

# ROS를 이용한 로봇 팔의 물체 검출 및 작업 제어

구모세, 고영준, 김규태, 박명숙, 김상훈\*  
 한경대학교 ICT로봇공학전공  
 \*e-mail: kimsh@hknu.ac.kr

## Object Detection and Operation Control of Robot Arm using ROS

Mo-Se Koo, Young-Jun Go, Kyu-Tae Kim, Myeong-Suk Park, Sang-Hoon Kim  
 Dept of ICT Robot Engineering, Hankyong National University

### 요 약

본 논문에서는 서비스용 다관절 로봇팔의 주요기능인 다양한 컵의 검출과 조작, 이동등을 지능적으로 구현하기 위해 로봇 오픈소스 운영체제인 ROS(Robot Operation System)을 기반으로 관련 프로그램을 개발하고 기능을 구현하였다. 연구의 주요 목표인 다양한 종류의 컵, 병과 같은 물체를 실수 없이 집어서 옮기기 위한 처리과정으로 관심물체인식, 3D좌표생성, 결과데이터의 역 기구학 해석등을 수행하였으며, 이를 통해 각 기구부의 축들이 물체에 정확히 도달하고 동작의 오류를 최소화하기 위해 ROS기반의 6축 서비스 로봇팔을 활용한 경로 생성과정과 물체의 검출 성능 과정 및 실험등을 제시하였다.

### 1. 서론

로봇 팔은 인간의 팔과 유사한 동작을 제공하며 단순 반복 작업이 이뤄지는 공장에서 사용되기 위해 제작된 기계적인 장치이다. 주로 고정된 위치에서 반복적인 작업을 대체하거나 사람이 하기 위험한 일을 대신하는 보조 역할의 산업용 로봇으로 많은 쓰임이 있었지만 [1] 최근에는 로봇 팔을 서비스 분야에 활용하려는 노력이 이루어지고 있다. 서비스 로봇 시장이 확대되고, 3d 프린팅 기술의 합세로 일반인들에게 로봇팔에 대한 접근성이 좋아짐에 따라 다양한 작업을 수행하는 역할의 쓰임도 기대할 수 있는 상황이다. [2]

본 논문에서는 직접 설계한 로봇 팔의 서비스적인 임무 수행을 위해 목표로 하는 컵, 병과 같은 물체를 집어 옮기는 동작을 하도록 관심물체를 검출하고 3D좌표를 생성하여 이를 역 기구학 해석을 통해 각 축이 물체에 도달하기 위한 경로 생성과정이 필수적이다. 따라서 로봇 응용 프로그램 개발을 위한 오픈소스 운영체제인 ROS (Robot Operation System)을 활용하여 설계한 6축 서비스용 로봇 팔을 바탕으로 제어를 위한 환경 구축과 프로그램 개발을 수행하였다. 이를 통해 고정된 위치의 물체가 아닌 임의의

위치에 존재하는 물체를 검출해 집어 옮기는 임무 수행이 가능한 로봇 팔을 ROS환경에서 제어해 실제 카페나 패스트푸드점과 같은 서비스 분야에서 활용 가능성을 검증해볼 수 있다.

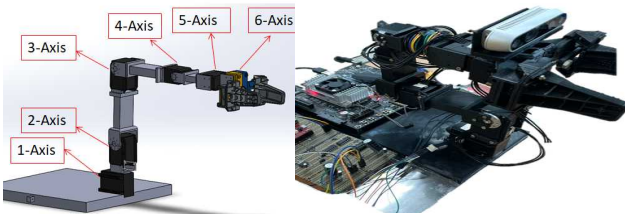
### 2. 설계한 로봇팔의 스펙과 물체 검출

#### 2.1 로봇 팔의 설계 사항

실내 환경에서 물체를 집어서 옮기는 작업을 수행할 수 있도록 설계한 로봇팔의 하드웨어 최종 스펙은 <표 1>과 같고, (그림 1)은 로봇 팔 3D 설계 모델링이다.

items	contents
Reach	484.8mm
DOF	5+1(yaw-pitch-pitch-pitch-yaw-yaw)
Payload	1kgf
Weight	1Kg
Gripper Stroke	2~10cm 이내
Speed(Joint)	약 20Rpm
Main Controller	Cortex-m4, Jetson-TX2
Camera minimum range	11cm
working time	total : 45s(one job)
actuator	Dynamixel

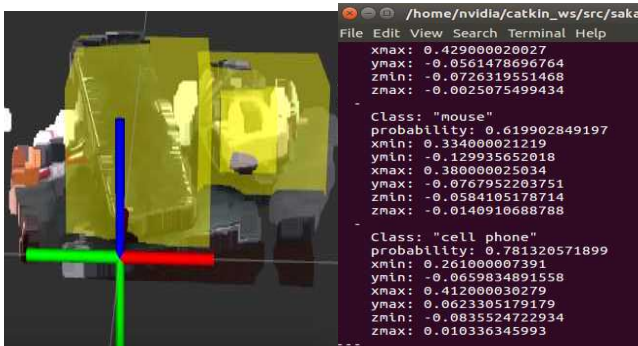
<표 1> 하드웨어 최종스펙



(그림 1). 로봇 팔 3D 설계 모델링과 실제 로봇 팔

## 2.2 darknet\_ros\_3d를 이용한 물체 검출

로봇 팔이 물체를 집어 옮기는 작업을 수행하기 위해선 먼저 카메라를 통한 물체검출이 필수적이다. 또한, 매번 다른 곳에 있는 물체를 상대하기 때문에 정확한 물체검출뿐만 아니라 정확한 물체의 3D 좌표를 얻어야 한다. 따라서 물체의 이미지와 depth 정보를 얻을 수 있는 3D카메라(Realsense-D435, Intel)를 활용하였다. 물체의 3D 좌표를 얻는 방법으로는 다양한 방법이 있지만, ROS환경에서 Moveit과 연동하기 쉽고, 인식률도 로봇의 목표로 하는 작업을 수행하기에 충분한 yolo를 사용한 ROS오픈소스인 darknet\_ros\_3d[3]를 이용하여 물체 검출을 진행하였다. darknet\_ros\_3d은 ROS에서 darknet을 사용할 수 있도록 지원하는 오픈소스인 darknet\_ros를 통해 yolo를 사용하여 얻은 darknet\_ros(yolo)의 결과 topic과 3D카메라의 point cloud 기반 정보를 매치시켜주면 해당 물체의 3D bounding box topic을 제공한다. 해당 정보에는 물체의 이름, 확실성의 확률, box의 x,y,z max,min값이 포함되어 있다. (그림 2)은 darknet\_ros\_3d로 인식한 물체의 정보와 해당 3D bounding box를 Rviz에서 확인해본 그림이다. coco data set을 기반으로 yolov3\_tiny weight파일을 사용하였으며, 관심물체인 cup, bottle을 검출하기에 충분한 인식률을 보이는 것을 확인하였다.

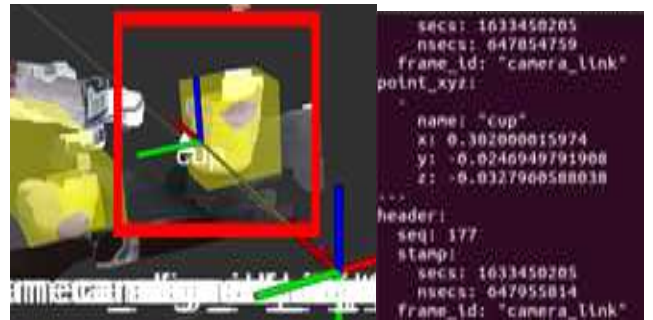


(그림 2) darknet\_ros\_3d로 검출한 bounding box

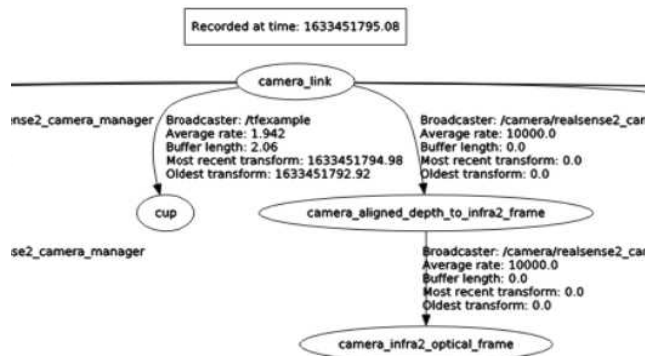
## 2.3 검출한 물체의 정보 가공과 3D좌표 생성

darknet\_ros\_3d를 통해 검출한 여러가지 물체 중 로

봇의 작업영역 안에 있는 관심물체만의 정보를 가지고 3D 좌표생성을 진행하기 위해 3D bounding box topic의 class, probability와 카메라 기준 40cm 이내의 물체의 정보만 획득해, 획득한 x,y,z의 max min 값으로 해당 bounding box의 중심점을 구하였다. Moveit으로 제어 및 플래닝을 할때 목표지점인 물체의 중심점까지 각 축의 joint 값을 계산하도록 하기 위해선, 구한 중심점을 바탕으로 로봇팔의 기준 좌표계에서 물체의 중심점까지의 position이 필요하기 때문에 ros의 좌표변환인 tf[4]를 사용해 tf publisher를 만들어 중심점을 tf frame으로서 다룰 수 있게 해주어 tf lookup을 통해 로봇팔의 기준 좌표계에서 관심물체의 중심점과의 좌표변환까지의 position을 쉽게 구할 수 있도록 하였다. (그림 3)은 검출한 관심물체에만 tf 좌표가 생성되는 그림이다.



(그림 3) 검출한 관심물체의 tf 좌표생성



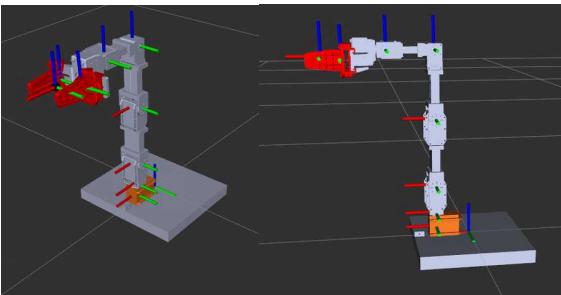
(그림 4) ROS rqt의 tf tree

## 3. 로봇 팔 통합 라이브러리 Moveit의 활용

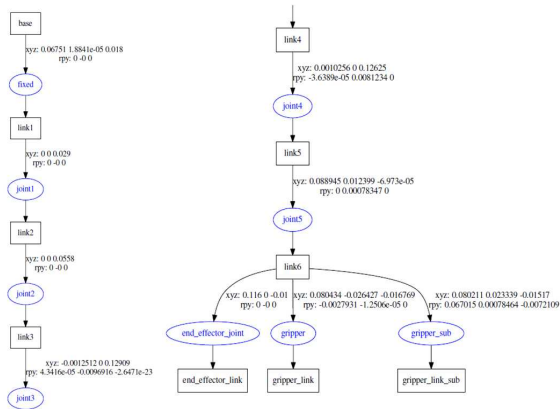
### 3.1 로봇팔 모델링을 위한 URDF작성

ROS 환경에서 역 기구학 해석을 위해 사용하는 Moveit을 활용하기 전에 로봇 모델링을 위한 시각화 툴인 Unified Robot Description Format(URDF) 파일작성이 필수적이다. URDF파일은 XML을 통해 손쉽게 제작할 수 있고, [5] 로봇 모델 정보들을 기록해두고, 해당 정보를 필요로 하는 다른 패키지, 노드에서 활용하게 된다. URDF의 세부내용으로는 로

봇을 구성하는 하나의 구성요소를 링크(Link)와, 링크와 링크를 연결하는 연결부(Joint)의 조합으로 작성된다. <link> 태그에 충돌, 시각화, 관성 정보를 기입하고, <joint> 태그에 관절의 운동형과 위치, 부모 링크와 자식링크, 동작의 제한 등을 기술한다. 설계한 파트들을 URDF 파일 형식에 맞춰 제작하고, 이를 Rviz에서 확인하였다. (그림 5)는 URDF로 정의한 매니플레이터의 모델링을 Rviz에서 확인한 것이고, (그림 6)는 매니플레이터의 URDF의 link와 joint의 관계를 나타내는 블록 다이어그램이다. 또한 URDF에 몇 가지 시뮬레이션 환경에서 사용될 태그들을 추가하면 Gazebo 시뮬레이션에서도 로봇 모델을 적용 해볼 수 있다. 따라서 (그림 7)과 <표 2> 같이 실제 모델에 적용하기 전 Gazebo 가상 시뮬레이션 환경에 적용해 설정한 각 축의 한계 각도 등을 수정해 문제점들을 확인해 실제 로봇을 제작하기 전 작업 반경과 각축의 한계 각도를 정해 놓음으로써 목표로 하는 설계에 타당성을 검증하였다.



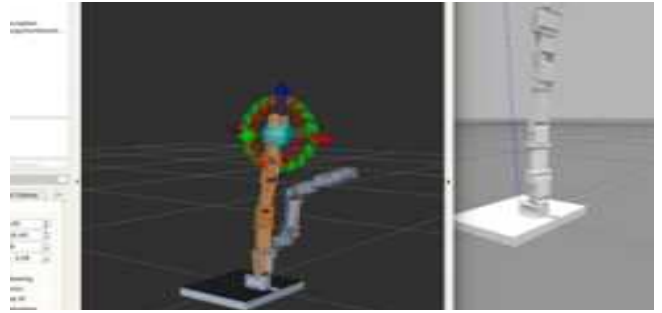
(그림 5) URDF로 정의한 매니플레이터의 모델링



(그림 6) 로봇팔의 URDF의 블록 다이어그램

Joint Limit (radian)	joint 1	joint 2	joint3	joint 4
lower	-3.14	-1.57	-3.14	-2.0
upper	3.14	1.57	0	1.8

<표 2> 각 축의 한계각도



(그림 7) Gazebo 환경에서의 robot model

### 3.2 Moveit의 활용한 실시간 역 기구학 해석

자체 제작한 로봇 팔의 위치 및 자세 제어를 위해 역 기구학 해석, 모션 및 경로 계획, 충돌 회피 등과 같은 복잡한 제어 과정을 필요로 한다. 이러한 부분을 해결하기 위해 Moveit을 활용하는데 MoveIt은 로봇 팔을 움직이기 위한 소프트웨어 프레임 워크로 KDL(Kinematics and Dynamics Library)과 OMPL(The Open Motion Planning Library)등의 오픈 라이브러리를 제공하고, 충돌 감지, 팔 이동 궤적 계획 등 역 기구학을 해석하고 경로를 생성하는 과정을 시각적으로 표현할 수 있고 시뮬레이션을 과정을 수행할 수 있다.[6] 작성한 URDF파일을 바탕으로, moveit setup assistant를 통해 설계한 로봇 팔의 Moveit 패키지를 작성하였고, 역 기구학 및 경로 계획을 Moveit C++ API인 Robot Model and State를 이용해 목표로 하는 관심물체의 중심점인 tf 좌표를 목표지점으로 설정해 IK(Inverse Kinematics) 해결의 시도횟수, 각 시도의 제한 시간 등을 설정해 해당 물체로의 역 기구학 해석을 통한 각 Joint 값을 도출하였다. 도출한 각도 값을 ROS serial node를 통해 MCU로 전달하여 로봇이 물체를 집도록 한다. (그림 8)은 ROS moveit을 통한 역 기구학 해석에 따른 각 축의 joint 값이다.

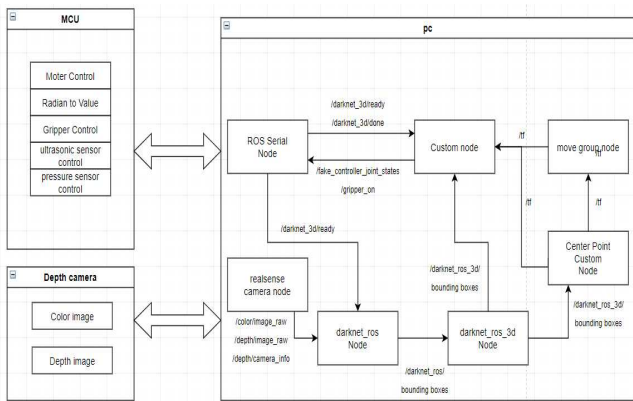


(그림 8) 역 기구학 해석에 따른 각 축의 Joint 값

## 4. 로봇 팔을 활용한 서비스 시나리오 구현

### 4.1 ROS에서의 전체 node의 동작 순서 및 구성

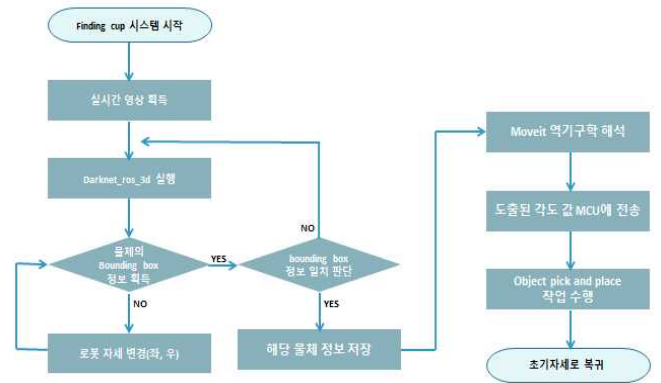
엔드이펙터 위치에 부착된 3D카메라에서 이미지 정보와 거리정보들을 카메라 노드를 실행시켜 얻어와 딥러닝 실시간 객체 탐지 yolo 알고리즘을 사용한 물체검출 노드인 darknet ros와 darknet ros\_3d노드로 물체 검출 및 3d bounding box 정보를 획득한다. 이를 받아 관심물체의 판별과 해당 중심점을 연산해 구하고 해당 좌표의 tf를 발행하는 center point custom node를 작성하였다. 3D좌표를 받아 해당 물체에 도달하기 위한 Moveit 역 기구학 해석을 진행해 각 축의 joint 값을 도출하게 되면 joint 값을 이를 rosserial 노드에 전달하여 MCU에서 모터를 구동해 로봇 팔이 움직이게 되고, 물체에 로봇 팔의 그리퍼가 위치하게 되면 그리퍼의 움직임을 명령을 rosserial 노드에 전달하여 그리퍼의 동작을 수행할 수 있게 하는 custom node를 작성하였다. (그림 9)는 전체적인 ROS환경에서의 노드 간 데이터 통신을 블록화해 다이어그램으로 나타낸 것이다.



(그림 9). 노드 간 데이터 통신

#### 4.2 Finding cup 알고리즘

Finding cup 알고리즘은 로봇의 3D카메라 시야에 관심 물체가 없을 경우 자세를 바꿔 물체를 찾는 알고리즘이다. bounding box의 결과 값 중 cup의 class의 probability가 60 이상의 조건을 만족하면, 해당 중심점을 계산해 3D 좌표를 추출하는 node를 작성하였다. 만약 3D카메라로 획득한 영상 이미지에 관심물체가 없는 경우 bounding box 메시지가 없기 때문에, 이때 물체를 찾기 위해 로봇팔의 자세를 변경하는 알고리즘으로 다양한 위치에 물체들을 로봇 팔이 동작할 수 있도록 하는 역할로 활용된다. (그림 10)은 Finding cup 알고리즘 흐름도와 물체를 집어 옮기는 작업을 수행할 때의 전체적인 동작 흐름도이다.



(그림 10). Find cup 알고리즘 동작 흐름도

#### 5. 결론

주요기능인 다양한 컵의 검출과 조작, 이동 등을 목적으로 설계한 서비스용 다관절 로봇팔을 지능적으로 구현하기 위한 처리과정인 관심물체 인식, 3D 좌표 생성, 결과 데이터의 역 기구학 해석 등을 수행하기위해 ROS를 이용해 프로그램과 관련 기능을 구현하였으며, 이를 통해 각 기구부의 축들이 물체에 정확히 도달하고 동작 수행 중 발생할 수 있는 돌발상황에 대한 오류를 최소화 하기위한 알고리즘 설계 및 실험 등을 제시하여 실제 카페 또는 패스트푸드점에서 활용 가능함을 검증해 볼 수 있다.

#### 감사의 글

이 논문은 2020년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임(No.2020R1F1A1067496)

[1] PYO Y.S, JO H.C, JUNG R.W, LIM T.H, 2017, ROS robot programming, Bucheon, Ruby paper press.  
 [2] DavidGreenfield, 2015, Inside the Human-Robot Collaboration Trend, <https://www.automationworld.com/inside-the-humanrobot-collaboration-trend>, viewed on September 24  
 [3] Francisco MartínRico,Fernando GonzálezRamos [https://github.com/IntelligentRoboticsLabs/gb\\_visua\\_l\\_detection\\_3d](https://github.com/IntelligentRoboticsLabs/gb_visua_l_detection_3d), viewed on September 01  
 [4] Ros wiki, <http://wiki.ros.org/tf>, viewed on September 04  
 [5] ROS wiki, <http://wiki.ros.org/urdf>, viewed on September 10.  
 [6] MoveIt, <https://moveit.ros.org>, viewed on 08 April