**캡스톤 디자인 중간 발표 스크립트**

01.

안녕하십니까 Field painting robot based on ROS and map recognition이라는 주제로 소프트웨어융합캡스톤디자인 프로젝트를 진행 중인 김정윤, 방지호입니다. 지금부터 발표 시작하겠습니다.

02.

발표는 다음과 같은 순서로 진행하겠습니다.

03.

Field painting이란 축구장과 야구장과 같은 경기장의 라인을 그리거나 도로 위의 차선을 그리는 것을 의미합니다. 최근 NISSAN에서 축구 경기장의 라인을 그려주는 피치-R 로봇을 개발하여 매번 번거롭게 라인을 그리지 않아도 되는 가능성을 보여주었습니다. 그렇지만 현재까지도 도로 위의 차선은 사람이 직접 도색 작업기를 이용해서 그리고 있습니다. 도색 작업기는 고온의 열을 내뿜기 때문에 사람이 작업하기에는 열악한 근무 환경을 제공합니다. 뿐만 아니라 도색 작업은 차가 다니는 도로 위에서 진행되므로 많은 위협에 노출되어 있습니다. 최근까지도 도색 작업을 진행하다가 도색기가 폭발하거나 주변 차량에 치이는 사고로 인해 근무자가 크게 다치거나 목숨을 잃는 사고가 발생하였습니다. 따라서 저희는 ROS를 활용하여 map을 인식하고 자동으로 라인을 그려주는 로봇을 개발하고자 합니다.

04.

Field painting을 구현하기 위해선 가장 먼저 Navigation에 대해서 알아야 합니다. 주변 장애물을 모두 회피하며 정해진 경로를 따라 라인을 그리기 위해선 Navigation 기능이 기초가 되기 때문입니다. 따라서 Navigation의 이론에 대해서 공부하였습니다. Navigation은 센싱을 통해 정보를 받아들이고 이를 바탕으로 자신의 위치를 추정하며, 도착지까지 어떻게 갈지 모션을 계획하고, 이동하는 일련의 과정을 의미합니다. Navigation 구현을 위해선 다음과 같은 총 6개의 정보를 필요로 합니다.

첫번쨰로 오도메트리입니다. 오도메트리는 로봇의 현재 속도 등의 정보를 받아서 국부 이동 경로를 생성하거나 장애물 회피 등에 사용됩니다.

두번째로 상대 위치 변환입니다. ROS에서는 tf라는 상대 위치 변환을 이용합니다. 오도메트리를통해 단순히 로봇의 위치를 알게 되는데 상대 위치 변환은 그 로봇의 위치로부터 센서가 얼만큼 떨어져 있는지 계산하고 이를 바탕으로 이동 경로 계획을 수행하게 됩니다.

세번째로 거리 센서는 센서로부터 측정된 거리의 값을 의미합니다.

네번째로 지도인데 ROS의 Navigation에서는 점유 격자 지도를 사용합니다.

다섯 번째로 목표 좌표는 사용자가 로봇을 어디로 이동시킬 것인지 위치 정보를 정하는 것입니다.

마지막으로 계획된 이동 궤적에 따라 로봇이 움직이는 속도를 결정하는 속도 명령이 있습니다.

위의 6개의 정보를 바탕으로 ROS에서는 Navigation을 수행하는 것입니다.

05.

Navigation에선 이동 가능한 영역을 계산하기 위해 Costmap을 사용합니다. Costmap이란 엔코더, IMU 센서로부터 얻은 오도메트리 정보를 기반으로 계산한 1) 로봇의 위치, 2) 센서 위치, 3) 장애물 위치 정보와 SLAM의 결과물로 얻은 4) 고정 지도를 불러와서, 이 4가지 요소를 바탕으로 장애물 영역, 장애물과 충돌이 예상되는 영역, 로봇이 이동 가능한 영역을 계산하는 것을 말합니다.

06.

몬테카를로(MCL, Monte Carlo Localization) 위치추정 알고리즘은 위치 추정 분야에서 널리 사용되는 알고리즘으로 AMCL(Adaptive Monte Carlo Localization)은 몬테카를로 위치추정 알고리즘에서 적은 수의 샘플 수를 사용하여 수행 시간을 줄여서 실시간성을 높인 몬테카를로 위치 추정의 기능 개선판이라고 볼 수 있습니다. 앞서 언급하였듯 Navigation에서 로봇이 어디에 위치해 있는지 위치를 파악하는 것은 굉장히 중요합니다. 그러나 주변 환경과 로봇의 센서 등 모든 것들은 불확실성이 존재하므로 로봇이 위치를 파악하기 위해선 확률론적 방법을 사용합니다. Xt는 로봇 위치를 의미하며, ut는 로봇에 입력한 명령을 의미합니다. Zt는 센서로부터 얻은 거리 정보를 의미합니다. 여기서 총 4가지 과정을 거쳐 위치를 추정하게 되는데 바로 예측과 보정, 샘플링, 리샘플링입니다. 각각 다음과 같은 과정을 거쳐서 위치를 추정하게 되는데 시간 관계상 결과만 보도록 하겠습니다. t1, t2, t3, t4로 시간이 감에 따라 로봇의 파티클이 이동하고 장애물을 회피하며 이동경로를 생성하는 것을 보실 수 있습니다.

07.

다음으로 이동 경로 계획 및 장애물 회피에 많이 사용되는 Dynamic Window Approach (DWA)에 대해 알아보겠습니다. DWA란 로봇이 장애물을 회피하며 목표점까지 가장 빠르게 도달할 수 있는 속도를 선택하는 방법입니다. 그림에서 볼 수 있듯이 각각의 선속도와 각속도에 따른 모든 경로를 생성하고 목표점에 가장 근사한 경로를 선정하게 됩니다.

08.

지금까지 이론에 대해 살펴보았다면 지금부턴 ROS를 사용해서 직접 실습해보며 구현한 내용을 말씀드리겠습니다. ROS에서는 오픈 소스 3D 로봇 공학 시뮬레이터인 Gazebo와 로봇과 로봇이 작업하는 환경 및 센서 데이터를 시각화 하는데 사용되는 3D 시각화 도구 Rviz를 많이 사용합니다.

09.

SLAM은 Simultaneous Localization and Mapping의 줄임 말로 지도를 그림과 동시에 로봇의 위치를 추정하는 것을 의미합니다. 쉽게 말해 LOL과 스타크래프트와 같은 게임에서 캐릭터가 이동하면서 보이지 않는 mini map을 확장해 나가는 과정과 비슷합니다. 다음은 ROS에서 기본적으로 제공되는 turtlebot Map에서 Mapping을 진행하고 완성된 Map을 저장한 모습입니다.

10.

동영상에서 볼 수 있듯이 로봇이 이동함에 따라 map을 만들어가는 것을 보실 수 있습니다.

11.

앞서 Mapping을 통해 map을 정보를 저장했습니다. 이를 활용하여 Rviz에서 Navigation 기능도 구현할 수 있습니다. 오른쪽 그림에서 볼 수 있듯이 미리 만들어 둔 map과 로봇을 일치시킨 것을 볼 수 있으며 SLAM을 활용해서 파란색 네모 박스 영역인 Cost map을 실시간으로 계산하고 있는 것을 볼 수 있습니다. 2D Pose Estimate를 통해 로봇의 초기 위치를 조정하여 Map과 로봇의 위치를 일치시킬 수 있으며 2D Nav Goal을 사용해서 로봇을 원하는 위치로 이동시킬 수 있습니다.

12.

동영상에서 볼 수 있듯이 Rviz에서 로봇의 도착점을 지정해주면 로봇이 알아서 도착점으로 이동해가는 것을 볼 수 있습니다.

13.

다음은 최종 목표인 Field painting robot을 구현하기 위해 지금까지 진행상황을 말씀드리겠습니다. 먼저 Field 라인을 원하는 궤적으로 그리기 위해선 지정한 포인트만을 따라가는 Waypoint Navigation 기능을 구현해야 합니다. 따라서 지정한 포인트만을 따라가는 Waypoint 기능을 구현하였습니다. 다음으로 로봇이 이동하면서 라인을 그리기 위해선 원하는 시점에 신호를 주어 가루를 분사하거나 스프레이를 분사하는 등의 역할을 수행할 수 있도록 해야 합니다. 따라서 이동 경로에 따라 신호를 생성하는 기능을 구현하였습니다.

14.

지금까지는 ROS의 Gazebo와 Rviz 상의 컴퓨터 시뮬레이션을 이용해서 Field paint robot을 구현하였습니다. 앞으로 남은 기간동안 ROS의 turtlebot3 burger model과 아두이노의 서보 모터를 커스터마이즈하여 현실에서 map을 인식하고 장애물을 회피하여 원하는 모양의 라인 혹은 도로를 그려주는 로봇을 구현할 계획입니다.

15.

예를 들어 다음과 같이 축구 혹은 야구 경기장의 라인을 그려주거나 오징어 게임 경기장을 그려주는 것입니다. 더 나아가서는 도로의 차선도 그릴 수 있습니다. 이러한 기술은 차선을 그리는 위험한 작업을 로봇이 대체함으로써 사고를 예방할 수 있는 기틀이 될 것입니다.

16.

이상으로 발표 마치겠습니다 감사합니다.