모듈형 서비스 매니퓰레이터의 제어를 위한 ROS 환경 설계 방법

구모세, 김상훈
국립한경대학교 전기전자제어공학과
yenuj2@hknu.ac.kr, kimsh@hknu.ac.kr

ROS Configuration Method for Effective Control of Modular Service Manipulator

Mose Koo, Sang-Hoon Kim
Dept. of Electrical, Electronic & Control Engineering, HanKyoung National University

요 약

본 연구에서는 서비스 역할을 수행하는 6축 모듈형 매니퓰레이터 개발을 목표로 하며, 최종 기술 사양에 따른 설계를 진행하는 과정에서 기구의 섬세한 동작을 효율적으로 제어하기 위해 로봇 제어 소프트웨어의 오픈소스 환경인 ROS를 사용한다. 매니퓰레이터의 동작 설계를 ROS 기반에서 제어하기 위해 중요한 기본 환경을 구축하였으며, 특히 로봇 모델링을 위한 시각화를 위해 URDF파일에 해당 매니퓰레이터의 필수 파라미터값들을 지정하여 적용하였고, 전체 동작 시나리오에 맞춰 매니퓰레이터가 특정 자세를 취할 경우의 역기구학적인 해석과 그에 따른 경로를 생성하도록 매니퓰레이터의 라이브러리인 MoveIt을 활용하여 시각적으로 표현하고 시뮬레이션을 수행하였다. 또한, 설계한 ROS 환경 설계 방법을 바탕으로 MCU와의 통신을 통해 모터의 실시간 각도 값을 제어하고, 3D 깊이 카메라의 거리정보와 이미지 정보의 융합을 통해 로봇의 서비스 내용의 개선을 기대할 수 있다.

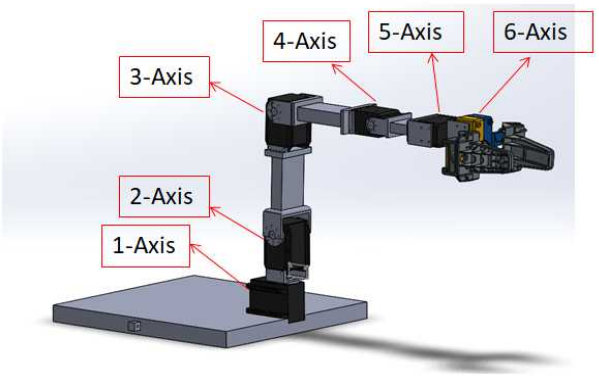
1. 서론

매니퓰레이터(Manipulator)는 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], 최근에 와서는 매니퓰레이터와 사람과의 협업을 목표로 하는 협동 로봇의 연구가 많이 진행되고 있다. [2] 또한, 최근에는 공장뿐만 아니라 매니퓰레이터를 서비스 분야에 활용하려는 노력이 이루어지고 있다. 서비스 로봇 시장이 확대되고, 3d 프린팅 기술의 합세로 일반인들에게 매니퓰레이터에 대한 접근성이 좋아짐에 따라 다양한 작업을 수행하는 역할의 쓰임도 기대할 수 있는 상황이다.

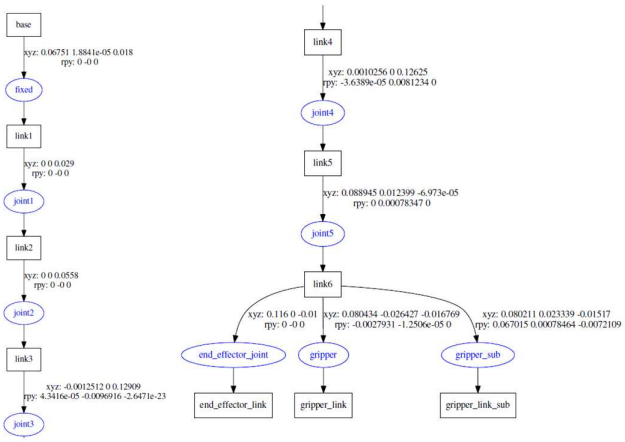
본 논문에서는 로봇 응용 프로그램 개발을 위한 오픈소스 운영체제인 ROS (Robot Operation System)을 활용하여 6축 모듈형 서비스용 매니퓰레이터 제작과 제어를 위한 환경 구축을 수행하였다. 2장에서는 자체 설계한 매니퓰레이터의 스펙, 3장에서는 ROS에서 시각화 모델링과 MoveIt 패키지를 작성하기 위해 필요한 URDF파일에 작성에 대한 설명, 4장에서는 작성한 URDF파일을 바탕으로 MoveIt setup assistant를 사용해 만든 패키지를 바탕으로 거동 시뮬레이션을 진행하였으며, 5장에서는 결론과 향후 연구 방향에 대해서 기술하였다.

2. 매니퓰레이터 설계 사항

실내 환경에서 베이스 기준 작업 반경 500mm 내의 물체를 집어서 옮기는 작업을 수행할 수 있고 다양한 시나리오에 유연한 동작이 가능하도록 6축으로 매니퓰레이터를 설계하였다. 안정적으로 물체를 집기 위하여, 각 모터 용량을 계산하여 가반 하중을 0.6 kg으로 설계하였다. 매니퓰레이터의 그리퍼 외형은 ROBOTIS사의 Open manipulator X를 참고하여 만들었으며, 각 축에 사용하는 모터는 ROBOTIS사의 Dynamixel 모터를 사용하였다. 그림 1은 매니퓰레이터 3D 설계 모델링이다.



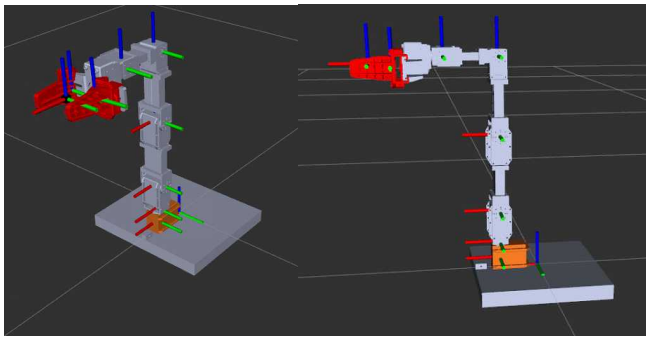
<그림 1> 매니퓰레이터 3D 설계 모델링



<그림 3> 매니퓰레이터의 URDF의 블록 다이어그램

3. 매니퓰레이터 모델링을 위한 URDF작성

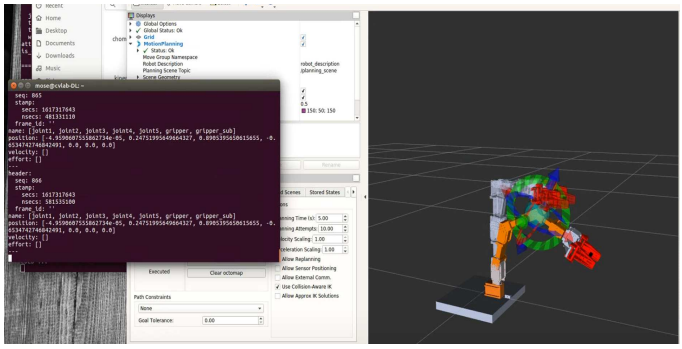
ROS 환경에서 로봇 모델링을 위한 시각화 툴로써 Unified Robot Description Format(URDF) 파일을 XML을 통해 손쉽게 제작할 수 있다. [3] 로봇 모델 정보들을 기록해두고, 해당 정보를 필요로 하는 다른 패키지, 노드에서 활용하게 된다. URDF의 세부내용으로는 로봇을 구성하는 하나의 구성요소를 링크(Link)와, 링크와 링크를 연결하는 연결부(Joint)의 조합으로 작성된다. <link> 태그에 충돌, 시각화, 관성 정보를 기입하고, <joint> 태그에 관절의 운동형과 위치, 부모 링크와 자식링크, 동작의 제한 등을 기술한다. 설계한 파트들을 URDF 파일 형식에 맞춰 제작하고, 이를 ROS 시각화 툴인 Rviz(ROS Visualiztion)에서 확인하였다. 또한 URDF에 몇 가지 시뮬레이션 환경에서 사용될 태그들을 추가하면 Gazebo 시뮬레이션에서도 로봇 모델을 spawn 해볼 수 있다. 그림 2는 URDF로 정의한 매니퓰레이터의 모델링을 Rviz에서 확인한 것이고, 그림 3은 매니퓰레이터의 URDF의 link와 joint의 관계를 나타내는 블록 다이어그램이다.



<그림 2> URDF로 정의한 매니퓰레이터의 모델링

4. 매니퓰레이터 통합 라이브러리 MoveIt의 활용

자체 제작한 매니퓰레이터 로봇의 위치 및 자세 제어를 위해 역기구학 해석, 모션 및 경로 계획, 충돌 회피 등과 같은 복잡한 제어 과정을 필요로 한다. 이러한 부분을 해결하기 위해 MoveIt을 활용한다. MoveIt은 로봇팔을 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 역기구학을 해석하고 경로를 생성하는 과정을 시각적으로 표현할 수 있고 시뮬레이션을 수행할 수 있다. [4] 그림 4는 MoveIt API를 이용하여 각 joint 값을 지정에 motion planning을 진행, /joint states 토픽이 publish되는 그림이다.



<그림 4> MoveIt API를 이용한 motion planning

5. 결론 및 향후 연구 방향

본 논문에서는 자체 설계한 6축 매니퓰레이터의 원하는 Task를 수행하기 위해 목표 성능을 선정하고 이를 기반으로, 각 조인트의 요구 성능에 맞춰 모터를 선정하여 로봇의 모델을 구성하였으며 구성한 로봇 모델의 URDF를 작성하였고 그것을 토대로

MoveIt 패키지를 만들어 거동 시뮬레이션을 진행해 보았다.

향후에는 MoveIt에서 역기구학 해석을 통해 계산한 각 joint의 각도를 MCU로 전송하면 MCU는 모터의 위치 제어를 통해 이동하게 되고 엔드이펙터가 목표 지점에 도착하게 되면 MCU로 그리퍼 제어 명령을 보내 물체를 파지하게 되는 시나리오를 진행할 예정이며 이를 위해 서비스 작업 수행과 물표물체 인식 및 3D 좌표를 추출하는 darknet_ros_3d 패키지[5]를 사용한 뎁스카메라(D435, Intel사)[6]를 엔드이펙터 위치에 부착해 제작한 뒤 서비스 시나리오에 맞는 알고리즘대로 구동시키기 위한 새로운 노드를 작성하여 제작한 매니퓰레이터의 활용을 검증할 수 있을 것이다.

감사의 글

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

참고문헌

[1] PYO Y. S, JO H. C, JUNG R. W. and LIM T. H, 2017, ROS robot programming, Bucheon, Ruby paper press.

[2] David Greenfield, 2015, Inside the Human-Robot Collaboration Trend, https://www.automationworld.com/inside-the-humanrobot-collaboration-trend, vewd on March 24

[3] ROS wiki, http://wiki.ros.org/urdf, viewd on March 24.

[4] MoveIt, https://moveit.ros.org, viewd on 08 April

[5] Fernando González Ramos, darknet_ros_3d, https://github.com/IntelligentRoboticsLabs/gb_visual_detection_3d, viewd on 08 April 2021

[6] Intel, Realsense d435, http://wiki.ros.org/RealSense, viewd 08 April 2021