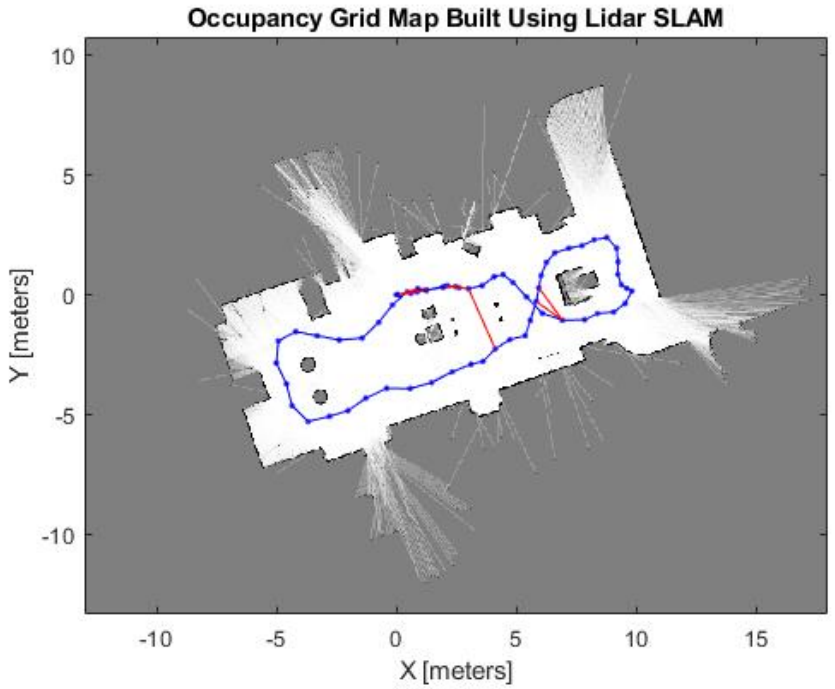
Gazebo SLAM

목차

WeGo

SLAM 기초이론

- SLAM은 Simultaneous Localization And Mapping의 줄임말
- 한글로 동시적 위치 추정 및 지도 작성
- 간단하게 새롭게 무인도를 탐험하면서 지도를 생성하는 것 == SLAM



WeGo

- SLAM의 문제
- 지도를 실제와 유사하게 만든다고 한다면 연속된 공간에서 무한히 많은 차원으로 존재할 수 있으므로 지도 생성이 굉장히 어려워진다.
- Grid Map과 같은 형태를 사용할 때도 수많은 변수로 설명이 가능하며, 이를 통해 위치에 대한 확률을 계산하는 것이 매우 어렵다.
- 지도가 존재할 때 Localization만을 수행하는 것은 어렵지 않으며, Localizaton이 성공할 수 있을 때 지도를 생성하는 것도 어렵지 않지만, 이 둘을 동시에 해결하기에는 어렵다. -> Chicken & Egg Problem

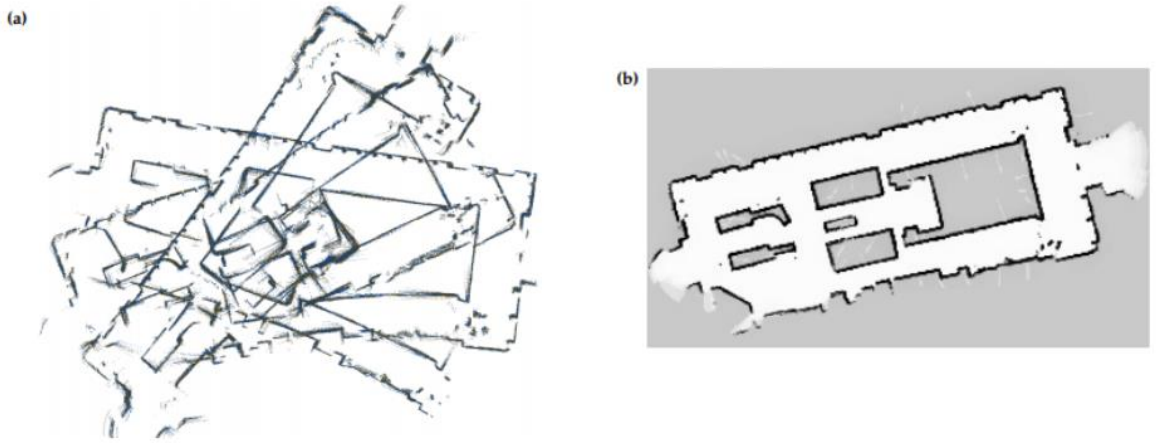
WeGo

- SLAM의 문제의 난이도를 결정하는 요소들
- 생성하려는 지도의 크기
- 당연하겠지만 주변 환경 센싱 및 자기 위치 추정을 위한 센서의 노이즈
- 주변 환경의 반복성 및 인식 모호성 -> 복도, 숲 등에서 쉽지 않음

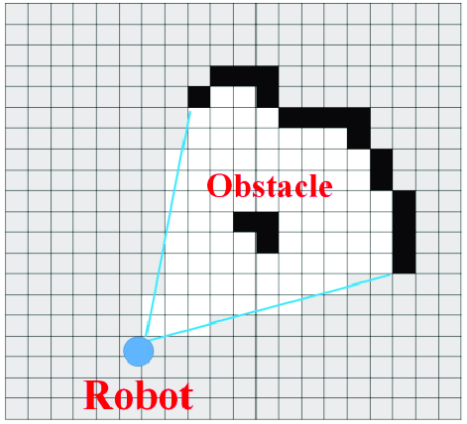
- 실내 이동 데이터를 앞서 배운 Odometry 정보만을 사용하여 SLAM을 한다면 그림과 같은 형상이 나오게 됩니다.(a) 이유 아무리 odometry값을 정확하게 받을 수 있다고 해도 오차가 있기 때문입니다.

(사람이 눈을 감고 다리로 얼마나 움직이는지 확인하고, 달팽이관으로 얼마나 회전했는지 확인한 걸로 자신의 위치를 추정하려고 하는 것)

- 이를 Mapping 알고리즘을 통해서 처리한 결과(b) -> 지속적으로 mapping한 값과 odom을 비교하면서 Map을 작성하기 때문에 odom에서 생긴 오차도 보정을 해 줍니다.



- Grid Map
- 공간을 Cell로 나누어서 표현하며, 이 공간 구조는 고정되어 있음
- 각 Cell은 Occupied Space와 Free Space로 구분된다.
- Non-Parametric Model – 파라미터의 개수가 정해져있지 않음
- Feature map이 아닌 전체 센서 데이터를 모두 저장하는 방식
- 위의 이유로 인해 상당히 많은 Memory를 사용
- Feature map이 아니므로 Feature Dector에 의존 하지 않음 (Feature map은 Feature만 저장하기 때문에 memory를 효율적으로 사용함)

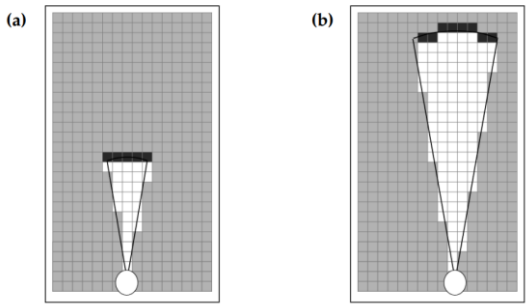


- Occupancy Grid Mapping
- 로봇의 t=0~T 까지의 모든 포즈(x)와 모든 측정값(z)가 주어져 있을 때, 이를 기반으로 아래의 식을 계산하는 방식

$$p(m|z_{1:t}, x_{1:t})$$

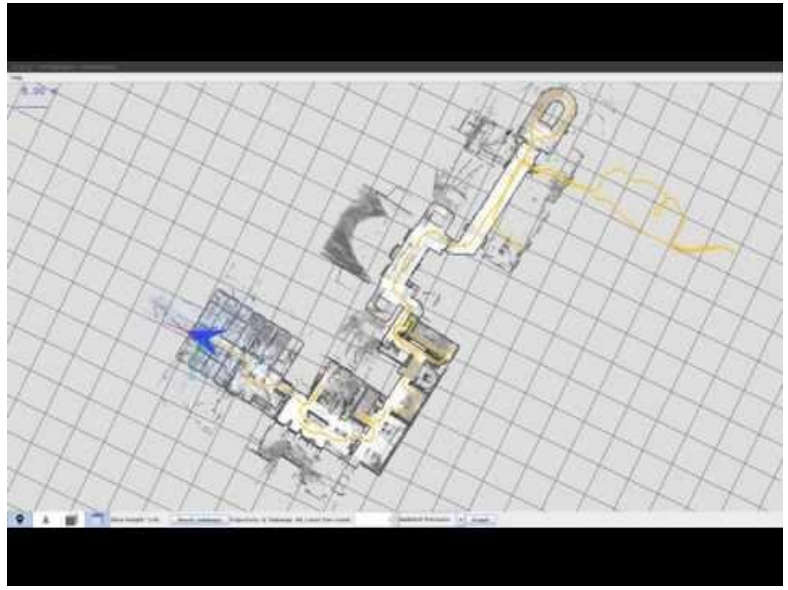
(m: map, z:측정 값, x: 로봇의 pose)

- t=0 ~ T 까지의 제어값(u)는 모든 포즈(x)에 포함되어 있기 때문에 사용하지 않는다.
- 지도 Grid Map을 이용하므로, 지도는 가 셀의 집합으로 되어있다.
- 이를 이용하여 Cell을 하나씩 채워 나가는 과정

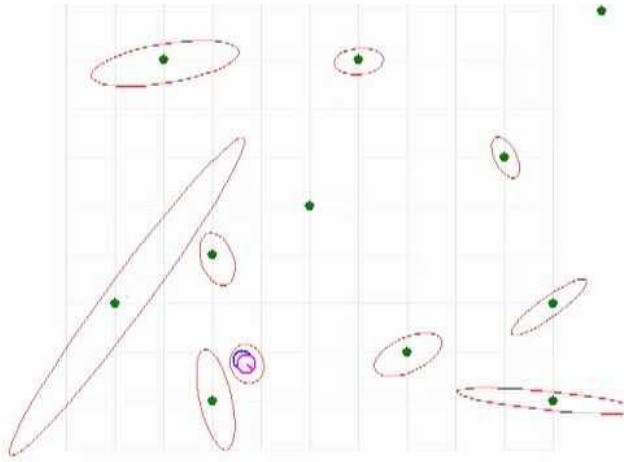


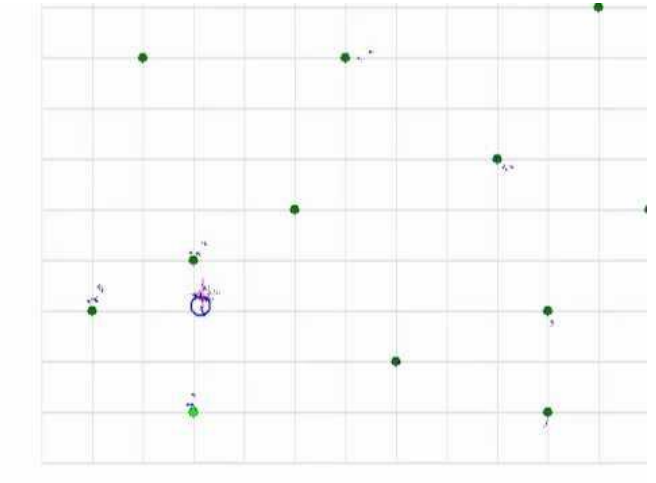
WeGo

- SLAM
- SLAM은 Occupancy Grid Mapping과 Locatlization을 동시에 진행
- 초기 위치를 특정 값([0, 0]과 같은)으로 정하고, 센서 데이터, 제어 데이터, 현재 위치 등의 데이터를 활용하여 지도 생성 및 위치를 측정을 동시에 진행
- http://www.youtube.com/watch?v=DM0dpHLhtX0



- SLAM
- 1990 ~ 2000 – Extended Kalman Filter 기반의 SLAM(pose-ek-slam)
- 2000 ~ 2007 – Particle Filter 기반의 SLAM (Gmapping)
- 2007 ~ 현재 – Maximum A Posteriori Estimation 기반의 SLAM(Cartographer)
- http://www.youtube.com/watch?v=vCVS9WAffi4
- http://www.youtube.com/watch?v=-hXEYh0O_XA





Hector mapping

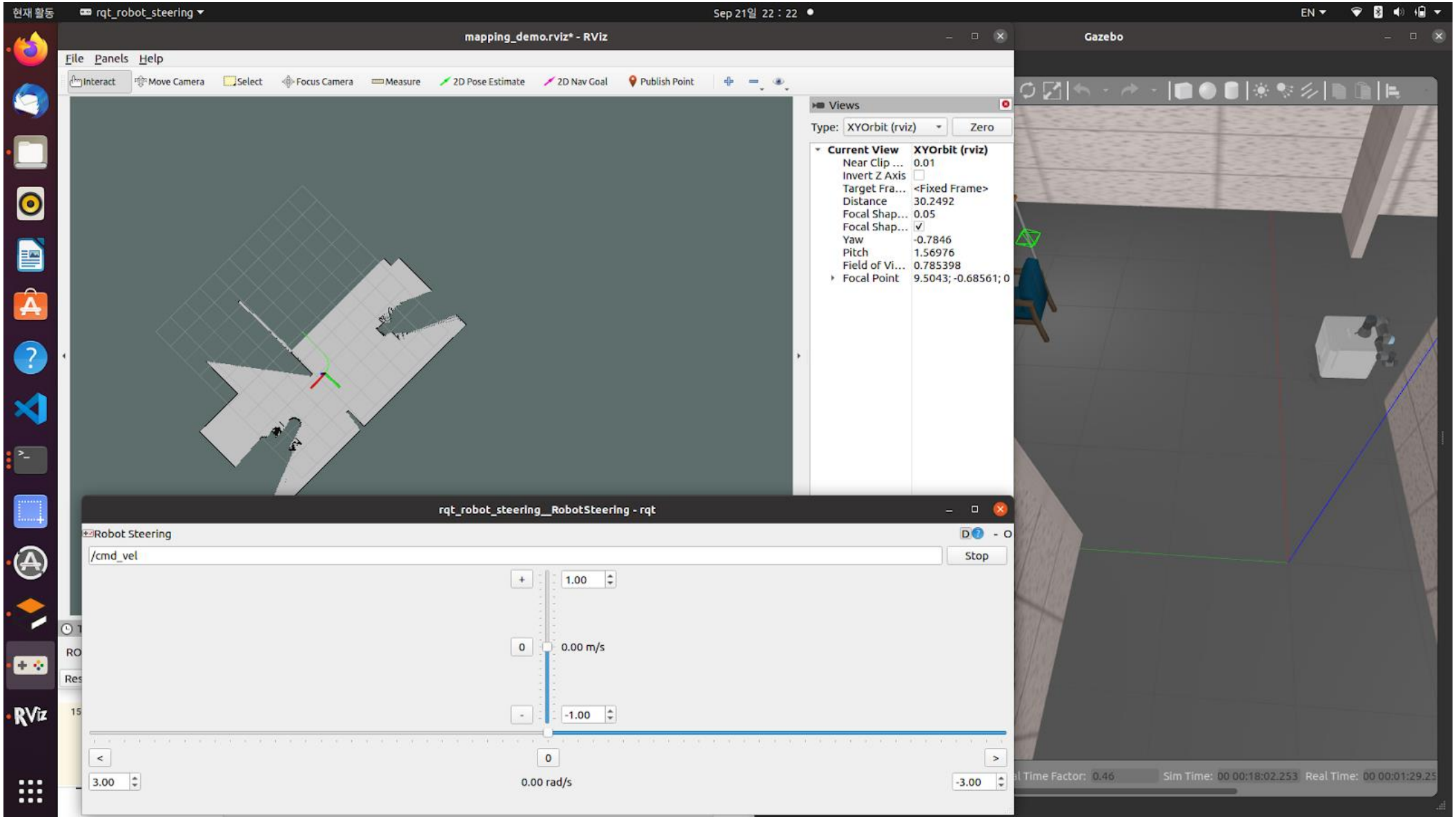
Hector mapping

- Hector mapping – Lidar Only SLAM using LaserScan Data
- EKF, Particle Filter, Maximum A Posteriori Estimation 방법 중 Maximum A Posteriori Estimation을 기반으로 한 SLAM알고리즘을 사용합니다.
- Input - 2D LIDAR의 거리 데이터
- Hector SLAM 알고리즘은 Lidar의 정보만을 사용하여, Mapping 및 Localization이 가능하며, 따라서 Hand-Held Mapping이 가능하다는 특징이 있음(Odom 값이 없어도 Map을 그릴 수 있음)
- Lidar의 정보만을 사용하므로, Feature가 부족한 개활지 및 복도 등의 환경에서는 위치를 제대로 잡지 못하는 문제가 발생

Hector mapping

- $ roslaunch tracer_gazebo_sim tracer_world.launch
- $ rosrun rqt_robot_steering rqt_robot_steering
- $ roslaunch hector_slam_launch hector_mapping.launch
- $ rosrun map_server map_saver

Hector mapping



Gmapping

WeGo

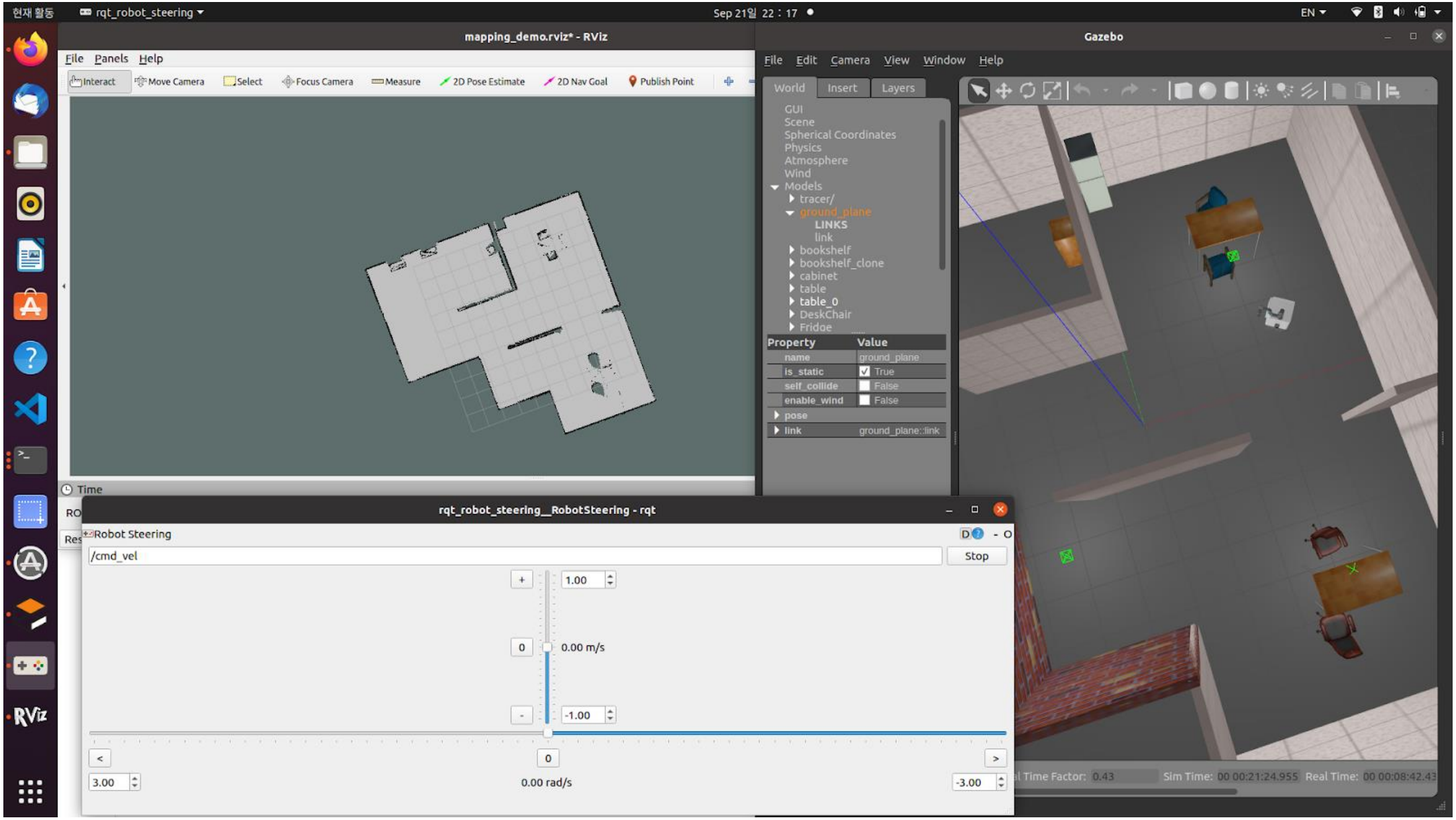
•Gmapping

- EKF, Particle Filter, Maximum A Posteriori Estimation방법 중 Particle Filter를 기반으로 한 SLAM알고리즘
- Input - 플랫폼의 움직임에 해당하는 정보인 Odometry, 2D LiDAR의 거리 데이터
- Odometry 정보를 사용하므로, 위치가 과도하게 튀는 현상은 없음
- Odometry 정보에 대한 의존도가 높으므로, Odometry 데이터가 오차가 심할 경우, 제대로 Mapping이 되지 않거나, Mapping 중간에 지도가 틀어질 수 있음
- Odometry에 오차가 있을 경우에도 주변 환경이 특징이 많아, Map - Sensor Data 사이의 Matching이 잘 되는 경우에 우수하게 동작함

Gmapping

- $ roslaunch tracer_gazebo_sim tracer_world.launch
- $ rosrun rqt_robot_steering rqt_robot_steering
- $ roslaunch hector_slam_launch gmapping.launch
- $ rosrun map_server map_saver

Gmapping


현재 활동
rqt_robot_steering
Sep 21일 22 : 17
EN
mapping_demo.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Time
Gazebo
File Edit Camera View Window Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
tracer/
ground_plane
LINKS
link
bookshelf
bookshelf_clone
cabinet
table
table_0
DeskChair
Fridge
Property
Value
name
ground_plane
is_static
True
self_collide
False
enable_wind
False
pose
link
ground_plane::link
rqt_robot_steering__RobotSteering - rqt
Robot Steering
/cmd_vel
Stop
1.00
0.00 m/s
-1.00
3.00
0.00 rad/s
-3.00
Time Factor: 0.43
Sim Time: 00 00:21:24.955
Real Time: 00 00:08:42.43

본사(기술연구소 및 사무실) : 16914 경기도 용인시 기흥구 구성로 357(청덕동) 용인테크노밸리 B동 513호

기술연구소(서울) : 04799 서울특별시 성동구 성수동2가 280-13, 삼환디지털벤처타워 401호

대표전화 : 031 – 229 – 3553

팩스 : 031 – 229 – 3554

제품문의: go.sales@wego-robotics.com

기술문의: go.support@wego-robotics.com