

ROS를 활용한 모바일 로봇과 매니플레이터의 통합

고동현^a, 이재현^a, 김정인^a, 박범솔^a, 이치범^{a*}

Integration of Mobile Robot and Manipulator Using the ROS

Donghyun Ko^a, Jaehyun Lee^a, Jungin Kim^a, Bumsol Park^a, Chibum Lee^{a*}^a Department of Mechanical System Design, Seoul National University of Science & technology

ARTICLE INFO

Article history:

Received	23	March	2020
Accepted	11	May	2020

Keywords:

Mobile manipulator
Service automation
SLAM and navigation
Inverse kinematics

ABSTRACT

In this study, a mobile robot and a manipulator were integrated and implemented using ROS (Robot Operating System). Using computer vision, the robot was intended for delivering objects such as beverages and cans to customers in service environments such as cafes and restaurants, and it will contribute to improvement in work efficiency and service automation. To reduce production costs and weight, we adopted the 4-axis manipulator with gears and motors inside the link so that the manipulator occupied a minimum space within the operating range. To achieve free movement in a narrow workspace, direct drive mechanism with low-cost geared motors was adopted for the mobile platform.

1. 서 론

현재 로봇산업에서는 실용성과 안정성을 높이기 위해 다양한 로봇을 개발하고 있다. 모바일 로봇과 매니플레이터를 활용하여 공장자동화 및 무인운용시스템을 구축하고 있으며, 최근에는 서비스 환경에서도 활용이 이루어지고 있다. 이 둘의 결합체인 모바일 매니플레이터는 현재 물류창고, 스마트 팩토리 등 지능형 자율 공장 환경에서 공장자동화를 위한 필수적 요소로 활용된다^[1,2]. 모바일 매니플레이터의 예로, Fetch mobile manipulator (Fetch robotics사, USA)는 물류 산업현장에서 사람을 대신하여 직접 화물을 들어서 운반하고, 정리하는 작업을 수행할 수 있다^[3]. 모바일 매니플레이터는 산업용 로봇 분야에서 연구와 개발이 주로 이루어져왔지만, 최근에는 서비스 분야에 활용하려는 노력이 이루어지고 있다. 서비스 로봇 시장이 확대되면서, 과거에는 단순 반복 작업을 대신하는 보조 역할의 로봇에서, 최근에는 스스로 움직이면서 어

렵고 다양한 작업을 수행하는 역할의 로봇으로 변화하고 있다.

본 연구에서는 로봇 응용 프로그램 개발을 위한 오픈소스 운영 체제인 ROS (robot operating system)을 활용하여 모바일로봇과 매니플레이터를 통합하여 구현하였고, 영상처리를 통하여 카페, 레스토랑 등 서비스 환경에서 스스로 음료, 캔 등의 물체를 사용자에게 배송하는 역할을 수행하며, 사용자의 업무 효율화 및 서비스의 완전 자동화에 기여하고자 한다.

기존 산업현장에서 사용되는 모바일 매니플레이터 같은 경우 높은 가반하중(payload)과 제어를 요구하기 때문에 고성능의 모터를 활용한 6축 이상의 다관절로 구성되어 있으며, 그에 비례하여 제작 비용 및 무게가 상승한다. 따라서 작동환경을 고려하여 기존 제품 대비 제작비용 절감과 무게 감소를 위해 허용하는 작동 범위 내에서 최소한의 공간을 차지하도록 링크 내부에 기어와 모터를 넣는 구성을 채택하였다. 또한, 일반적인 테이블 높이에 맞춰 다관절 로봇의 길이를 약 700 mm로 설계하였고 4축의 다관절 로봇으로 구

* Corresponding author. Tel.: +82-2-970-6337

E-mail address: chibum@seoultech.ac.kr (Chibum Lee).

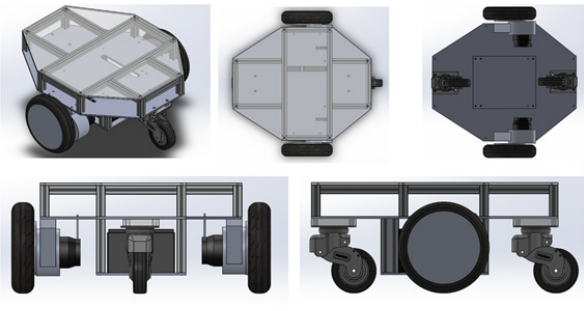


Fig. 1 Designs of mobile robot

성하여 기존의 모바일 매니퓰레이터의 높은 비용과 무게에 대한 단점을 보완하였다.

본 연구에서는 터틀봇^[4]을 참조하고 ROS를 활용하여 모바일로봇 노드, 매니퓰레이터 노드, 이 둘을 결합한 모바일 매니퓰레이터 노드를 만들었다. 본 논문의 구성은 2장에서는 모바일 로봇 개발, 3장에서는 매니퓰레이터 개발, 4장에서는 모바일로봇과 매니퓰레이터의 통합, 5장에서는 서비스 환경의 활용 순으로 이루어져있다.

2. 모바일 로봇

2.1 모바일 로봇 설계 사항

실내 환경에서 사람과의 상호작용이 주된 로봇을 개발하기 위해서는 실내 작업 환경과 공간을 적응하며 구동해야 한다. 이에 따라서 제자리 회전이 가능한 2륜 직접 구동(direct drive) 방식에 높이 조절기능이 있는 캐스터를 부착한 모바일 로봇으로 설계되었다. 또한, 사람들이 움직이는 환경을 고려하여, 로봇의 이동속도는 일반인의 평균 걷는 속도인 3~4 km/h에 근접한 속도로 주행 가능하도록 모터를 선정하고 구동부를 설계하였다. 추가적으로 상단에 약 15 kg의 매니퓰레이터를 장착하여 물체 이동 서비스를 수행하고자 하는 목적이 있기에 목표한 크기의 프레임, 컨트롤러, 모터 및 기타 전자제품의 무게를 고려하여 최대 40 kg으로 제한하였다. 다음의 Fig. 1은 모바일 로봇 3D 설계 모델링이다.

2.2 모바일 로봇 시스템 구성

구동 모터는 비용과 내구성을 감안하여, 감속기가 내장되어 고회출력을 얻을 수 있으며 요구 속도에 부합하는 저렴한 전동 휠체어용 BLDC 모터(PEWM-01, Pei Scooter사, China)를 선정하였다. 구동 모터에는 엔코더가 부착되어있지 않고 BLDC 제어를 위한 홀센서만 존재하므로, 이 홀센서를 이용하여 위치를 측정하였다. 8개의 극(pole)을 가지고 있고 3비트의 그레이 코드의 6개의 시퀀스를 가지고 있기 때문에 15도의 분해능을 가진다. 감속비가 17:1이므로, 바퀴의 각도는 0.8823도의 분해능으로 측정할 수 있다.

이러한 홀 센서의 신호를 저주파 필터를 거친 후 최대 분해능으로 사용하기 위해 그레이 코드를 바이너리 코드로 변환하여 LSB (least significant bit)의 펄스를 받는다. 방향도 최대 분해능으로 판단할 수 있도록 총 6개의 상승 엣지, 하강 엣지의 펄스를 아날로그하게 더하여 계단 형식의 신호가 발생하도록 하였다. 이를 통해 MCU (micro control unit)에서는 한 개의 디지털 포트로 방향을 측정하고, 한 개의 아날로그 포트로 회전 각도를 측정하여, 총 두 개의 입력단자로 휠의 회전 각도를 측정할 수 있다.

측정한 각도를 통해 모바일 로봇의 이동 거리를 계산한다.

$$\Delta d = \frac{R \times (\Delta \theta_l + \Delta \theta_r)}{2} \quad (1)$$

(Δd : 이동한 거리, R: 바퀴 반지름, $\Delta \theta$: 바퀴 회전 각도)

왼쪽 바퀴와 오른쪽 바퀴의 거리의 차이를 통해 모바일 로봇이 회전한 요우(Yaw) 각도를 계산한다.

$$\Delta \Psi = \frac{R \times (\Delta \theta_l - \Delta \theta_r)}{s} \quad (2)$$

(Ψ : 요우 각도, s: 윤거(Tread))

이를 통해 절대 좌표로부터 현재 위치까지의 거리를 누적하여 계산한다.

$$x_{i+1} = x_i + \Delta d \times \cos(\Psi_i + \frac{\Delta \Psi}{2}) \quad (3)$$

$$y_{i+1} = y_i + \Delta d \times \sin(\Psi_i + \frac{\Delta \Psi}{2}) \quad (4)$$

$$\Psi_{i+1} = \Psi_i + \Delta \Psi \quad (5)$$

(x : 절대 좌표의 x축 이동거리, y : 절대 좌표의 y축 이동거리, Ψ : 절대 좌표의 z축을 기준의 각도)

MCU에서는 계산된 위치(odometry) 정보를 30 Hz의 주기로 내보낸다. 또한, 현재 속도와 각속도는 한 스텝 동안 이동한 거리와 이동한 각도를 스텝 타임으로 나누어 계산하여 마찬가지로 30 Hz 주기로 내보낸다.

모바일 로봇의 제어는 모터의 회전을 측정하고 제어하는 MCU와 환경과 상호 작용하면서 전체적인 로봇의 거동을 제어하는 PC에서 나누어 이루어진다. MCU에서는 모터를 직접 제어하고, 모터의 현재 속도, 각도로부터, 로봇의 위치(odometry)를 계산하고 지속적인 통신을 통해 PC에 보내게 된다. PC에서는 라이다(lidar) 데이터를 이용하여, 주변 지형을 인식하고 내장된 지도와 비교하여 위치를 추정하고, 확률적인 수렴을 통하여 지도에 존재하는 로봇의 위치를 상정한다. 사용자에 의해 목표 지점이 설정되

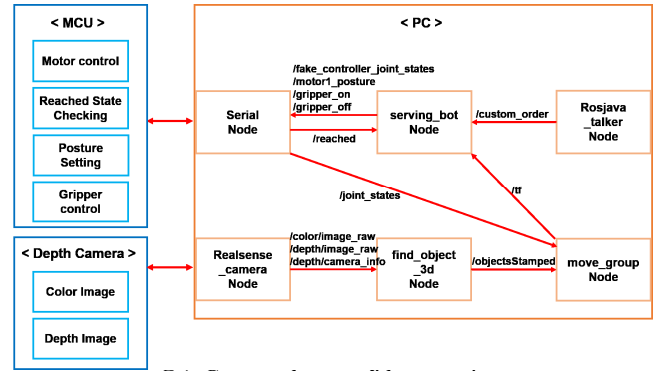
면 글로벌 플래너가 경로(path)를 생성하고, 이동하면서 로컬 플래너가 장애물을 회피하면서, 설정된 파라미터에 따라 구동 요구 속도를 산출하여 MCU에 전송하게 된다. 이에 따라 MCU는 모터의 회전을 제어한다. Fig. 2는 모바일로봇 노드와 데이터 이동에 관한 그림이다.

2.3 SLAM & Navigation

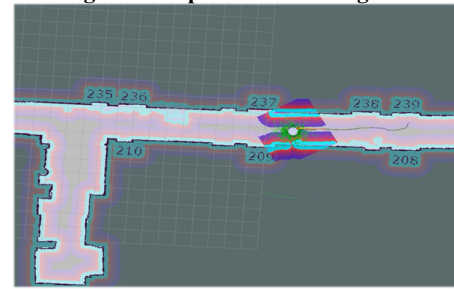
SLAM 및 navigation 알고리즘은 로보티즈사의 터틀봇을 참고하여 구성하였다^[4,5]. 작업 환경에서 지도를 획득하기 위한 SLAM에는 일반적으로 Gmapping^[5]과 Cartographer^[6]를 많이 이용한다. 본 연구에서 모바일로봇을 구현하기 위해 저렴한 저전력 임베디드 PC를 사용하므로, 실시간으로 복잡한 알고리즘을 구동하기에는 성능이 충분하지 않다. 따라서 저용량이고, 라이다 데이터로부터 격자 지도를 그리는 Rao-Blackwellized particle filter를 이용하는 Gmapping을 이용하였다^[7]. 위치 추정(Localization) 알고리즘은 2D 지도를 기반으로 이동하는 로봇의 위치를 추종하는 particle filter를 사용하는 확률적 위치 인식 시스템인 AMCL (adaptive monte carlo localization) 알고리즘을 이용하였다. Fig. 3(a)은 모바일로봇을 활용하여 실제 도서관내 카페를 맵핑한 결과이다.

AMCL 알고리즘으로 위치 추정을 하는데 있어서 이동하면서 위치 오차가 증가하는 경우가 발생하였는데, 이 경우 위치를 재추정할 필요가 있다. 위치 오차가 크게 발생하는 경우, 일정 양의 입자를 다시 생성하고 수렴시켜서 현재 포즈의 추정 값을 출력값을 갱신하여 위치 정확도를 향상시켰다.

Navigation에서는 SLAM을 통해 작성된 지도를 기반으로 글로벌 플래너가 글로벌 비용 지도에서 종합적인 현재 위치에서부터 목표 지점까지의 경로를 생성한다. 또한, 이동하면서 로컬 플래너가 현재 로봇의 위치를 기반으로 로봇의 라이다 센서가 인식하는 점들을 기반으로 로컬 비용 지도에서 주변 영역 만큼에 대한 짧은 시간성을 지닌 경로(trajecory)를 생성한다. 본 연구에서는 장애물 회피를 위하여 DWA (dynamic window approach) 로컬 플래너를 사용하였는데 이는 장애물을 고려한 시뮬레이션을 통하여 V_x , V_y , w 값을 결정하고, 경로를 수정하게 된다. 모바일 로봇의 설계를 활용하여 URDF (Unified robot description format)와 설계 파라미터를 지정하였고, 더불어 태스크 수행에 적합한 파라미터들을 설정하였다. Navigation의 경로 생성에 가장 영향을 미치는 것은 로컬비용 지도의 파라미터이다. Fig. 3(b)는 실제 모바일로봇의 navigation의 그림이다. 실제 실험을 통해 로컬 비용 지도의 파라미터가 장애물을 회피하고, 로봇의 현재 경로 생성에 영향을 미치는 것을 확인했다. 그 중에서 경로를 생성할 때 장애물 근처에 가지 못하도록 우선순위를 설정하는 cost scaling factor,



(a) Generated map with gmapping
Fig. 5 Manipulator node diagram



(b) Mobile robot navigation

Fig. 3 SLAM and navigation

회전 반경을 설정하는 inflation ratio, 현재 로봇의 2D 크기를 설정하여 자동적으로 장애물과의 최단 거리를 설정하는 footprint 등의 영향이 큰 것으로 나타났다. 따라서 실험적으로 위의 파라미터의 최적값을 찾아내어 서비스 수행에 적절하게 활용하였다.

3. 매니플레이터

3.1 매니플레이터 설계 사항

실내 환경에서 700 mm 높이의 테이블 위의 물체를 집어서 옮기는 작업을 수행하고, 유연한 동작이 가능하도록 4축으로 매니플레이터를 설계하였다. 또한, 물체 이동 서비스를 안정적으로 수행하기 위하여, 가반 하중 1.5 kg으로 설계하였다. 설계 목표 중량 15 kg이었으나, 실제로는 13 kg으로 제작하였고, Fig. 4는 매니플레이터 3D 설계 모델링이다.

3.2 매니플레이터 시스템 구성

매니플레이터는 상대적으로 출력이 높은 5상 스테퍼 모터(AK series, Autronics사, Korea) 4개를 이용하여 4축의 다관절 로봇을 구성하였다. MCU단에서 스테퍼 모터를 제어하고, 현재 목표 지점에 도달한 상태인지 목표 지점으로 이동하고 있는 상태인지를 판단하여 ROS 토픽으로 PC로 전송한다. 또한 초기 위치 자세 지정을 통해 누적된 오차를 초기화 시킨다. 또한 목표 지점에 도달했을

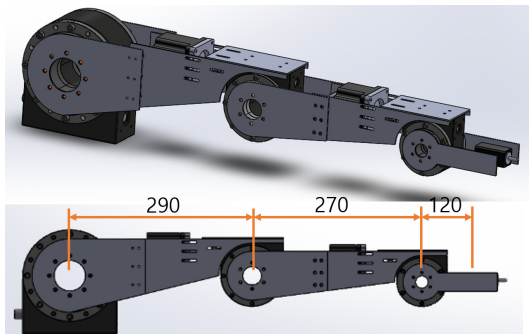


Fig. 4 Designs of manipulator

때 그리퍼를 제어하여 목표 물체를 집는다. 현재 조인트 등의 위치와 도착 여부를 시리얼 노드를 이용하여 PC로 전송한다.

STEP 모터의 모션 제어에는 AccelStepper 라이브러리를 이용하였다. 다관절 로봇의 부하를 줄이고 부드러운 움직임을 위해 사다리꼴 프로파일을 적용하였다. 각각의 STEP 모터에 사다리꼴 프로파일의 최대 속도를 적용하였다. PC로 부터 받은 목표 각도를 각 모터의 분해능과 기어(robolink joint series, Igus사, Germany)비를 고려하여 목표 스텝을 정했다. 진동을 줄이기 위해 각각의 모터의 감·가속 시간과 일정한 속도를 유지하는 시간이 동일하게 구성하기 위해 각도의 변화에 비례하게 속도를 결정하는 시스템으로 만들었다. Fig. 5는 매니플레이터 노드 간 데이터 통신에 관한 그림이다.

3.3 MoveIt을 활용한 역기구학 해석

4축 매니플레이터이므로, 해석적으로 역기구학을 해석하는 것이 어렵지 않으나, 가상 공간에서 시뮬레이션을 하고, 다양한 시나리오를 검증할 수 있으므로 오픈소스인 MoveIt^[9]을 활용하였다. 이를 통해 매니플레이터의 역기구학을 해석하고 경로를 생성하는 과정을 시각적으로 표현할 수 있고 시뮬레이션을 수행할 수 있다. 엔드 이펙터의 경로 생성은 OMPL (open motion planning library)중에서 가볍고 보편적인 RRT (rapidly-exploring random trees)^[10]를 사용하였다. RRT는 여러 번의 경로 계획을 한 후 그 중에서 가장 짧은 경로를 선택하는 방법을 사용했다. 속도 프로파일은 사다리꼴 프로파일을 적용했다.

역기구학을 풀기위하여 매니플레이터의 구성에 관한 정보인 회전 축 정의, 링크의 길이, 조인트의 종류와 제한 각도 등이 필요하다. MoveIt에서는 일반적으로 URDF의 파일을 이용하여 매니플레이터 모델을 구성한다. URDF 파일은 기구학 해석에 필요한 매니플레이터의 베이스 프레임과 차일드 프레임간의 좌표 변환(tf)과 각도 차, 동역학해석에 필요한 관성데이터, 시뮬레이션 시 가시화를 위한 그래픽 데이터가 포함된다. 본 매니플레이터는 4개의 축을 가지고 있기 때문에 URDF 모델은 베이스 프레임을 제외한 4개의

링크가 이웃한 링크 사이의 프레임 변환 정보를 가지고 있다. Fig. 6(a)는 URDF로 정의한 매니플레이터의 모델링이고, (b)는 실제 MoveIt을 활용하여 매니플레이터의 거동 시뮬레이션이며, (c)는 매니플레이터의 URDF 블록다이어그램이다.

PC에서 역기구학을 통해 계산한 각 모터의 각도를 MCU로 전송하면 MCU는 위치 제어를 통하여 이동하게 된다. 엔드 이펙터가 목표 지점에 도착하게 되면 PC에서 MCU로 그리퍼를 제어 명령을 보내어 물체를 파지된다. 매니플레이터가 경로를 부드럽게 따라가도록, 실험적으로 경로 포인트(Way Points) 전송 주기의 최적값을 찾아 정확성을 향상하였다.

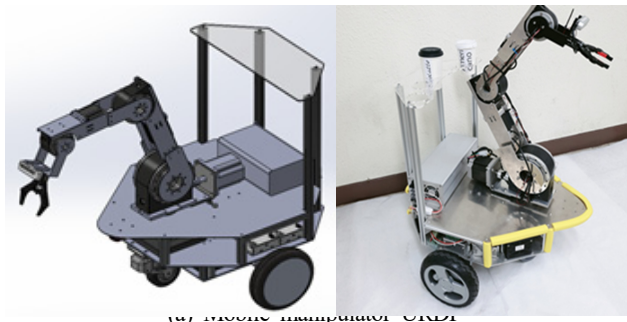
4. 모바일 로봇과 매니플레이터의 통합

4.1 모바일 매니플레이터 시스템 구성

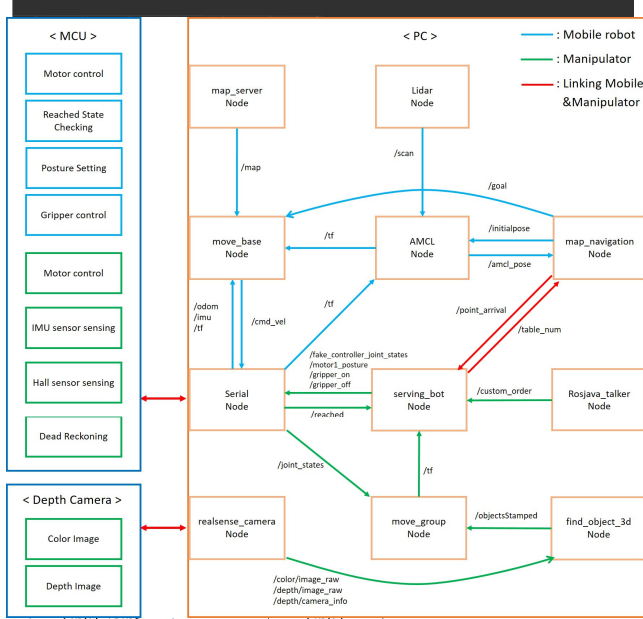
모바일 로봇과 매니플레이터는 각각 MCU에서 모터의 회전을 제어하고, 전체적인 거동은 PC에서 처리된다. 모바일로봇과 매니플레이터의 통합은 하나의 PC와 하나의 MCU에서 처리하도록 구성하였다. 시리얼 노드를 통해 PC와 MCU가 Odometry, tf, joint state등의 정보를 30 Hz 주기로 송수신한다. 이를 통해 MCU는 하단부 모바일 로봇 단의 휠제어용 BLDC모터 2개와 STEP 모터 4개를 제어한다. MCU는 모터 6개와 센서를 연결하기 포트 수가 충분하고, ROS와의 통신하기 위한 접근성이 뛰어나며, 제어 성능이 충분한 로보티즈사의 OpenCR^[11]을 사용했다. 임베디드 PC는 저전력이지만 SLAM과 navigation, 역기구학 등을 수행하여야하므로, 영상처리 성능이 우수한 Nvidia사의 Jetson TX2^[12]를 선정하였다. 라이다는 USB통신을 통해 PC에 연결되어있으며, SLAM과 navigation을 위한 스캔 데이터를 송수신하여 주변 지형 정보를 받아온다. 이 밖에 서비스 작업 수행과 목표물체 인식 및 특징점을 찾아내기 위한 딥스 카메라(D425, Intel사, US)^[13]를 엔드이펙터 위치에 부착하였고, USB 통신으로 PC와 연결된다. 모바일 로봇에 해당하는 부분과 매니플레이터에 해당하는 부분을 별도의 독립적인 구조로 제작한 뒤 서비스 시나리오에 맞는 알고리즘대로 구동을 시키기 위해 새로운 노드(serving_bot)를 작성하여 매니플레이터와 모바일로봇을 연동시키는 방식으로 구성하였다. Fig. 7(a)는 모바일 매니플레이터의 전체적인 데이터 통신에 관한 그림이고, Fig. 7(b)는 하드웨어 구조에 관한 그림이다.

4.2 모바일 로봇과 매니플레이터의 통합

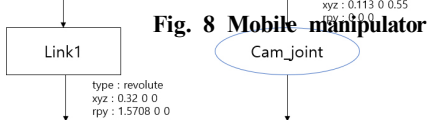
본 연구는 서비스 환경에서 사용하는 로봇의 개발이므로, 실 사용 시간을 고려한 배터리의 장착과 로봇의 경량화가 중요한 요소이다. 따라서 모바일로봇은 6시간 연속 동작을 목표로 배터



(a) Design of mobile manipulator



(b) Mobile manipulator node diagram

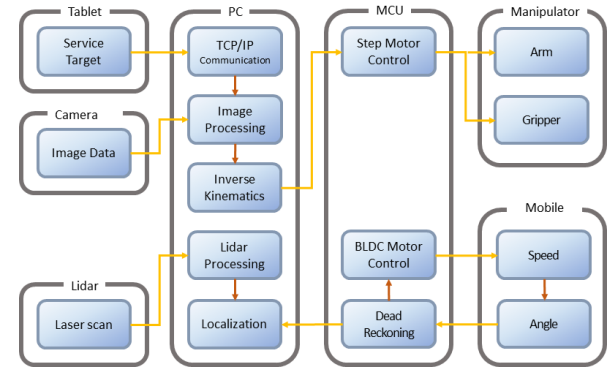


(c) URDF block diagram

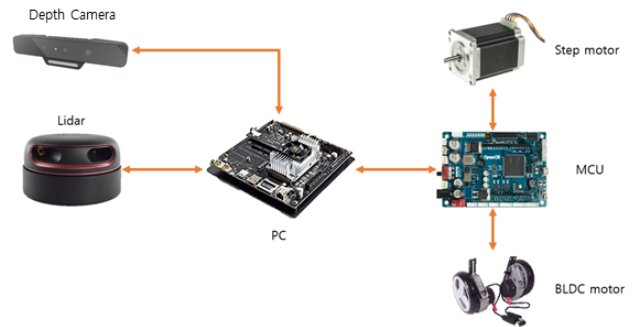
Fig. 6 Manipulator structure

리(12 V 2개+ 24 V 1개)를 포함하여 약 35 kg인 서비스용 로봇을 개발하였다.

모바일 로봇의 navigation과 매니플레이션을 결합하기 위해서 모바일 로봇의 tf와 매니플레이터의 tf를 결합하였다. 매니플레이터와 모바일 로봇의 설계 파일을 통해 앞서 매니플레이터에서 링크간의 tf 관계 정보를 담은 URDF 파일을 작성했다. 매니플레이터의 엔드 이펙터를 원하는 대로 제어함에 있어서, 모바일 로봇단이 이동하기 때문에 베이스 링크를 매니플레이터가 아닌 모바일 로봇으로 설정했다. 모바일 로봇에 설정한 베이스 링크는 SLAM을 통해 작성했던 지도의 절대 좌표를 기준으로 위치가 계속 변화한다. 따라서 그에 따라 모바일로봇의 링크와 매니플레이터의 링크의 위치도 지속적으로 변화한다. move_base 노드에서 계산되는



(a) Software structure



(b) Hardware structure

Fig. 7 System structure of mobile manipulator

위치를 기반으로 베이스 링크의 위치와 각도가 정해지게 되면 엔드 이펙터는 모바일로봇단의 베이스 링크를 기준으로 역기구학을 해석하여 매니플레이션을 수행한다.

Fig. 8(a)는 모바일 매니플레이터의 모델링과 실제 제작사진이고, Fig. 8(b)는 모바일 매니플레이터의 전체 노드와 데이터 이동에 관한 전체 구조이다.

5. 모바일 매니플레이터를 활용한 서비스 구현

개발된 모바일 매니플레이터의 활용성을 검토하기 위하여 실내 카페나 음식점에서 음료를 배송하는 서비스 작업을 구현해 보았다. 우선 앱을 활용하여 테이블 위치와 음료 주문에 대한 정보를 로봇에게 전달한다. 로봇은 수신된 정보를 기반으로 뎁스 카메라를 사용해 음료수 컵을 판별하여 매니플레이터로 집어서 트레이에 넣어서 운반하고 각 해당하는 테이블에 배송하게 된다.

영상 처리를 통하여 목표 물체를 인식하고 3D 좌표를 생성하기 위해 ROS의 Find_Object^[14]를 바탕으로 소프트웨어를 제작하였다. 뎁스 카메라(Real Sense D415, Intel사, US)를 이용하여 2D 이미지 정보와 거리 정보가 획득한다. 먼저, 2D 이미지로 부터 SURF (speeded-up robust features) 알고리즘을 이용하여 특징점들을 추출하고 미리 저장해준 물체의 특징점과 비교하여 물체의

일치 여부를 판단하고, 현재 일치된 물체의 중심점 좌표를 추출하여 해당 좌표의 거리 정보를 가져와 3D 좌표를 생성한다^[15]. 전체적인 3D 영상처리 구성도는 Fig. 9(a)와 같다. 3D 공간상에 물체의 위치가 특정되면 PC에서 현재 목표 지점의 위치로의 역기구학을 계산하여 목표지점에 도달하게 된다. Fig. 9(b)는 Rviz를 활용하여 영상인식을 통해 물체 인식 및 거리계산과 이를 이용한 역기구학 해석을 나타낸 그림이다.

목표위치에 도달한 매니퓰레이터는 그리퍼를 활용하여 물체를 집게 된다. 그 후 현재 로봇의 남은 적재 공간을 판단하여 물체가 없는 곳에 수납하고 목표 물체가 모두 수납하면 PC로 다음 구동 정보를 전달한다.

모바일로봇은 사전에 제작된 지도를 바탕으로 주문된 테이블의 위치까지의 최적경로를 파악한 후, 형성된 경로를 통해 navigation을 수행한다. 이동 중에 라이다를 통해 장애물을 인식하면 우회하여 장애물을 회피하고, 도착 지점에 도달하면 다시 매니퓰레이션을 통해 음료를 트레이로부터 집어서 테이블에 옮겨두고 음료 배송 서비스를 마무리한다^[16].

6. 결 론

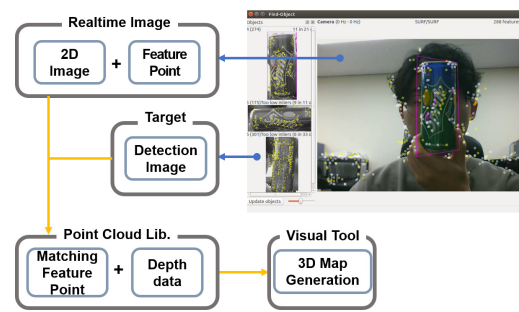
본 연구에서는 서비스 환경에서 작업이 가능한 로봇 개발을 주제로 모바일 로봇과 매니퓰레이터의 결합을 통해 모바일 매니퓰레이터를 개발하였다. 서비스 환경이 이루어지는 실내 제한 조건에 맞춰 무게중심의 변화에 따른 안정적인 주행과 제자리 회전이 가능하도록 하단의 모바일 로봇을 설계하였다. 그리고 상단의 매니퓰레이터는 스테퍼모터를 활용하고 알루미늄을 주재료로 외형을 설계하여 안정적인 매니퓰레이터 구동을 할 수 있게 설계하였다.

전체적인 모바일 매니퓰레이터를 구동하기 위해서는 navigation, 매핑, 비전, 매니퓰레이션 등 로봇의 서비스를 위한 알고리즘이 필요하므로, ROS를 기반으로 프로그램을 개발하였다. ROS 하위에 라이다 센서, 딥스 카메라를 제어하기 위한 프로그램을 작성하였고, 실시간 모터제어를 위하여 OpenCR을 사용되는 펌웨어를 작성하였다^[17].

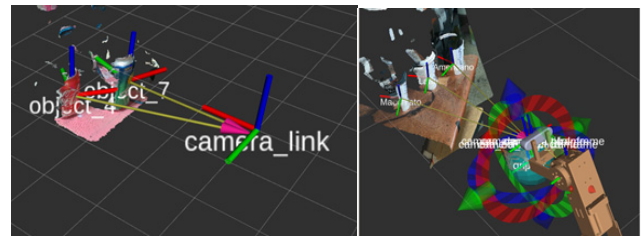
이를 통해 개발된 로봇은 실제 카페에서 실험한 결과, 태블릿을 통한 음료 주문부터 음료 배송까지 음료 서비스 작업 수행의 자동화를 구현하여 모바일 매니퓰레이터의 성능 및 활용성을 검증하였다.

후 기

이 연구는 서울과학기술대학교 교내 학술연구비 지원으로 수행되었습니다.



(a) Image processing diagram



(b) Detection image in Rviz

Fig. 9 Image processing and inverse kinematics

References

- [1] Leidner, D., Dietrich, A., Schmidt, F., Borst, C., AlbuSchaffer, A., 2014, Object-Centered Hybrid Reasoning for WholeBody Mobile Manipulation, International Conference on Robotics and Automation, 1828-1835.
- [2] Hamner, B., Koterba, S., Shi, J., Simmons, R., Singh, S., 2010, An Autonomous Mobile Manipulator for Assembly Tasks, Auton Robot, 28:131, <https://doi.org/10.1007/s10514-009-9142-y>.
- [3] Fetch Robotics, n.d., viewed 30 August 2019, Fetch Mobile Manipulator, <<https://fetchrobotics.com/robotics-platforms/fetch-mobile-manipulator/>>.
- [4] Robotis, n.d., viewed 20 March 2019, Navigation tuning, <<http://emanual.robotis.com/docs/en/platform/turtlebot3/navigation/#tuning-guide/>>.
- [5] ROS wiki, n.d., viewed 20 March 2019, OpenSlam's Gmapping in ROS, <<http://wiki.ros.org/gmapping/>>.
- [6] Hess, W., Kohler, D., Rapp, H., Andor, D. 2016, Real-Time Loop Closure in 2D LIDAR SLAM, in Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 1271-1278.
- [7] Grisetti, G., Stachniss, C., Burgard, W., 2005, Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling, Proceedings of

the 2005 IEEE International Conference on Robotics and Automation, 2432-2437.

- [8] ROS wiki, n.d., viewed 10 February 2019, ROS in AMCL, <<http://wiki.ros.org/amcl>>.
- [9] MoveIt, n.d., viewed 30 June 2019, ROS with MoveIt, <<https://moveit.ros.org/>>.
- [10] LaValle, S. M., 1998, Rapidly-exploring random trees: A new tool for path planning, Technical Report. Computer Science Department, Iowa State University (TR 98-11).
- [11] OpenCR, n.d., viewed 20 March 2019, OpenCR, <<http://emanual.robotis.com/docs/en/parts/controller/opencr10/>>.
- [12] NVIDIA, n.d., viewed 30 June 2019, Realsense in Jetson TX2, <<https://www.jetsonhacks.com/2017/08/14/intel-real-sense-camera-librealsense-nvidia-jetson-tx-dev-kits/>>.
- [13] Intel, n.d., viewed 30 June 2019, Intel Realsense d415, <<http://wiki.ros.org/RealSense>>.
- [14] ROS wiki, n.d., viewed 10 November 2019, Find_object, <http://wiki.ros.org/find_object_2d>.
- [15] Lee, K. M., Na, S. M. R., Park, J. H., Ann, K. W., Lee, C., 2018, Development of a Dual-arm Service Robot Using Stereo Vision and 6-Axis Manipulators, Journal of the Korean Society of Manufacturing Technology Engineers, 27:3 271-277, <https://doi.org/10.7735/ksmte.2018.27.3.271>.
- [16] Youtube, n.d., viewed 14 November 2019, Performance “Mobile Manipulator”, <<https://www.youtube.com/watch?v=7-oHuMfVjQM&t=1s/>>.
- [17] Github, n.d., viewed 2 December 2019,, Mobile Manipulator ROS Code, <https://github.com/BeomSol/servingbot_ros/>.



Donghyun Ko

Undergraduate student in the Department of Mechanical System Design Engineering, Seoul National University of Science & Technology. His research interest is robotics.
E-mail: gdh8652@gmail.com



Jaehyun Lee

Undergraduate student in the Department of Mechanical System Design Engineering, Seoul National University of Science & Technology. His/Her research interest is robotics.
E-mail: jaehyun0752@gmail.com



Jungin Kim

Undergraduate student in the Department of Mechanical System Design Engineering, Seoul National University of Science & Technology. His research interest is the design of mechanical system.
E-mail: kimji0708@naver.com



Bumsol Park

Undergraduate student in the Department of Mechanical System Design Engineering, Seoul National University of Science & Technology. His/Her research interest is programming and robotics.
E-mail: pbs1995@naver.com



Chibum Lee

Associate Professor in the Department of Mechanical System Design Engineering, Seoul National University of Science & Technology. His/Her research interest is control systems and robotics.
E-mail: chibun@seoultech.ac.kr