

# Gazebo Navigation

# 목차

- Navigation 기초
- Navigation 실행
- Navigation Parameter 수정

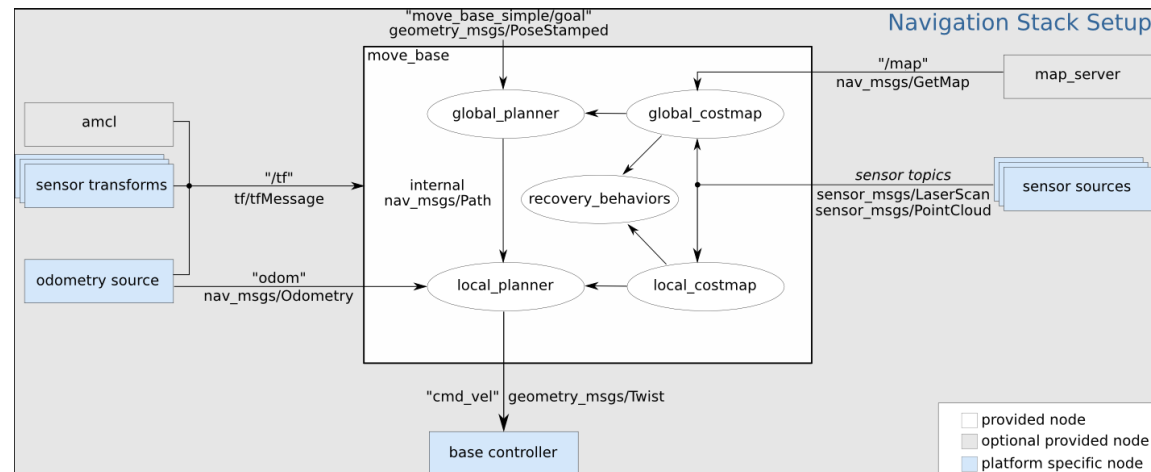
# Navigation 기초

## • Navigation 기초

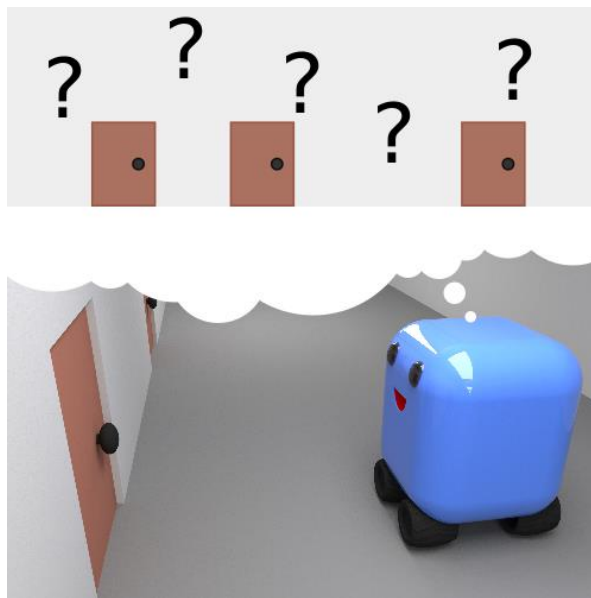
- Navigation은 다음과 같이 나뉘져 있습니다.
- Localization: Map을 기반으로 현재 로봇의 위치를 추정하는 부분입니다.
- Path Planning
  - Global Path Planning: 현재 위치를 기반으로 목표 위치까지의 경로를 생성하는 부분
  - Local Path Planning: Global Path를 따라가기 위해서 Local 적인 부분에서 경로를

생성하는 부분

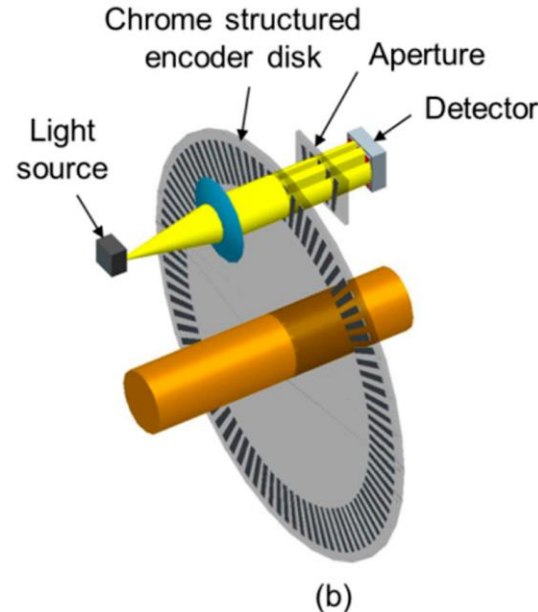
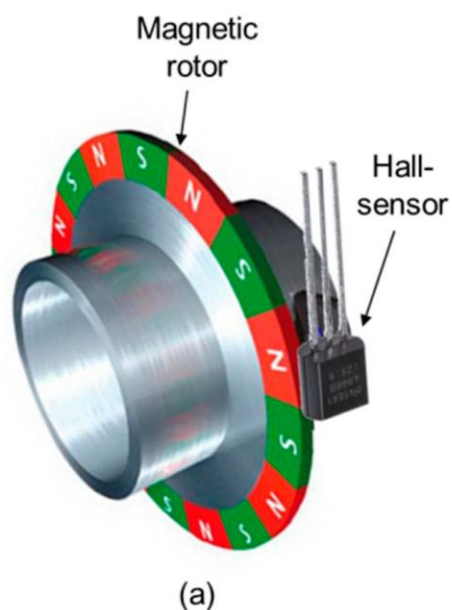
- Path following : 그려진 Path를 어떠한 방식으로 따라갈 것인지



- Localization
- Navigation은 Mobile Robot 관점에서 핵심적인 기능 중 하나입니다.
- Navigation을 위해서는 로봇의 현재 위치를 파악하는 것이 중요합니다.
- 주어진 지도 상에서 로봇의 현재 위치를 파악하는 기술을 Localization(지역화) 기술이라고 합니다. 또는 위치 추정(Pose Estimation)이라고도 합니다.



- Localization – Odometry(Encoder 값을 이용, 주행 기록계)
- 로봇의 형태 및 자유동 따라서 추정해야 하는 차원 수가 달라집니다.
- 일반적인 2D 환경에서 이동하는 Mobile Robot의 경우는 지도 상에서의 위치( $x, y$ ) 및 헤딩 각( $\theta$ )의 세가지 정보를 Pose  $xx_t = (x, y, \theta)^T$  와 같이 표시한다.
- 사용하는 센서로는 내부의 이동을 측정하는 Odometer(주행 기록계) 및 Lidar 센서 데이터 두 가지를 사용한다



- Differential Drive에서 Encoder 값으로 Odometry를 계산하는 방법

$$\dot{x} = \frac{r}{2}(v_l + v_r)\cos(\theta)$$

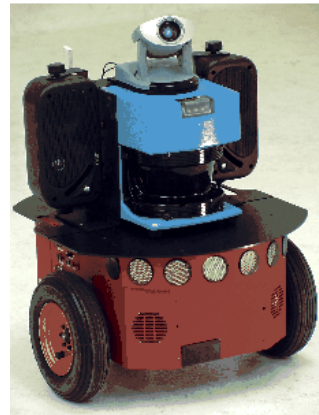
$$\dot{y} = \frac{r}{2}(v_l + v_r)\sin(\theta)$$

$$\dot{\theta} = \frac{r}{L}(v_r - v_l)$$

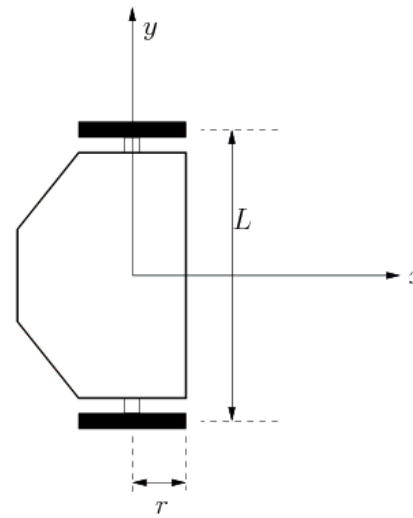
$$\dot{x} = v\cos(\phi)$$

$$\dot{y} = v\sin(\phi)$$

$$\dot{\phi} = \omega$$



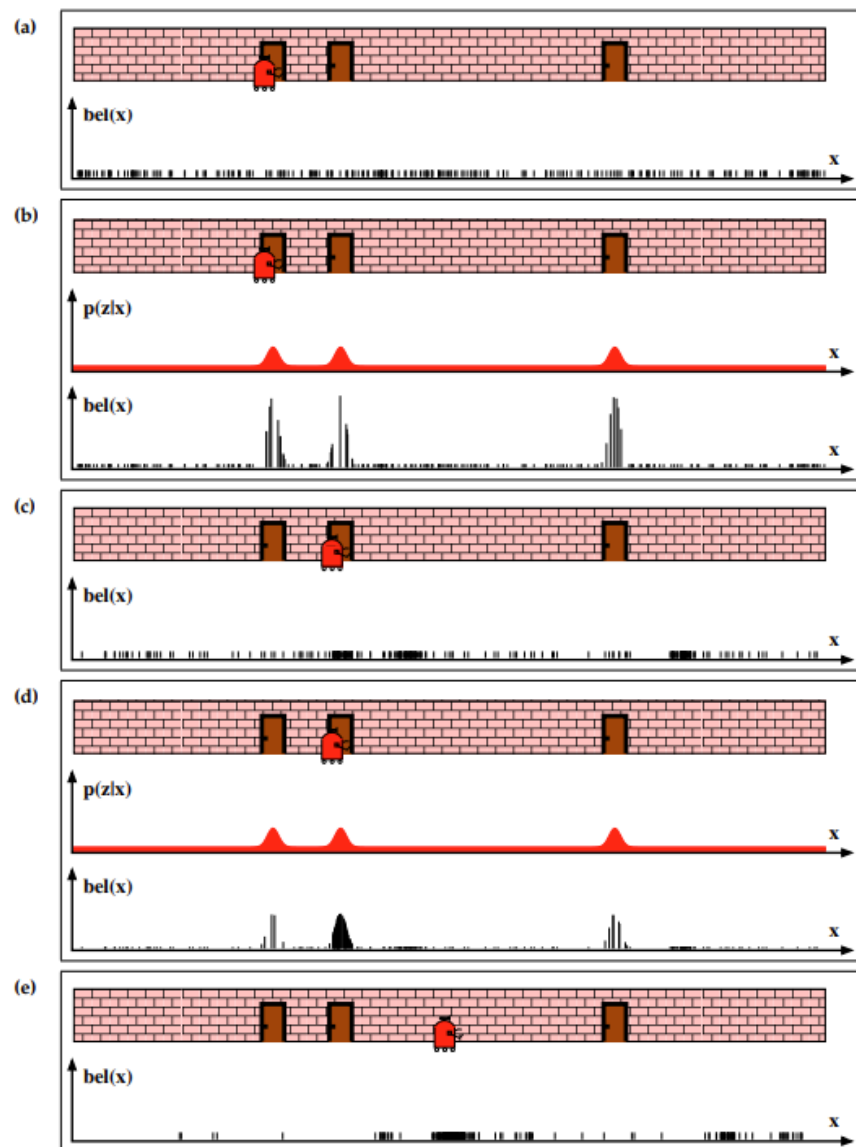
(a)



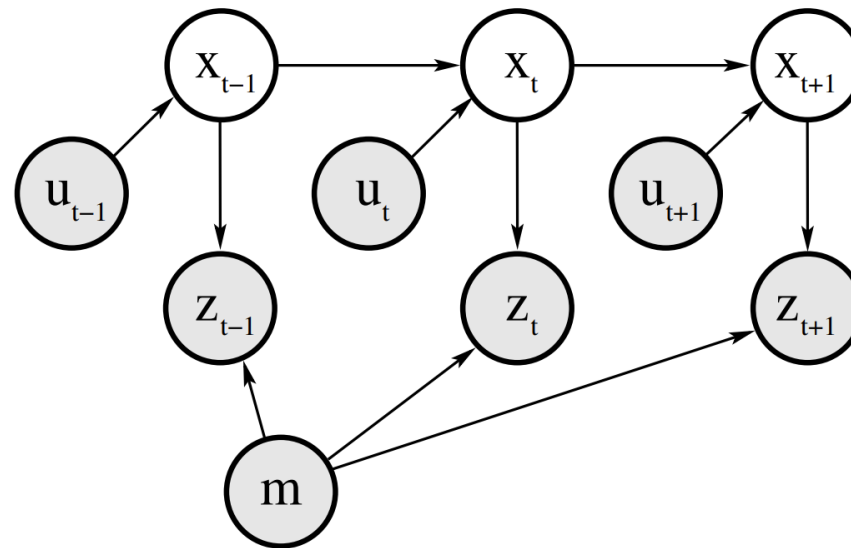
(b)

- Localization – Lidar(Lidar 값을 이용하여 현재의 map과 Lidar data를 비교하며 자신의 위치를 찾음)
- Monte Carlo Localization(Particle Filter)
- 전체 지도에서 균일하게 확률 분포를 생성한다.
- 로봇의 센서를 통해 주변 환경을 확인하고 확률 분포를 업데이트한다.
- 로봇을 이동시키고, 로봇의 이동한 값을 이용하여 확률 분포를 업데이트한다.
- 2~3의 과정을 반복하여 주렴시킨다.

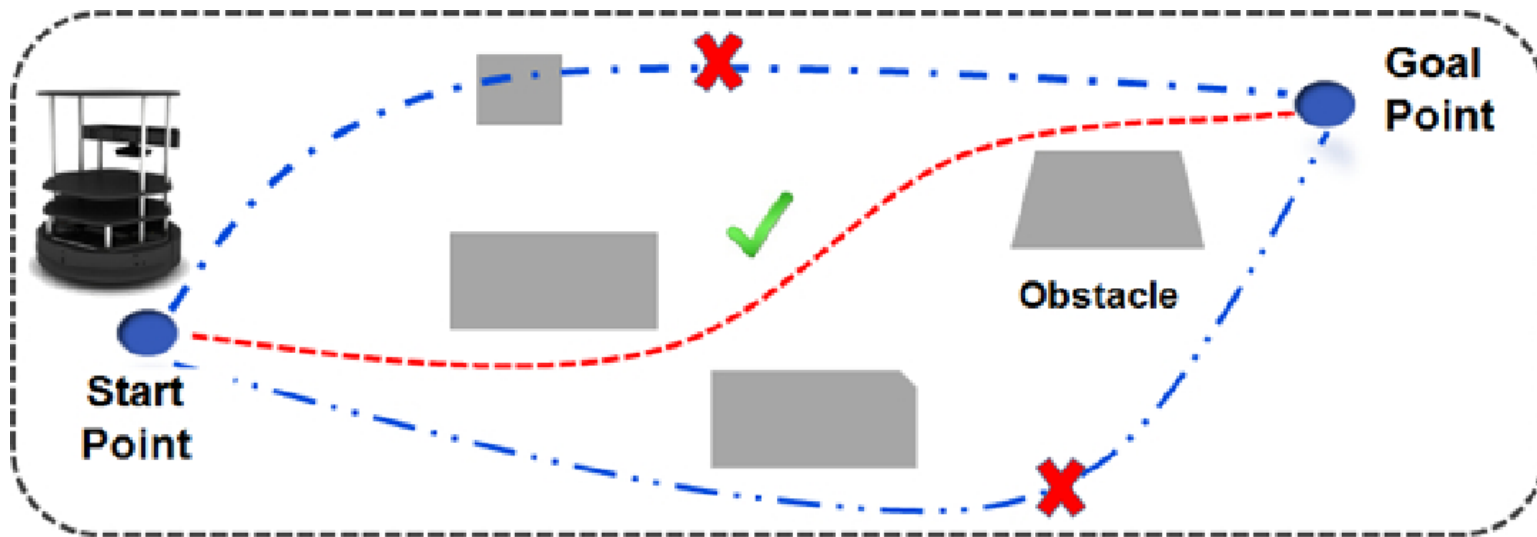




- Lidar 와 Odometry를 이용하여
- Mobile Robot Localization Graphical Model
- Robot State
- Map Data
- Measurement (LiDAR)
- Control Data(Odometry)



- Path Planning
- Path Planning이란, 지도 상의 로봇의 위치와 목적 장소가 주어졌을 때, 로봇의 형태 및 크기, 주변 상황을 고려하여, 이동 경로를 생성하는 기술
- 로봇 외의 다양한 상황에서도 경로 찾기 및 모션 계획에 사용된다.
- Planner는 크게 Global Planner와 Local Planner로 나뉜다.



- Global Path Planner
- 전체 지도에서 로봇의 현재 위치와 목표 두 지점 두 가지 정보를 입력으로, 지도 상에 있는 장애물과 충돌 없이 도착하는 경로를 생성
- 입력으로 필요한 값은 지도, 지도 상의 출발 위치, 지도 상의 도착 위치이며, 현재 로봇이 측정하고 있는 주변 환경과는 무관하게 동작한다(센서 데이터와는 무관)



- Local Path Planner
- 입력 값으로 현재 로봇의 State(위치 및 속도 포함), Sensor Data(주변 상황), Global Path(Global Path Planner가 생성)를 입력으로 받는다.
- 현재 로봇의 위치에서 Global path를 추종하는 최적의 Local Path를 생성

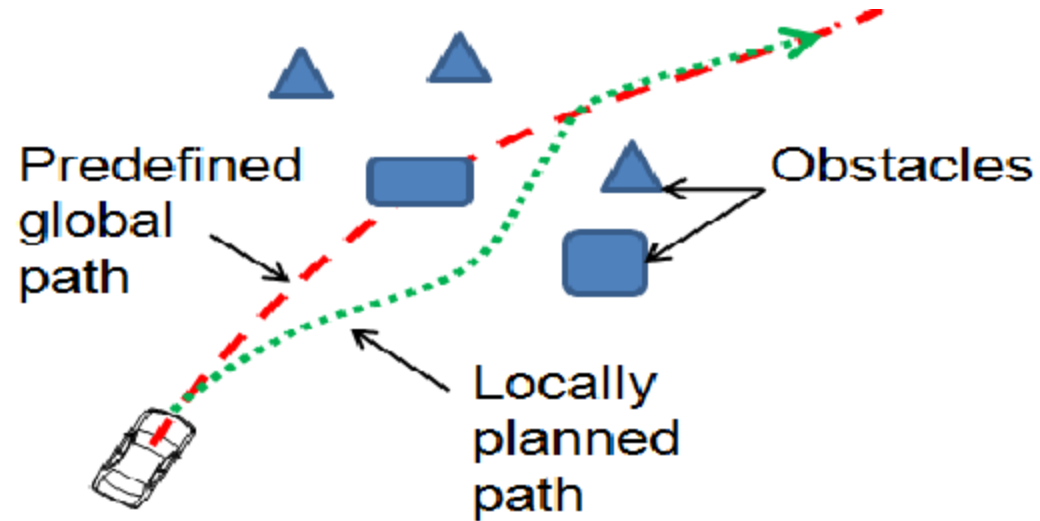
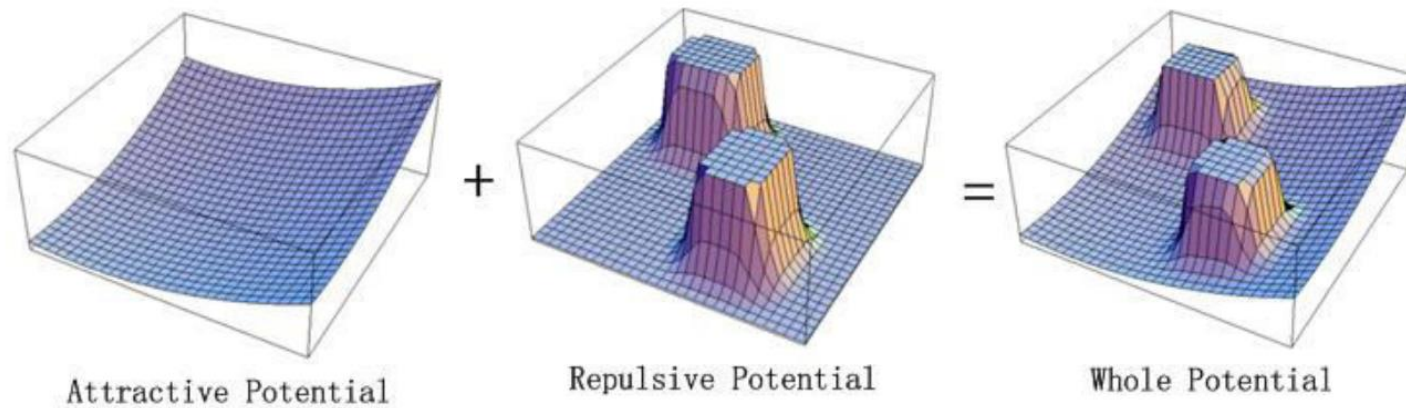


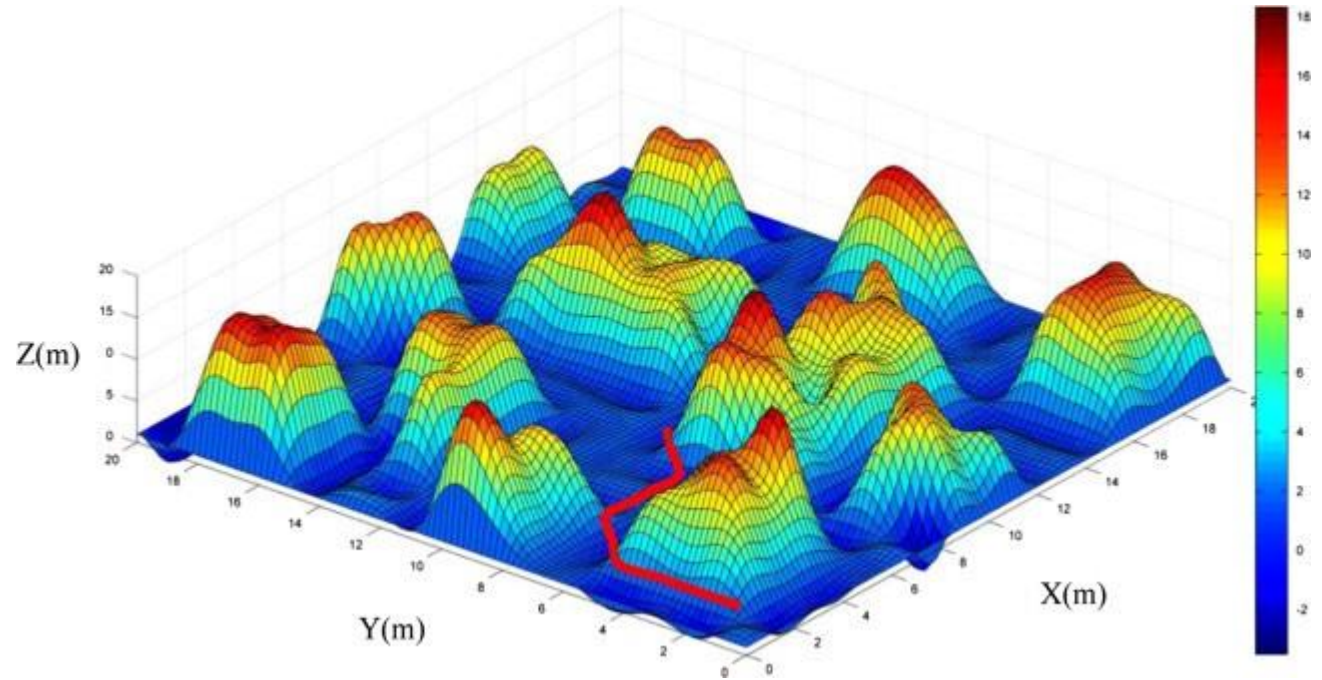
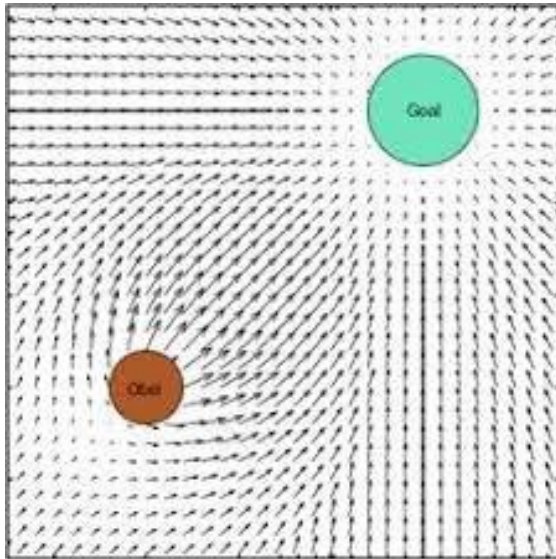
Fig 1 Local path planning to track a planned global path

- 여기서는 Grid 기반의 Path Planning을 의미하며, 이전에 생성한 Occupancy Grid Map을 기반으로 생성하는 Path를 의미합니다.
- Grid 기반의 Planning 방법은 크게 Potential Field 방식. Graph 기반 Search 알고리즘. RRT 등을 활용할 수 있습니다.

- **Potential Filed**
- 출발 지점의 Potential을 가장 높게, 도착 지점의 Potential을 가장 낮게 설정한다.
- 중간에 있는 장애물의 위치에 대해서도 Potential를 Gaussian 함수와 같은 형태로 생성
- 위 두개의 Potential을 합쳐서 전체 Potential Field Map을 생성하고, 각 위치에서의 전체 Map에서의 Gradient를 계산하여 이를 연결하는 방법을 통해서 경로를 생성

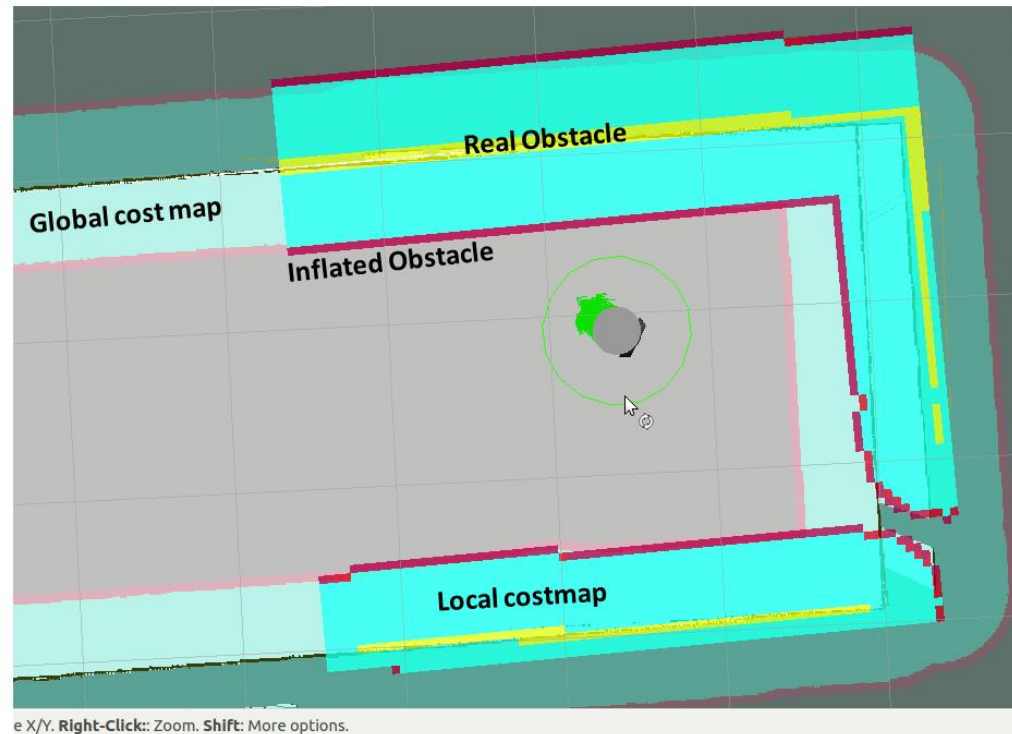


- Potential Filed
- Gradient Descent(경사 하강법) 방식을 통해서 경로를 탐색
- Gradient(기울기)를 이용하는 방식이므로, Local Minima(극소점), Saddle Point(인장점) 으로 인한 문제가 발생함.

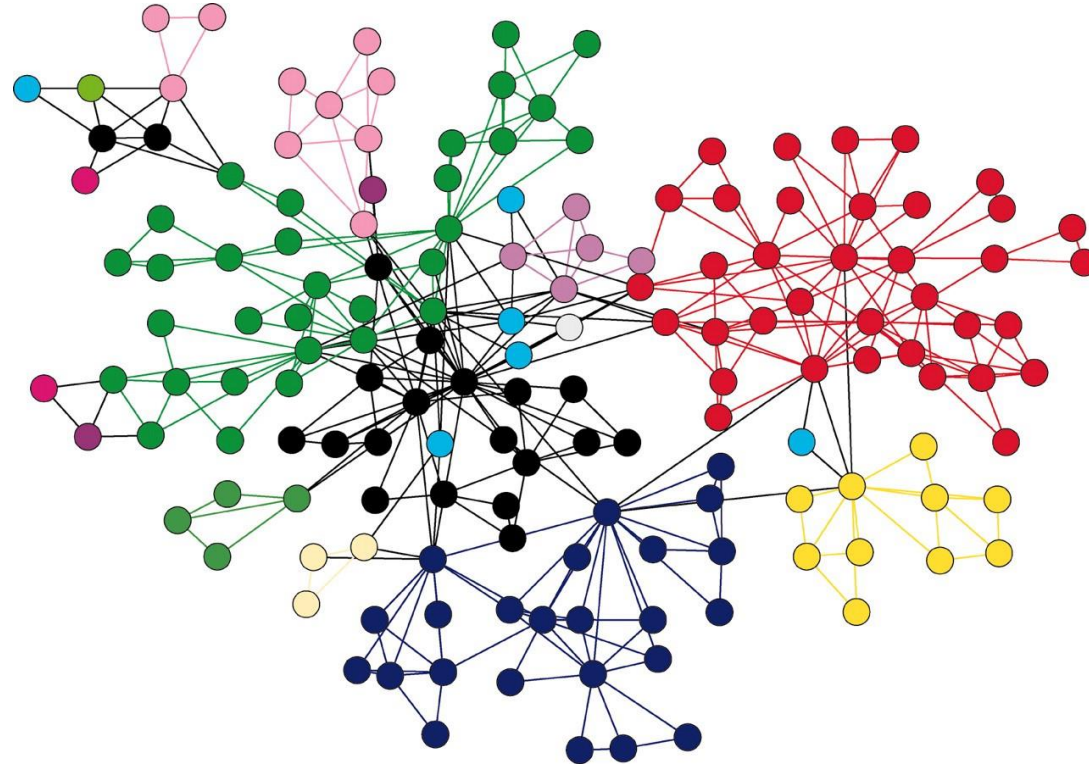




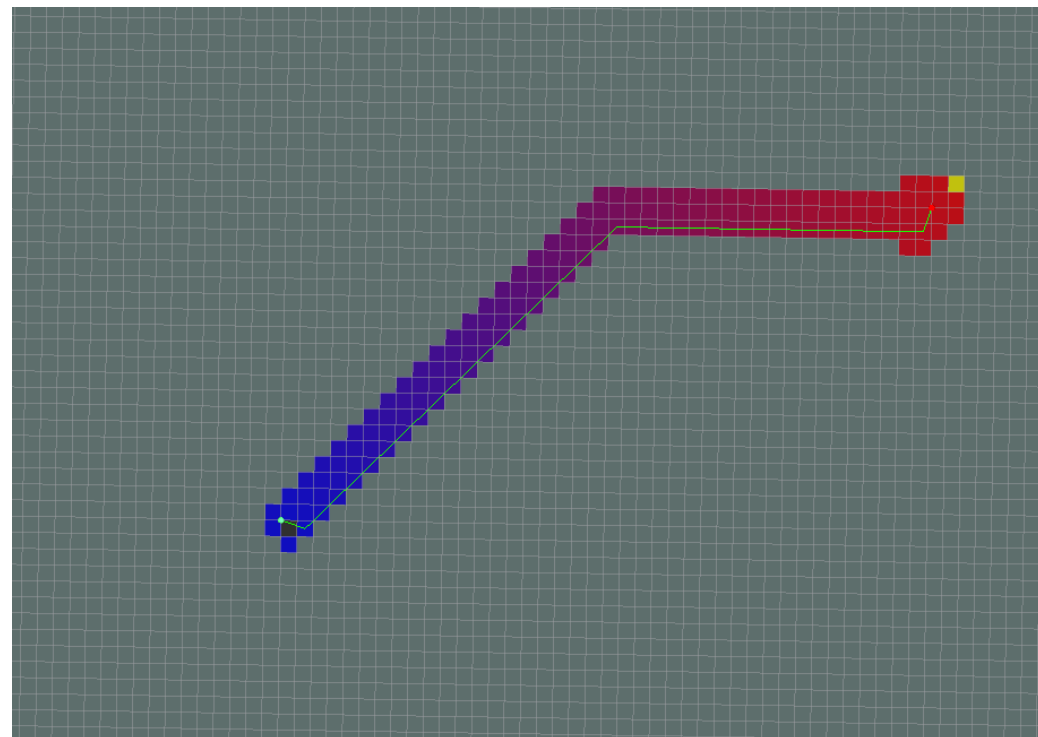
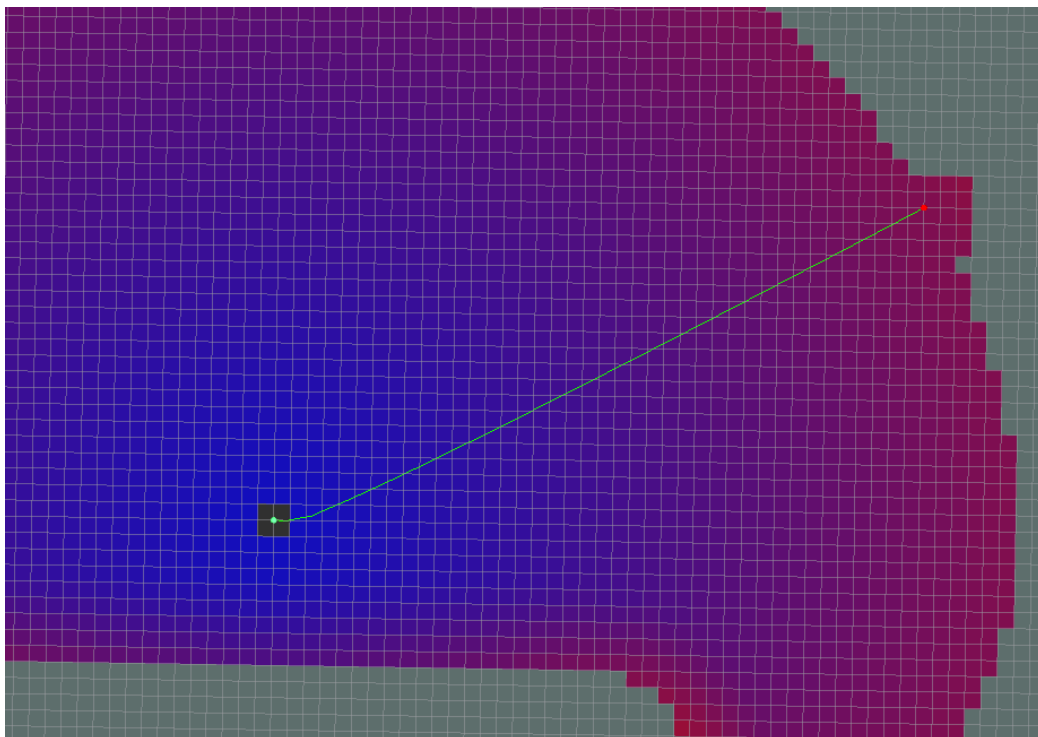
- Graph-Based Search Algorithm
- Dijkstra's algorithm, A\*, Breadth-First Search, Depth-First Search 등 (전부 Graph에 관한 알고리즘)
- 우선적으로 Obstacle Growing(장애물 설정)을 적용해야 하며, Growing Region(설정된 장애물의 영역)의 경우, 이동하는 로봇을 원형으로 가정하였을 때, 최소 원의 반지름 이상이 되어야 한다.



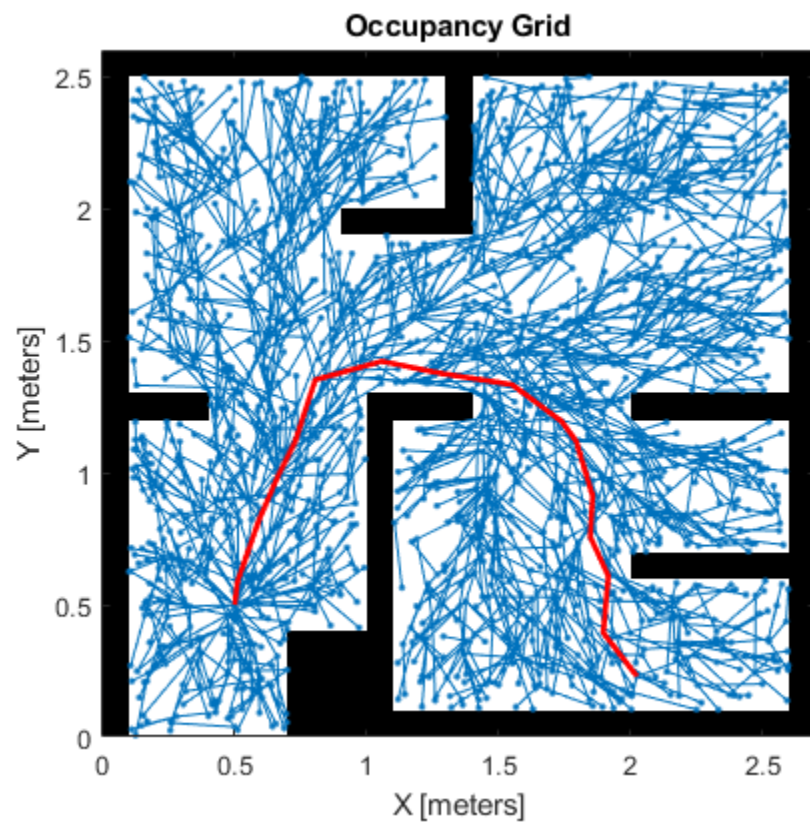
- Graph-Based Search Algorithm
- Growing된 결과를 이용하여, 이를 Graph로 변환한다.
- Graph 변환시에는 4-neighbor인지 8-neighbor인지 확인하여 변환한다.



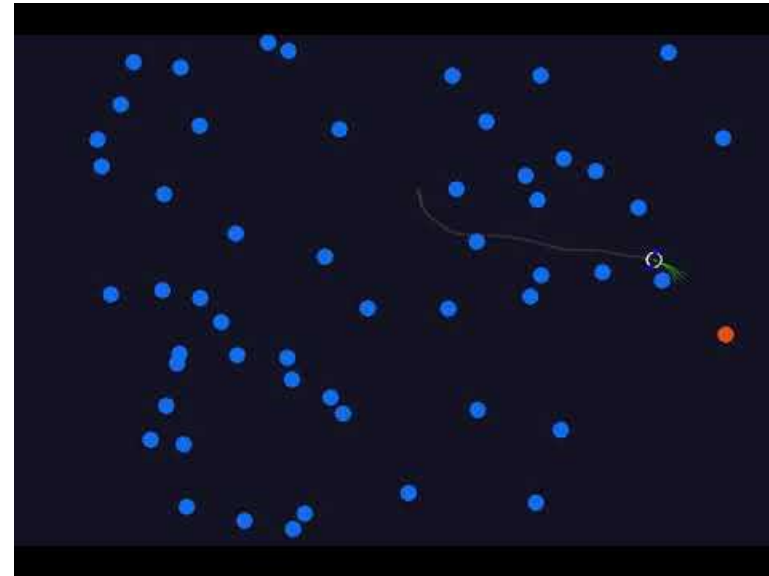
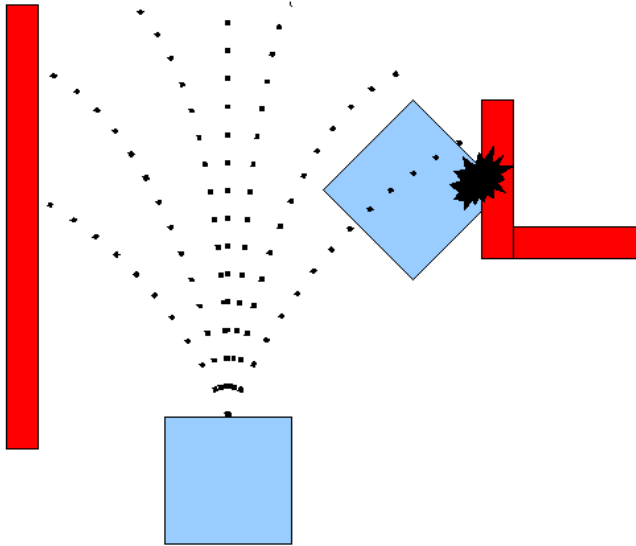
- Graph-Based Search Algorithm
- 변환 후, 입력으로 받은 시작 장소와 목표 장소를 기반으로
- Graph-based Search Algorithm을 적용하여 경로를 찾는다.



- RRT(Rapidly-Exploring Random Tree)
- 시작 장소로부터 Random Tree를 생성하여, 목표 장소에 도착하는 경로를 생성



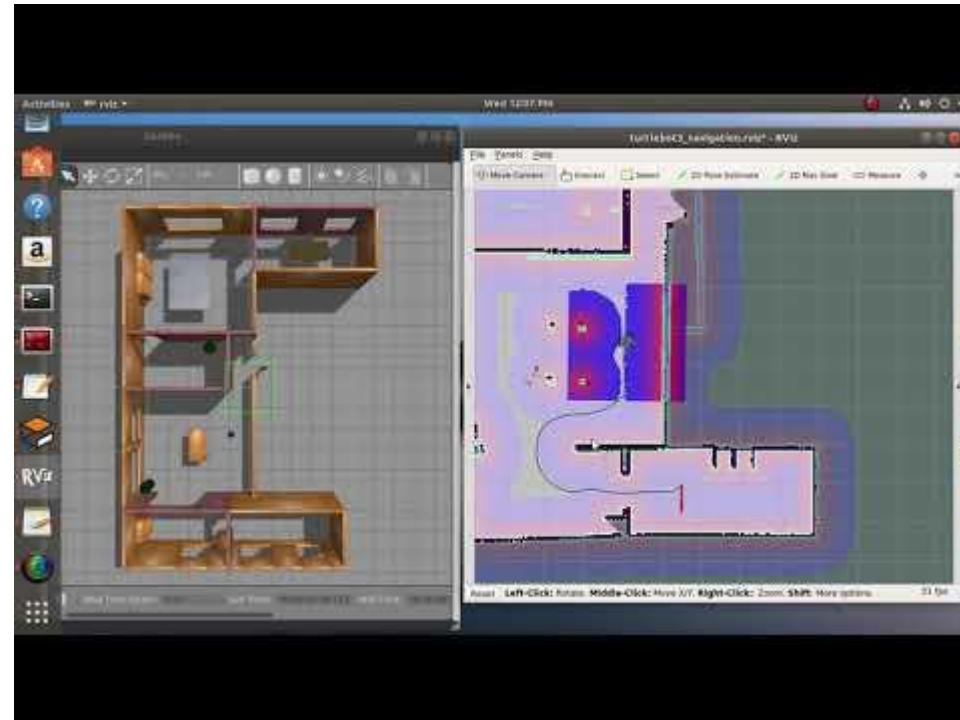
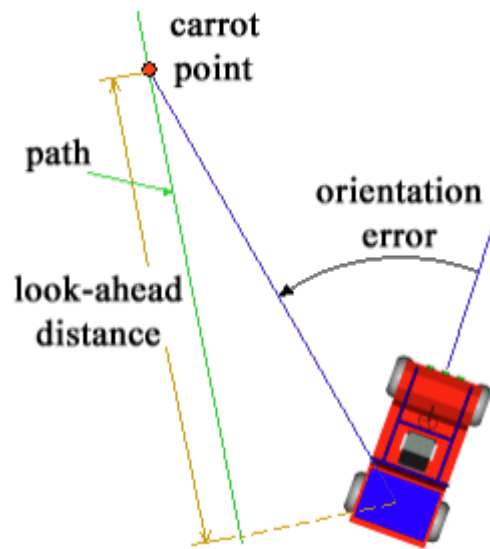
- Local Planner
- Dynamic Window Approach
- Sampling을 통해 다양한 선속도 및 회전 속도를 생성하고 이를 기반으로 시뮬레이션을 진행하여, 점수 측정을 통해 얻은 높은 점수를 얻은 경로로 주행하는 방식
- <http://www.youtube.com/watch?v=Mdg9ElewwA0>



- Time Elastic Band Planner
- 시간, 장애물, 역학적인 제한(Steering Angle 등)을 고려하여 최적의 경로를 생성
- <http://www.youtube.com/watch?v=e1Bw6JOgHME>



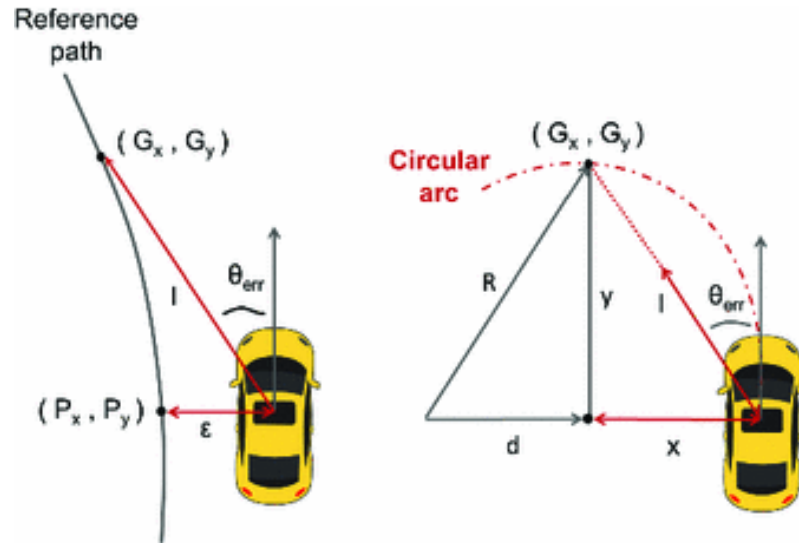
- Active Scene Recognition Local Planner
- Follow the Carrot과 유사한 형태의 Planner
- Carrot Point를 지정하여, Orientation Error를 최소화하는 Controller
- <http://www.youtube.com/watch?v=QvmeDFhwg7Q>



- Path Following
- Local Planner와 유사하게 Local Planner의 경로를 추종하는 제어 명령을 생성
- 적용하는 알고리즘이 같아도 사용하는 모델에 따라 출력 결과가 달라질 수 있음
- 일반적으로 사용하는 모델은 Ackerman Steering Model(Car-Like Model) 및 Differential-Drive Model두 가지가 모바일 로봇에서는 일반적인 형태

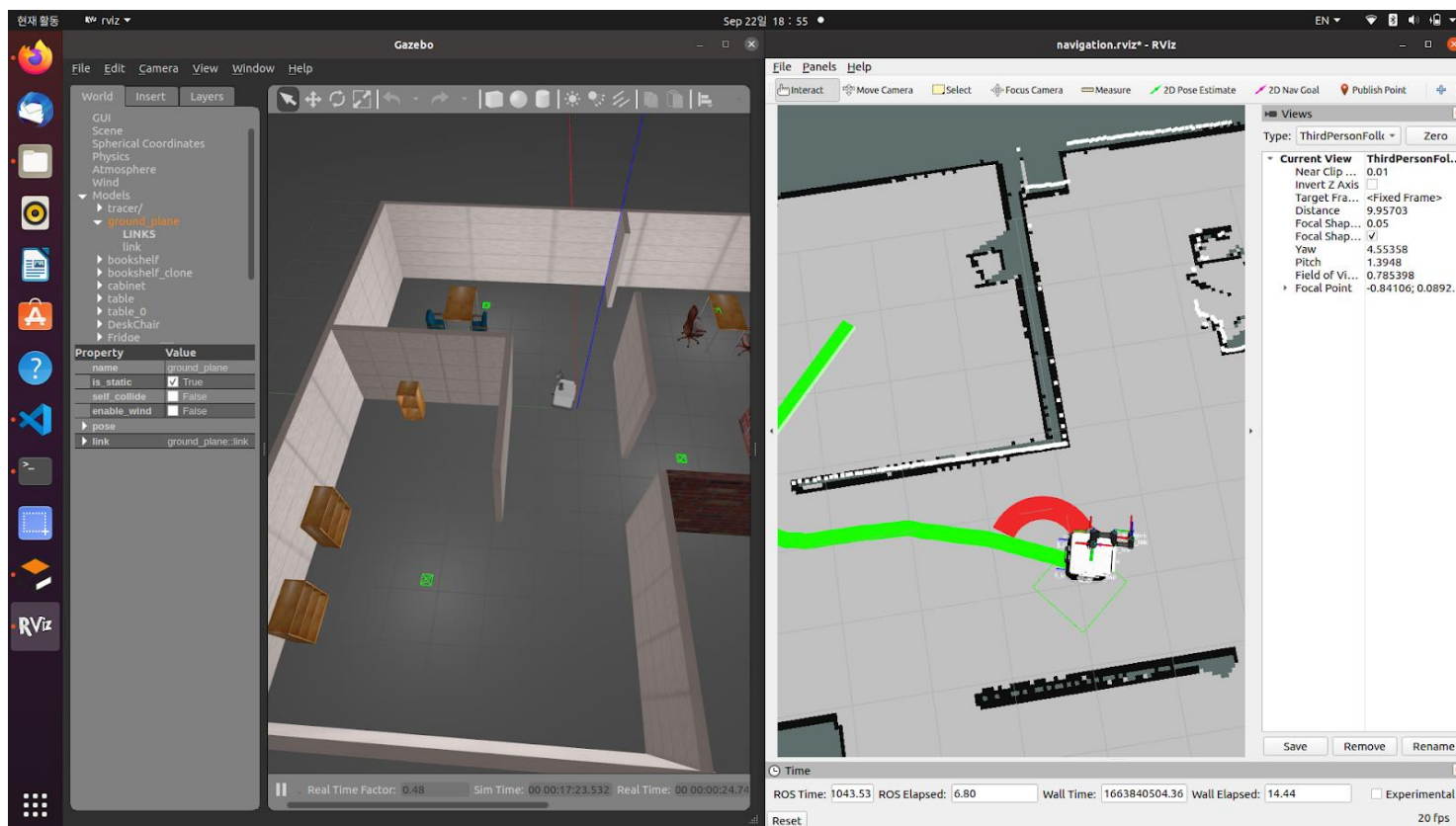


- Path Following
- Pure Pursuit 알고리즘
- 현재 위치에서 목표한 지점까지 이동하기 위한 원궤적을 생성하고, 이 원 궤적을 따라가기 위한 제어 값을 계산하는 알고리즘
- Ackerman Steering Model에서는 바퀴의 각도를 구하는 것이 목적
- 앞서 설명한 Model을 적용하면 제어 값을 계산할 수 있음



# Navigation 실행

- `$ roslaunch tracer_gazebo_sim tracer_world.launch`
- `$ roslaunch wego navigation.launch`



- 이번에는 다음과 같은 방식으로 Navigation을 실행 할 수 있습니다.
- Global Planner – Dijkstra, A\*
- Local Planner – DWA, Time Elastic Band Planner

어떤 것들을 설정했을 때, 해당 planner로 설정이 되는지 알아보겠습니다.

- 다음 파일들을 수정해야 합니다. 또한 뒤에서 설명을 쉽게 하기 위해서 순서대로 1번 2번 3번이라고 말하겠습니다.

~/catkin\_ws/src/scount\_mini\_2dnav/launch/move\_base\_only.launch

~/catkin\_ws/src/scount\_mini\_2dnav/params/base\_local\_planner\_params.yaml

~/catkin\_ws/src/scount\_mini\_2dnav/params/teb\_local\_planner\_params.yaml

- Dijkstra, DWA 실행 할 때의 설정

1번 파일에 13번째 줄을 주석처리 하고 사용합니다.

2번 파일의 48번째 줄에서 53번째 줄을 주석처리하고 사용합니다.

```

1 <launch>
2 <!--node name="map_server" pkg="map_server" type="map_server" args="$(find scout_mini_2dnav)/map/map.yaml"/-->
3
4 <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
5
6   <rosparam file="$(find scout_mini_2dnav)/params/modify_costmap_common_params.yaml" command="load" ns="global_costmap" />
7   <rosparam file="$(find scout_mini_2dnav)/params/modify_costmap_common_params.yaml" command="load" ns="local_costmap" />
8
9   <rosparam file="$(find scout_mini_2dnav)/params/map_nav_params/modify_local_costmap_params.yaml" command="load" />
10  <rosparam file="$(find scout_mini_2dnav)/params/map_nav_params/modify_global_costmap_params.yaml" command="load" />
11
12  <rosparam file="$(find scout_mini_2dnav)/params/base_local_planner_params.yaml" command="load" />
13  <!-- <rosparam file="$(find scout_mini_2dnav)/params/teb_local_planner_params.yaml" command="load" /> -->
14
15
16  <rosparam file="$(find scout_mini_2dnav)/params/move_base_params.yaml" command="load" />
17
18  <param name="base_global_planner" type="string" value="navfn/NavfnROS" />
19
20  <param name="base_local_planner" value="base_local_planner/TrajectoryPlannerROS"/>
21  <param name="clearing_rotation_allowed" value="true" />
22
23  <!--remap from="odom" to="/odometry/filtered" /-->
24 </node>
25 </launch>
26

```

```

28 controller_frequency: 20.0
29
30 # Trajectory scoring parameters
31 meter_scoring: true # Whether the gdist_scale and pdist_scale parameters should assume that goal_d
32 occdist_scale: 0.1 #The weighting for how much the controller should attempt to avoid obstacles.
33 pdist_scale: 2.5 # The weighting for how much the controller should stay close to the path it
34 gdist_scale: 1.0 # The weighting for how much the controller should attempt to reach its local
35
36 heading_lookahead: 0.1 #How far to look ahead in meters when scoring different in-place-rotation
37 heading_scoring: false #Whether to score based on the robot's heading to the path or its distance
38 heading_scoring_timestep: 0.8 #How far to look ahead in time in seconds along the simulated traj
39 dwa: true #Whether to use the Dynamic Window Approach (DWA) or whether to use Trajectory Rollout
40 simple_attractor: false
41 publish_cost_grid_pc: true
42
43 #Oscillation Prevention Parameters
44 oscillation_reset_dist: 0.25 #How far the robot must travel in meters before oscillation flags are
45 escape_reset_dist: 0.1
46 escape_reset_theta: 0.1
47
48 # BaseGlobalPlanner:
49 # allow_unknown: false
50 # use_dijkstra: false #Use A* instead
51 # use_quadratic: true
52 # use_grid_path: false
53 # old_navfn_behavior: false
54

```

- A\*, DWA 실행 할 때의 설정

1번 파일에 13번째 줄을 주석처리 하고 사용합니다.

2번 파일의 48번째 줄에서 53번째 줄을 주석을 풀어주고 사용합니다.

```

1 <launch>
2 <!--node name="map_server" pkg="map_server" type="map_server" args="$(find scout_mini_2dnav)/map/map.yaml"/-->
3
4 <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
5
6   <roscpp param file="$(find scout_mini_2dnav)/params/modify_costmap_common_params.yaml" command="load" ns="global_costmap" />
7   <roscpp param file="$(find scout_mini_2dnav)/params/modify_costmap_common_params.yaml" command="load" ns="local_costmap" />
8
9   <roscpp param file="$(find scout_mini_2dnav)/params/map_nav_params/modify_local_costmap_params.yaml" command="load" />
10  <roscpp param file="$(find scout_mini_2dnav)/params/map_nav_params/modify_global_costmap_params.yaml" command="load" />
11
12  <roscpp param file="$(find scout_mini_2dnav)/params/base_local_planner_params.yaml" command="load" />
13  <!-- <roscpp param file="$(find scout_mini_2dnav)/params/teb_local_planner_params.yaml" command="load" /> -->
14
15  <roscpp param file="$(find scout_mini_2dnav)/params/move_base_params.yaml" command="load" />
16
17  <param name="base_global_planner" type="string" value="navfn/NavfnROS" />
18
19  <param name="base_local_planner" value="base_local_planner/TrajectoryPlannerROS"/>
20  <param name="clearing_rotation_allowed" value="true" />
21
22  <!--remap from="odom" to="/odometry/filtered" /-->
23
24 </node>
25 </launch>
26

```

```

29
30 # Trajectory scoring parameters
31 meter_scoring: true # Whether the gdist_scale and pdist_scale parameters should assume that goal_distance a
32 occdist_scale: 0.1 #The weighting for how much the controller should attempt to avoid obstacles. default 0
33 pdist_scale: 2.5 # The weighting for how much the controller should stay close to the path it was give
34 gdist_scale: 1.0 # The weighting for how much the controller should attempt to reach its local goal, al
35
36 heading_lookahead: 0.1 #How far to look ahead in meters when scoring different in-place-rotation trajector
37 heading_scoring: false #Whether to score based on the robot's heading to the path or its distance from the
38 heading_scoring_timestep: 0.8 #How far to look ahead in time in seconds along the simulated trajectory wh
39 dwa: true #Whether to use the Dynamic Window Approach (DWA)_ or whether to use Trajectory Rollout
40 simple_attractor: false
41 publish_cost_grid_pc: true
42
43 #Oscillation Prevention Parameters
44 oscillation_reset_dist: 0.25 #How far the robot must travel in meters before oscillation flags are reset (d
45 escape_reset_dist: 0.1
46 escape_reset_theta: 0.1
47
48 BaseGlobalPlanner:
49   allow_unknown: false
50   use_dijkstra: false #Use A* instead
51   use_quadratic: true
52   use_grid_path: false
53   old_navfn_behavior: false
54

```

- Dijkstra, Time Elastic Band Planner 실행할 때의 설정

1번 파일에 12번째 줄을 주석처리 하고 사용합니다.

3번 파일의 79번째 줄에서 84번째 줄을 주석 처리하고 사용합니다.

```

WeMMI_Gazebo > scout_mini_2dnav > launch > move_base_only.launch
1 <launch>
2 <!--node name="map_server" pkg="map_server" type="map_server" args="$(find scout_mini_2dnav)/map/map.yaml/-->
3
4 <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
5
6   <rosparam file="$(find scout_mini_2dnav)/params/modify_costmap_common_params.yaml" command="load" ns="global_costmap" />
7   <rosparam file="$(find scout_mini_2dnav)/params/modify_costmap_common_params.yaml" command="load" ns="local_costmap" />
8
9   <rosparam file="$(find scout_mini_2dnav)/params/map_nav_params/modify_local_costmap_params.yaml" command="load" />
10  <rosparam file="$(find scout_mini_2dnav)/params/map_nav_params/modify_global_costmap_params.yaml" command="load" />
11
12  <!-- <rosparam file="$(find scout_mini_2dnav)/params/base_local_planner_params.yaml" command="load" /> -->
13  <rosparam file="$(find scout_mini_2dnav)/params/teb_local_planner_params.yaml" command="load" />
14
15
16  <rosparam file="$(find scout_mini_2dnav)/params/move_base_params.yaml" command="load" />
17
18  <param name="base_global_planner" type="string" value="navfn/NavfnROS" />
19
20  <param name="base_local_planner" value="base_local_planner/TrajectoryPlannerROS"/>
21  <param name="clearing_rotation_allowed" value="true" />
22
23  <!-- remap from="odom" to="/odometry/filtered" /-->
24 </node>
25 </launch>
26

```

```

44 # Optimization
45
46 no_inner_iterations: 5
47 no_outer_iterations: 4
48 optimization_activate: True
49 optimization_verbose: False
50 penalty_epsilon: 0.1
51 weight_max_vel_x: 2
52 weight_max_vel_theta: 1
53 weight_acc_lim_x: 1
54 weight_acc_lim_theta: 1
55 weight_kinematics_nh: 1000
56 weight_kinematics_forward_drive: 1
57 weight_kinematics_turning_radius: 1
58 weight_optimaltime: 1
59 weight_obstacle: 50
60 weight_dynamic_obstacle: 10 # not in use yet
61
62 # Homotopy Class Planner
63
64 enable_homotopy_class_planning: False
65 enable_multithreading: True
66 simple_exploration: False
67 max_number_classes: 2
68 selection_cost_hysteresis: 1.0
69 selection_obst_cost_scale: 1.0
70 selection_alternative_time_cost: False
71 roadmap_graph_no_samples: 15
72 roadmap_graph_area_width: 5
73 h_signature_prescaler: 0.5
74 h_signature_threshold: 0.1
75 obstacle_keypoint_offset: 0.1
76 obstacle_heading_threshold: 0.45
77 visualize_hc_graph: False
78
79 # BaseGlobalPlanner:
80 #   allow_unknown: false
81 #   use_dijkstra: false #Use A* instead
82 #   use_quadratic: true
83 #   use_grid_path: false
84 #   old_navfn_behavior: false

```

- A\*, Time Elastic Band Planner 실행할 때의 설정

1번 파일에 12번째 줄을 주석처리 하고 사용합니다.

3번 파일의 79번째 줄에서 84번째 줄을 주석을 풀어주고 사용합니다.

```

WeMMI_Gazebo > scout_mini_2dnav > launch > move_base_only.launch
1 <launch>
2 <!--node name="map_server" pkg="map_server" type="map_server" args="$(find scout_mini_2dnav)/map/map.yaml"/-->
3
4 <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
5
6   <roscparam file="$(find scout_mini_2dnav)/params/modify_costmap_common_params.yaml" command="load" ns="global_costmap" />
7   <roscparam file="$(find scout_mini_2dnav)/params/modify_costmap_common_params.yaml" command="load" ns="local_costmap" />
8
9   <roscparam file="$(find scout_mini_2dnav)/params/map_nav_params/modify_local_costmap_params.yaml" command="load" />
10  <roscparam file="$(find scout_mini_2dnav)/params/map_nav_params/modify_global_costmap_params.yaml" command="load" />
11
12  <!-- <roscparam file="$(find scout_mini_2dnav)/params/base_local_planner_params.yaml" command="load" /> -->
13  <roscparam file="$(find scout_mini_2dnav)/params/teb_local_planner_params.yaml" command="load" />
14
15
16  <roscparam file="$(find scout_mini_2dnav)/params/move_base_params.yaml" command="load" />
17
18  <param name="base_global_planner" type="string" value="navfn/NavfnROS" />
19
20  <param name="base_local_planner" value="base_local_planner/TrajectoryPlannerROS"/>
21  <param name="clearing_rotation_allowed" value="true" />
22
23  <!--remap from="odom" to="/odometry/filtered" /-->
24 </node>
25 </launch>
26

```

```

46 no_inner_iterations: 5
47 no_outer_iterations: 4
48 optimization_activate: True
49 optimization_verbose: False
50 penalty_epsilon: 0.1
51 weight_max_vel_x: 2
52 weight_max_vel_theta: 1
53 weight_acc_lim_x: 1
54 weight_acc_lim_theta: 1
55 weight_kinematics_nh: 1000
56 weight_kinematics_forward_drive: 1
57 weight_kinematics_turning_radius: 1
58 weight_optimaltime: 1
59 weight_obstacle: 50
60 weight_dynamic_obstacle: 10 # not in use yet
61
62 # Homotopy Class Planner
63
64 enable_homotopy_class_planning: False
65 enable_multithreading: True
66 simple_exploration: False
67 max_number_classes: 2
68 selection_cost_hysteresis: 1.0
69 selection_obst_cost_scale: 1.0
70 selection_alternative_time_cost: False
71 roadmap_graph_no_samples: 15
72 roadmap_graph_area_width: 5
73 h_signature_prescaler: 0.5
74 h_signature_threshold: 0.1
75 obstacle_keypoint_offset: 0.1
76 obstacle_heading_threshold: 0.45
77 visualize_hc_graph: False
78
79 BaseGlobalPlanner:
80   allow_unknown: false
81   use_dijkstra: false #Use A* instead
82   use_quadratic: true
83   use_grid_path: false
84   old_navfn_behavior: false

```



# Navigation Parameter 수정

- 이번 단원에서는 Global Planner는 상관없지만, Local Planner를 DWA일 때만 해당되는 내용입니다.
- Navigation package는 Localization을 위한 AMCL, Path planning 및 following을 위한 move\_base 두개의 패키지로 이루어져 있습니다.
- navigation.launch

```
<launch>
  <!-- map file -->
  <arg name="map_file" default="$(find wego)/maps/test_map.yaml"/>

  <!-- map server -->
  <node name="map_server" pkg="map_server" type="map_server" args="$(arg map_file)" />

  <!-- amcl -->
  <include file="$(find scout_mini_2dnav)/launch/amcl.launch" />

  <!-- move base -->
  <include file="$(find scout_mini_2dnav)/launch/move_base_only.launch" />

  <!-- run rviz -->
  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find wego)/rviz/navigation.rviz" />

  <!-- run tf2 -->
  <node pkg="tf2_ros" type="static_transform_publisher" name="velodyne_broadcast" args="0 0 0 0 0 0 base_link velodyne" />
</launch>
```

- AMCL

- amcl.launch

- min\_particles

- 최소 파티클

- max\_particles

- 최대 파티클

- 파티클이 많을수록 파티클필터의 성능이 상승

```
<param name="use_map_topic" value="$(arg use_map_topic)"/>
<!-- Publish scans from best pose at a max of 10 Hz -->
<param name="odom_model_type" value="diff"/>
<param name="odom_alpha5" value="0.1"/>
<param name="gui_publish_rate" value="10.0"/>
<param name="laser_max_beams" value="720"/>
<param name="laser_min_range" value="0.1"/>
<param name="laser_max_range" value="30.0"/>
<param name="min_particles" value="500"/>
<param name="max_particles" value="2000"/>
<!-- Maximum error between the true distribution and the estimated distribution. -->
<param name="kld_err" value="0.05"/>
<param name="kld_z" value="0.99"/>
<param name="odom_alpha1" value="0.2"/>
<param name="odom_alpha2" value="0.2"/>
<!-- translation std dev, m -->
<param name="odom_alpha3" value="0.2"/>
<param name="odom_alpha4" value="0.2"/>
<param name="laser_z_hit" value="0.5"/>
<param name="laser_z_short" value="0.05"/>
<param name="laser_z_max" value="0.05"/>
<param name="laser_z_rand" value="0.5"/>
<param name="laser_sigma_hit" value="0.2"/>
<param name="laser_lambda_short" value="0.1"/>
<param name="laser_model_type" value="likelihood_field"/>
```

- AMCL

- amcl.launch

- update\_min\_d

- 업데이트 최소값
- 높아질수록 잦은 업데이트
- 높은성능, 높은연산량
- 좀더 자세한 파라미터 설명은 다음 링크 참조
- <http://wiki.ros.org/amcl>

```
<!-- Maximum distance to do obstacle inflation on map, for use in likelihood_field model. -->
<param name="laser_likelihood_max_dist" value="2.0"/>
<!-- Translational movement required before performing a filter update. -->
<param name="update_min_d" value="0.05" />
<!-- Rotational movement required before performing a filter update. -->
<param name="update_min_a" value="0.05"/>
<param name="odom_frame_id" value="odom"/>
<param name="base_frame_id" value="base_link"/>
<param name="global_frame_id" value="map"/>
<!-- Number of filter updates required before resampling. -->
<param name="resample_interval" value="1"/>
<!-- Increase tolerance because the computer can get quite busy -->
<param name="transform_tolerance" value="1.0"/>
<!-- Exponential decay rate for the slow average weight filter, used in deciding when to recover by adding random poses. A good value might be 0.001. -->
<param name="recovery_alpha_slow" value="0.0"/>
<!-- Exponential decay rate for the fast average weight filter, used in deciding when to recover by adding random poses. A good value might be 0.1. -->
<param name="recovery_alpha_fast" value="0.1"/>
<!-- Initial pose mean -->
<param name="initial_pose_x" value="0.0" />
<param name="initial_pose_y" value="0.0" />
<param name="initial_pose_a" value="0.0" />
<!-- When set to true, AMCL will subscribe to the map topic rather than making a service call to receive its map.-->
<param name="receive_map_topic" value="false"/>
<!-- When set to true, AMCL will only use the first map it subscribes to, rather than updating each time a new one is received. -->
<param name="first_map_only" value="false"/>
<remap from="/scan" to="/velodyne_scan"/>
```

- Move\_base
- move\_base\_only.launch
- global map과 local map의 설정파일

```
<node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">

  <rosparam file="$(find scout_mini_2dnav)/params/costmap_common_params.yaml" command="load" ns="global_costmap" />
  <rosparam file="$(find scout_mini_2dnav)/params/costmap_common_params.yaml" command="load" ns="local_costmap" />

  <rosparam file="$(find scout_mini_2dnav)/params/map_nav_params/local_costmap_params.yaml" command="load" />
  <rosparam file="$(find scout_mini_2dnav)/params/map_nav_params/global_costmap_params.yaml" command="load" />

  <rosparam file="$(find scout_mini_2dnav)/params/base_local_planner_params.yaml" command="load" />
  <rosparam file="$(find scout_mini_2dnav)/params/move_base_params.yaml" command="load" />

  <param name="base_global_planner" type="string" value="navfn/NavfnROS" />
  <param name="base_local_planner" value="base_local_planner/TrajectoryPlannerROS"/>

  <remap from="odom" to="/odometry/filtered" />
</node>
```

•costmap\_common\_param.yaml

•footprint\_padding

- 장애물과 로봇의 여유분
- 0.01->1cm
- 늘린다면 좀 더 안정성있는 경로를 생성
- 하지만 통로를 지나기 어려워짐

```
map_type: costmap
origin_z: 0.0
z_resolution: 1
z_voxels: 2

obstacle_range: 2.5
raytrace_range: 3.0

publish_voxel_map: false
transform_tolerance: 0.5
meter_scoring: true

footprint: [[-0.4, -0.3], [-0.4, 0.3], [0.4, 0.3], [0.4, -0.3]]
footprint_padding: 0.01

plugins:
- {name: obstacles_layer, type: "costmap_2d::ObstacleLayer"}
- {name: inflater_layer, type: "costmap_2d::InflationLayer"}

obstacles_layer:
  observation_sources: scan
  scan: {sensor_frame: laser, data_type: LaserScan, topic: /velodyne_scan, marking:

inflater_layer:
  inflation_radius: 0.25
```

•base\_local\_planner\_param.yaml

•표시된 param

- 직선속도, 회전속도
- 직선가속도, 회전가속도
- 4개의 파라미터를 변경 가능
- 자율 주행할때 최대, 최소 속도 및 가속도 설정

```
▼ TrajectoryPlannerROS:
# Robot Configuration Parameters
acc_lim_x: 8.0
acc_lim_theta: 3.0

max_vel_x: 0.5
min_vel_x: 0.1

max_vel_theta: 0.3
min_vel_theta: -0.5
min_in_place_vel_theta: 0.314

holonomic_robot: false
escape_vel: -0.3

# Goal Tolerance Parameters
yaw_goal_tolerance: 0.157
xy_goal_tolerance: 0.25
latch_xy_goal_tolerance: false

# Forward Simulation Parameters
sim_time: 2.0
sim_granularity: 0.02
angular_sim_granularity: 0.02
vx_samples: 6
vtheta_samples: 20
controller_frequency: 20.0

# Trajectory scoring parameters
meter_scoring: true # Whether the gdist
occdist_scale: 0.3 #The weighting for
pdist_scale: 0.75 # The weighting
gdist_scale: 0.8 # The weighting fo

heading_lookahead: 0.325 #How far to l
heading_scoring: false #Whether to sco
heading_scoring_timestep: 0.8 #How fa
dwa: true #Whether to use the Dynamic W
simple_attractor: false
publish_cost_grid_pc: true

#Oscillation Prevention Parameters
oscillation_reset_dist: 0.05 #How far t
escape_reset_dist: 0.1
escape_reset_theta: 0.1
```

- base\_local\_planner\_param.yaml

- 표시된 param

- yaw, xy의 차이가 설정된 값 이하라면 도착으로 간주
- 줄인다면, 좀 더 목적지에 정확하게 도착

- sim\_time

- dwa(local\_planner)의 속도를 조정
- 높일수록 경로를 자주 생성

- 

move\_base의 자세한 파라미터는 아래 링크 참조

- [http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base)

```
~ TrajectoryPlannerROS:

# Robot Configuration Parameters
acc_lim_x: 8.0
acc_lim_theta: 3.0

max_vel_x: 0.5
min_vel_x: 0.1

max_vel_theta: 0.3
min_vel_theta: -0.5
min_in_place_vel_theta: 0.314

holonomic_robot: false
escape_vel: -0.3

# Goal Tolerance Parameters
yaw_goal_tolerance: 0.157
xy_goal_tolerance: 0.25
latch_xy_goal_tolerance: false

# Forward Simulation Parameters
sim_time: 2.0
sim_granularity: 0.02
angular_sim_granularity: 0.02
vx_samples: 6
vtheta_samples: 20
controller_frequency: 20.0

# Trajectory scoring parameters
meter_scoring: true # Whether the gdist
occdist_scale: 0.3 # The weighting for
pdist_scale: 0.75 # The weighting
gdist_scale: 0.8 # The weighting fo

heading_lookahead: 0.325 #How far to l
heading_scoring: false #Whether to sco
heading_scoring_timestep: 0.8 #How fa
dwa: true #Whether to use the Dynamic W
simple_attractor: false
publish_cost_grid_pc: true

#Oscillation Prevention Parameters
oscillation_reset_dist: 0.05 #How far t
escape_reset_dist: 0.1
escape_reset_theta: 0.1
```





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](mailto:go.sales@wego-robotics.com)

기술문의: [go.support@wego-robotics.com](mailto:go.support@wego-robotics.com)