대한 로봇 매니퓰레이터 동작을 명령한다. 해당 동작은 초기 식별 자세 이동, 마커 위치 도달, Pick-and-Place 동작, 식별 자세 반복으로 이루어진다. 초기 식별 자세 이동 단계에서는 Open-manipulator-X 모델이 앞서 설정한 “detect” 자세를 취하도록 명령한다. 마커 위치 도달 단계에서는 로봇 매니퓰레이터 엔드 이펙터가 ArUco 마커의 중심에 도달하도록 명령한다. ArUco 마커의 위치, 즉, 물체의 상단부에 엔드 이펙터가 위치하여 물체를 잡을 수 있는 범위에 도달하면 그리퍼를 통해 물체를 잡고 지정된 장소로 옮기는 Pick-and-Place 동작 단계가 이루어진다. 일련의 과정이 끝난 후 다시 ArUco 마커를 식별하기 위한 “detect” 자세를 취하는 식별 자세 반복 단계를 명령한다.

Controller Node는 Open-Manipulator-X 모델과 ROS 시스템 간의 동작 제어를 담당한다. 해당 Node에서는 로봇 상태 정보를 ROS에 전달하고 Command Node로부터 받은 동작 명령을 통해 Open-Manipulator-X가 지정된 자세를 취하도록 제어한다.

### III. 실험 결과

#### 3.1 시뮬레이션 환경

본 연구 시스템을 시뮬레이션 하기 위해 Gazebo를 사용하였다. Gazebo는 ROS 플랫폼과 메시지 통신 과정을 통해 가상 환경에서 로봇 모델을 불러와 제어할 수 있는 시뮬레이션 도구이다. Open-Manipulator-X 모델은 ROBOTIS사에서 제공하는 Gazebo 환경상의 로봇 매니퓰레이터 제어 패키지를 사용하면 간단하게 불러올 수 있다. 해당 패키지는 실제 환경과 유사하게 작용하며 마찬가지로 함께 실행되는 Rviz를 통해 가상 환경상의 로봇 매니퓰레이터 상태를 확인 할 수 있다. 이때 Gazebo 환경에서는 Open-Manipulator-X 모델에 맞게 패키지 내부에 주어진 카메라 캘리브레이션 데이터를 사용하기 위해서 Nicla Vision이 아닌 Raspberry PI Camera 모델을 상정한다. 시뮬레이션 환경에서는

로봇 매니퓰레이터 베이스 X축 기준 0.17m 떨어진 위치에 놓인 지름 4.3cm, 높이 7cm 원기둥을 목표 물체로 설정하였다. 해당 물체 위에 3cm 길이의 정사각형 ArUco 마커를 부착하였다. 초기 설정된 “detect” 자세에서 가상 환경 카메라를 통해 ArUco 마커 인식을 바탕으로 그림 6과 같이 위치 정보를 불러온다. 이후 물체를 잡고 로봇 매니퓰레이터 베이스 기준 왼쪽으로 45°, 0.07m 떨어진 장소에 물체를 옮기는 Pick-and-Place 과정을 수행한다. 그 결과 그림 7과 같이 엔드 이펙터가 마커의 위치에 도달하였을 때 중심으로부터 약 0.001m 오차를 보였다.

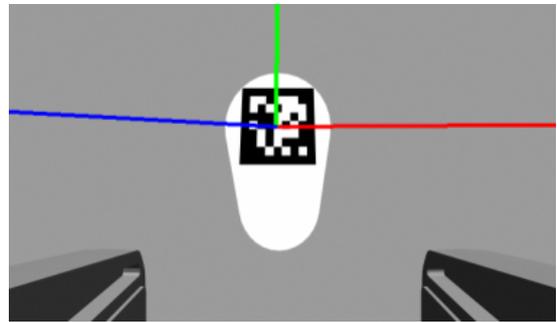


그림 6. 마커 인식  
Fig. 6. Marker recognition

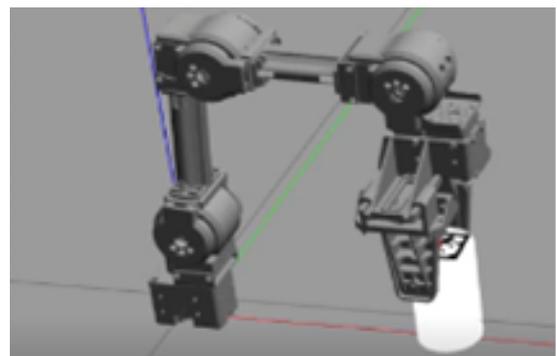


그림 7. 마커 위치 도달  
Fig. 7. Marker position reached

#### 3.2 실제 환경

실제 환경에서 실험 구성은 그림 8과 같이 로봇 매니퓰레이터 베이스로부터 0.15 ~ 0.21m 떨어진 지점에 ArUco 마커를 부착한 가로, 세로, 높이 4.5cm, 4.5cm, 6.5cm의 직육면체 물체를 로봇 매니퓰레이터를 통해 Pick-and-Place 하는 과정으로 이루어져있다.

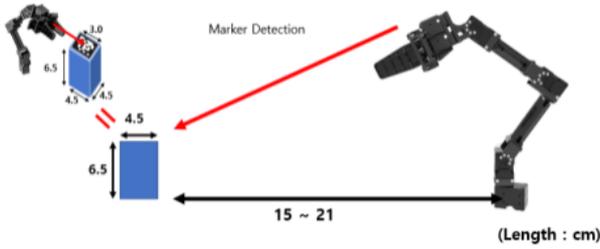


그림 8. 실험 구성  
Fig. 8. Experiment configuration

실험은 마커의 위치와 실제 로봇 엔드 이펙터가 도달한 위치 차이를 측정하였다. 측정은 그림 9와 같이 로봇 매니플레이터 정면을 기준으로 반원을 그려 0°, 90°, 180°방향에서 각각 10회씩 실시하였다. 실험 결과, Aruco 마커 인식부터 로봇 엔드 이펙터의 마커 위치 도달까지 걸리는 시간은 평균 6.61초가 소요되었다.

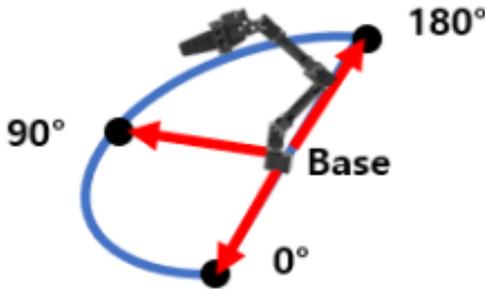


그림 9. 실험 측정 기준  
Fig. 9. Experimental measurement criteria

표 3. 0°기준 측정 결과

Table 3. Measurement results based on 0°

	x1	y1	x2	y2	distance
1	0.1957	0.0049	0.1854	0.0007	0.0111
2	0.1725	0.003	0.1592	-0.0011	0.0139
3	0.1838	0.0037	0.1890	-0.0043	0.0095
4	0.2026	-0.0041	0.1860	-0.0051	0.0166
5	0.1719	0.0045	0.1518	0.0	0.0206
6	0.1808	-0.0108	0.1997	-0.0081	0.0191
7	0.2077	-0.0140	0.2041	-0.0102	0.0052
8	0.1511	-0.0071	0.1665	-0.0043	0.0157
9	0.1927	0.0017	0.1993	0.0037	0.0069
10	0.1677	-0.0047	0.1744	-0.0017	0.0073

표 3은 0°에서 측정한 결과이며 x1, y1은 카메라를 통해 측정한 로봇 매니플레이터 베이스에 대한 ArUco 마커의 x, y 좌표, x2, y2는 마커에 도달한

로봇 엔드 이펙터의 x, y 좌표를 나타낸다. distance는 획득한 두 좌표 간의 유클리드 거리를 의미한다. 그림 10은 표 3의 ArUco 마커 좌표와 로봇 엔드 이펙터 좌표에 대한 그래프이다. 기호를 다르게 표시하여 마커와 로봇 엔드 이펙터를 구분하였으며 각 시도 횟수별 마커와 로봇 엔드 이펙터 순서쌍을 같은 색으로 나타내었다. 0°에서 측정한 마커와 로봇 엔드 이펙터 간 유클리드 거리 평균 제곱근 오차 RMSE(Root Mean Squared Error)는 0.0136이다.

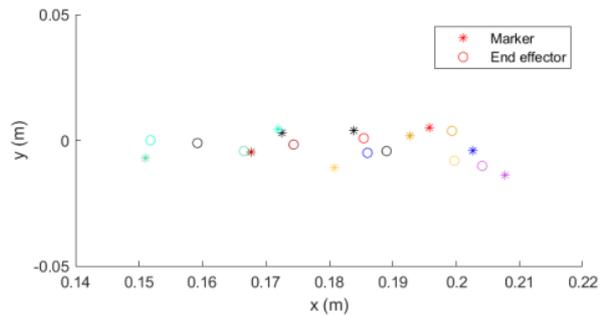


그림 10. 0°기준 측정 결과  
Fig. 10. Measurement results based on 0°

이와 같이 표 4와 그림 11은 90°에서, 표 5와 그림 12는 180°에서 측정한 결과를 나타내며 각각의 RMSE는 0.0117, 0.0121이다. 이를 통해 본 시스템을 통한 로봇 매니플레이터 제어가 비교적 오차가 적은 높은 성능이 보임을 알 수 있다. 또한 영상 설정을 통한 식별 마커 위치 정확도 측정 실험[14]과 달리 카메라와 마커간 거리 측정뿐만 아닌 로봇 엔드 이펙터의 마커 위치 도달을 통해 직접적으로 차이를 관찰할 수 있었다.

표 4. 90°기준 측정 결과

Table 4. Measurement results based on 90°

	x1	y1	x2	y2	distance
1	0.1589	-0.0072	0.1771	-0.0026	0.0188
2	0.1703	-0.0016	0.1745	0.0035	0.0065
3	0.1586	-0.0060	0.1636	0.0023	0.0096
4	0.1807	-0.0117	0.1958	-0.0083	0.0155
5	0.1817	-0.0024	0.1719	0.0047	0.0122
6	0.1827	-0.0008	0.1940	0.0011	0.0114
7	0.1848	0.0028	0.1887	0.0044	0.0042
8	0.1784	0.0094	0.1933	0.0121	0.0151
9	0.1664	-0.0006	0.1759	0.0026	0.0100
10	0.2027	-0.0144	0.2075	-0.0137	0.0048

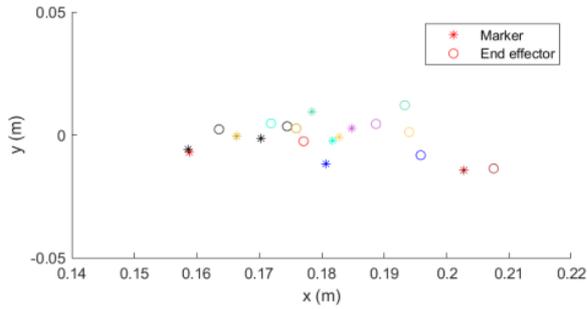


그림 11. 90°기준 측정 결과  
Fig. 11. Measurement results based on 90°

표 5. 180°기준 측정 결과

Table 5. Measurement results based on 180°

	x1	y1	x2	y2	distance
1	0.1659	-0.0084	0.1694	-0.0022	0.0071
2	0.1853	0.0024	0.2110	0.0051	0.0258
3	0.1690	0.0017	0.1792	0.0043	0.0105
4	0.1898	-0.0004	0.2015	0.0019	0.0120
5	0.1688	0.0062	0.1758	0.0125	0.0094
6	0.1736	-0.0011	0.1820	0.0031	0.0094
7	0.1597	0.0064	0.1633	0.0090	0.0044
8	0.1723	-0.0067	0.1835	-0.0040	0.0115
9	0.1698	-0.0017	0.1815	-0.0006	0.0118
10	0.1808	0.0072	0.1816	0.0129	0.0057

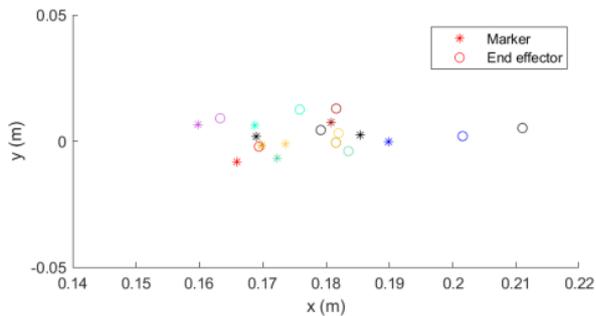


그림 12. 180°기준 측정 결과  
Fig. 12. Measurement results based on 180°

#### IV. 결 론

본 논문에서는 ROS를 기반으로 ArUco 마커의 정보를 비전 카메라를 통해 측정하고 로봇 매니퓰레이터를 제어하는 시스템을 제안하였다. ROS 플랫폼을 통해 시스템의 부분별로 Node를 나눠 시스템을 구성하였으며 OpenCV의 ArUco 마커 인식을 통해 마커 정보를 획득하여 로봇 매니퓰레이터의 Pick-and-Place 동작을 수행하였다. 실험을 통해

마커 추정 위치와 로봇이 도달한 위치 간의 차이를 측정하였으며 유클리드 거리 평균 제곱근 오차 RMSE가 0.0136이라는 준수한 결과를 보였다. 제안하는 시스템은 로봇 매니퓰레이터, 비전 카메라 모델 변경, 카메라 해상도 변화를 통해 정확도 향상을 기대해 볼 수 있다.

#### References

- [1] <https://kiria.org/portal/policysut/portalPlcyReportsDetail.do?jsessionid=684D4232FE6445C607DDB77A359A78B1> [accessed: Apr. 24, 2024]
- [2] N. Mohamed, J. Al-Jaroodi, and S. Lazarova-Molnar, "Industry 4.0: Opportunities for Enhancing Energy Efficiency in Smart Factories", 2019 IEEE International Systems Conference (SysCon), Orlando, FL, USA, pp. 1-7, Apr. 2019. <https://doi.org/10.1109/SYSCON.2019.8836751>.
- [3] M. Gao, L. Ye, and Y. Yan, "Design and Implementation of Teach Pendant for Six Degrees of Freedom Industrial Robot", 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Qinhuangdao, China, pp. 1929-1933, Sep. 2015. <https://doi.org/10.1109/IMCCC.2015.409>.
- [4] R. Jain, M. N. Zafar, and J. C. Mohanta, "Modeling and analysis of articulated robotic arm for material handling applications", IOP Conference Series: Materials Science and Engineering, Vol. 691, No. 1, Dec. 2019. <https://doi.org/10.1088/1757-899X/691/1/012010>.
- [5] M. Plappert, et al., "Multi-goal reinforcement learning: Challenging robotics environments and request for research", arXiv:1802.09464, Feb. 2018. <https://doi.org/10.48550/arXiv.1802.09464>.
- [6] K.-J. Kwak, D.-Y. Kim, and J. Park, "Design of Voice Control Solution for Industrial Articulated Robot", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 21, No. 2,

- pp. 55-60, Apr. 2021. <https://doi.org/10.7236/JIIBC.2021.21.2.55>.
- [7] J. Kim and J. Jeong, "Design and Implementation of OPC UA-based Collaborative Robot Guard System Using Sensor and Camera Vision", *The Journal of The Institute of Internet, Broadcasting and Communication*, Vol. 19, No. 6, pp. 47-55, Dec. 2019. <https://doi.org/10.7236/JIIBC.2019.19.6.47>.
- [8] M. Fiala, "ARTag, a fiducial marker system using digital techniques", 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, Vol. 2, pp. 590-596, 2005. <https://doi.org/10.1109/CVPR.2005.74>.
- [9] M. Kalaitzakis, et al., "Fiducial Markers for Pose Estimation", *Journal of Intelligent & Robotic Systems*, Vol. 101, No. 4, Mar. 2021. <https://doi.org/10.1007/s10846-020-01307-9>.
- [10] J. Wang, Z. Gong, S. Yu, and B. Tao, "A Pose Estimation and Optimization Method for Mobile Manipulator's End-effectors Based on Stereo Vision and ICoP Algorithm", 2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Shanghai, China, pp. 744-749, Nov. 2021. <https://doi.org/10.1109/M2VIP49856.2021.9665072>.
- [11] A. Dwivedi, G. Gorjup, Y. Kwon, and M. Liarokapis, "Combining Electromyography and Fiducial Marker Based Tracking for Intuitive Telemanipulation with a Robot Arm Hand System", 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), New Delhi, India, pp. 1-6, Oct. 2019. <https://doi.org/10.1109/RO-MAN46459.2019.8956456>.
- [12] J. Li, Y. Sun, H. Su, G. Zhang, and C. Shi, "Marker-Based Shape Estimation of a Continuum Manipulator Using Binocular Vision and Its Error Compensation", 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, pp. 1745-1750, Oct. 2020. <https://doi.org/10.1109/ICMA49215.2020.9233616>.
- [13] I. P. Vieira, A. A. Neto, and L. A. Mozelli, "Fiducial Markers Applied for Pose Tracking of a Robotic Manipulator: Application in Visual Servoing Control", 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium, Recife, Brazil, pp. 287-292, Oct. 2016. <https://doi.org/10.1109/LARS-SBR.2016.55>.
- [14] D. C. Popescu, M. O. Cernaianu, P. Ghenuche, and I. Dumitrache, "An assessment on the accuracy of high precision 3D positioning using planar fiducial markers", 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, pp. 471-476, Oct. 2017. <https://doi.org/10.1109/ICSTCC.2017.8107079>.
- [15] [https://emmanual.robotis.com/docs/en/platform/open\\_manipulator\\_x/specification/#hardware-specification](https://emmanual.robotis.com/docs/en/platform/open_manipulator_x/specification/#hardware-specification) [accessed: Dec. 07, 2023]
- [16] A. Singh, A. Singla, and S. Soni, "D-H parameters augmented with dummy frames for serial manipulators containing spatial links", *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Edinburgh, UK, pp. 975-980, Aug. 2014. <https://doi.org/10.1109/ROMAN.2014.6926379>.
- [17] "Nicla Vision", <https://docs.arduino.cc/hardware/nicla-vision> [accessed: Dec. 15, 2023]
- [18] A. Das, M. W. Ansari, and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV", 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, pp. 1-5, Dec. 2020. <https://doi.org/10.1109/INDICON49873.2020.9342585>.

저자소개

김 영 빈 (Yeong-Bin Kim)



2023년 2월 : 금오공과대학교  
전자공학부 정보전자전공(공학사)  
2023년 3월 ~ 현재 :  
금오공과대학교 IT융복합공학과  
석사과정  
관심분야 : 지능형로봇 및 제어, ROS,  
인간-로봇 상호작용(HRI) 기술

최 낙 원 (Nak-Won Choi)



2023년 2월 : 금오공과대학교  
전자공학부 정보전자전공(공학사)  
2023년 3월 ~ 현재 :  
금오공과대학교 IT융복합공학과  
석사과정  
관심분야 : 지능형로봇 및 제어, ROS,  
인간-로봇 상호작용(HRI) 기술

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 부교수

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