ROS Localization

WeGo & Logistics Robot

WeGo

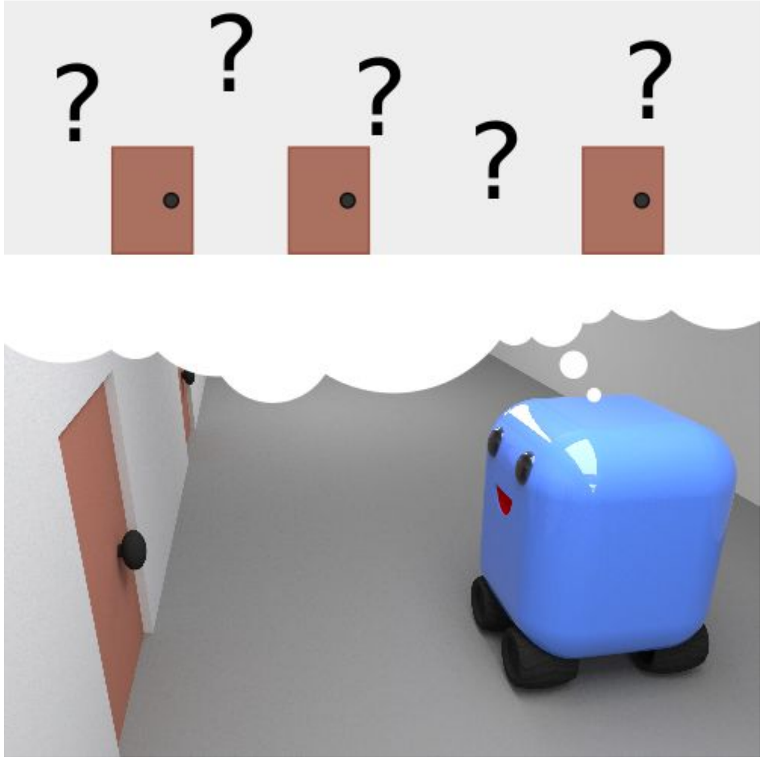
목 차

1. Localization Introduction
2. Localization Package
3. Localization Demo

01 Localization Introduction

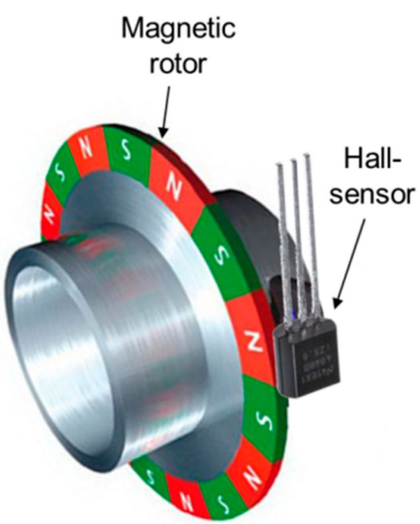
01 Localization Introduction

- 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(Position Estimation)이라고도 합니다.

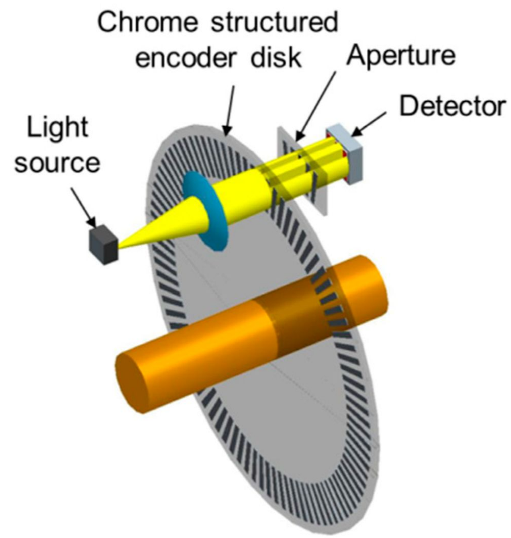


01 Localization Introduction

- 로봇의 형태 및 자유도에 따라서 추정해야하는 차원 수가 달라집니다.
- 일반적인 2D 환경에서 이동하는 Mobile Robot의 경우는 지도 상에서의 위치(x, y) 및 헤딩 각(θ)의 세 가지 정보를 Pose $x_t = (x, y, \theta)^T$ 와 같이 표시한다.
- 사용하는 센서로는 내부의 이동을 측정하는 Odometer (주행 기록계) 및 LiDAR 센서 데이터 두 가지를 사용한다.



(a)



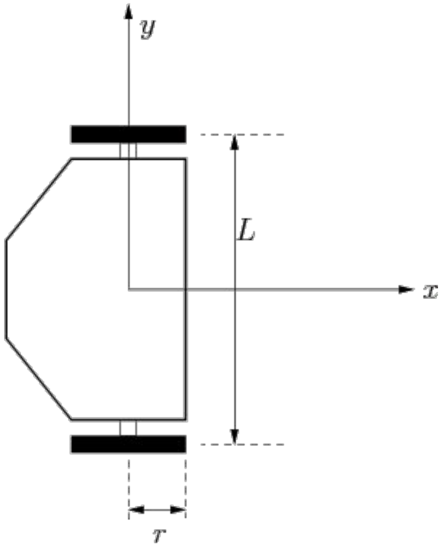
(b)

- **Differential Drive Model**

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

$$\dot{x} = vcos(\phi)$$
$$\dot{y} = vsin(\phi)$$
$$\dot{\phi} = \omega$$


(a)



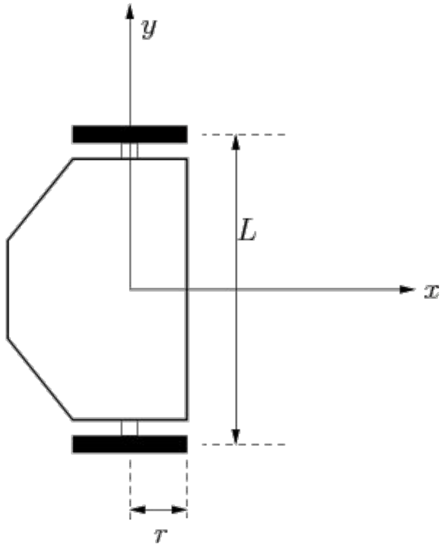
(b)

01 Localization Introduction

- **Differential Drive Model**

$$v = \frac{R}{2}(v_r + v_l)$$
$$\omega = \frac{R}{L}(v_r - v_l)$$

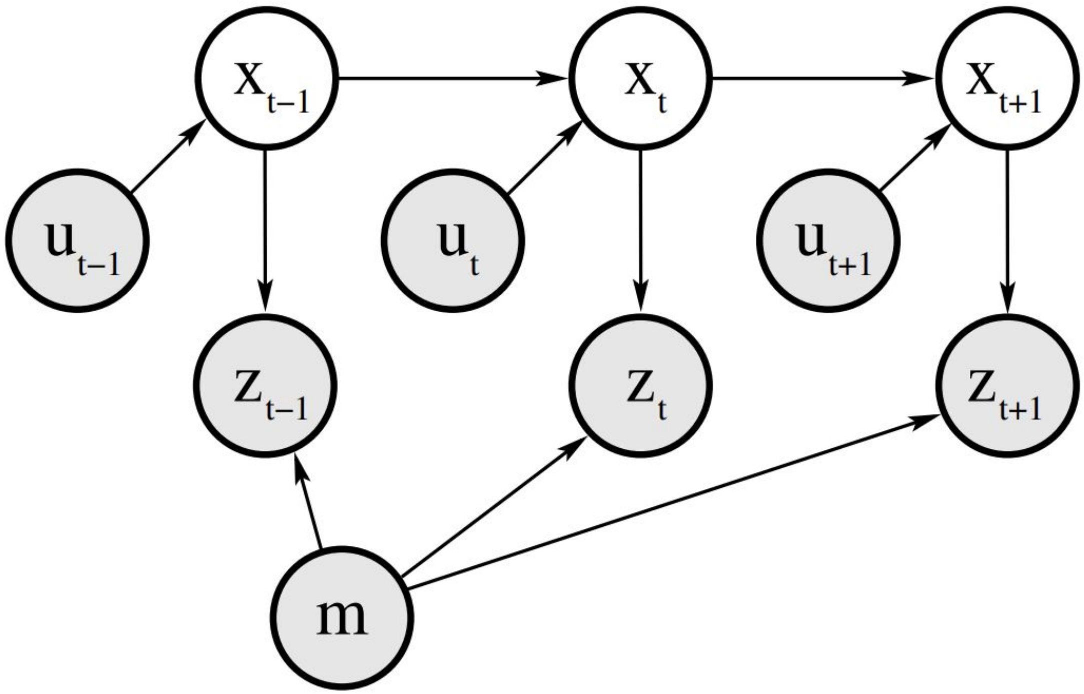
(a)



(b)

01 Localization Introduction

- Mobile Robot Localization Graphical Model
  - Robot State x_{t-1}
  - Map Data m
  - Measurement (LiDAR) z_{t-1}
  - Control Data (Odometry) u_{t-1}



WeGo

01 Localization Introduction

- Position Tracking – 로봇의 초기 위치를 알고 있고, 이를 기반으로 로봇의 움직임을 추적하여, 로봇의 위치를 추정하는 방법 (Local Problem)
- Global Localization – 로봇의 초기 위치를 모르는 상태이며, 지도 상의 한 점에서 시작을 하지만 위치는 알 수 없는 상태(Global Problem)

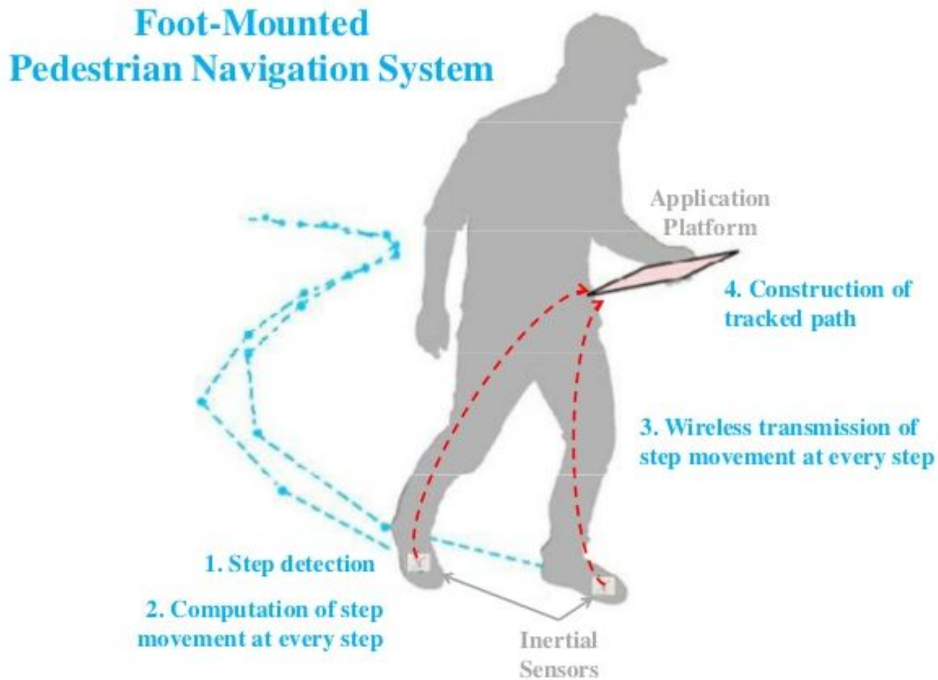
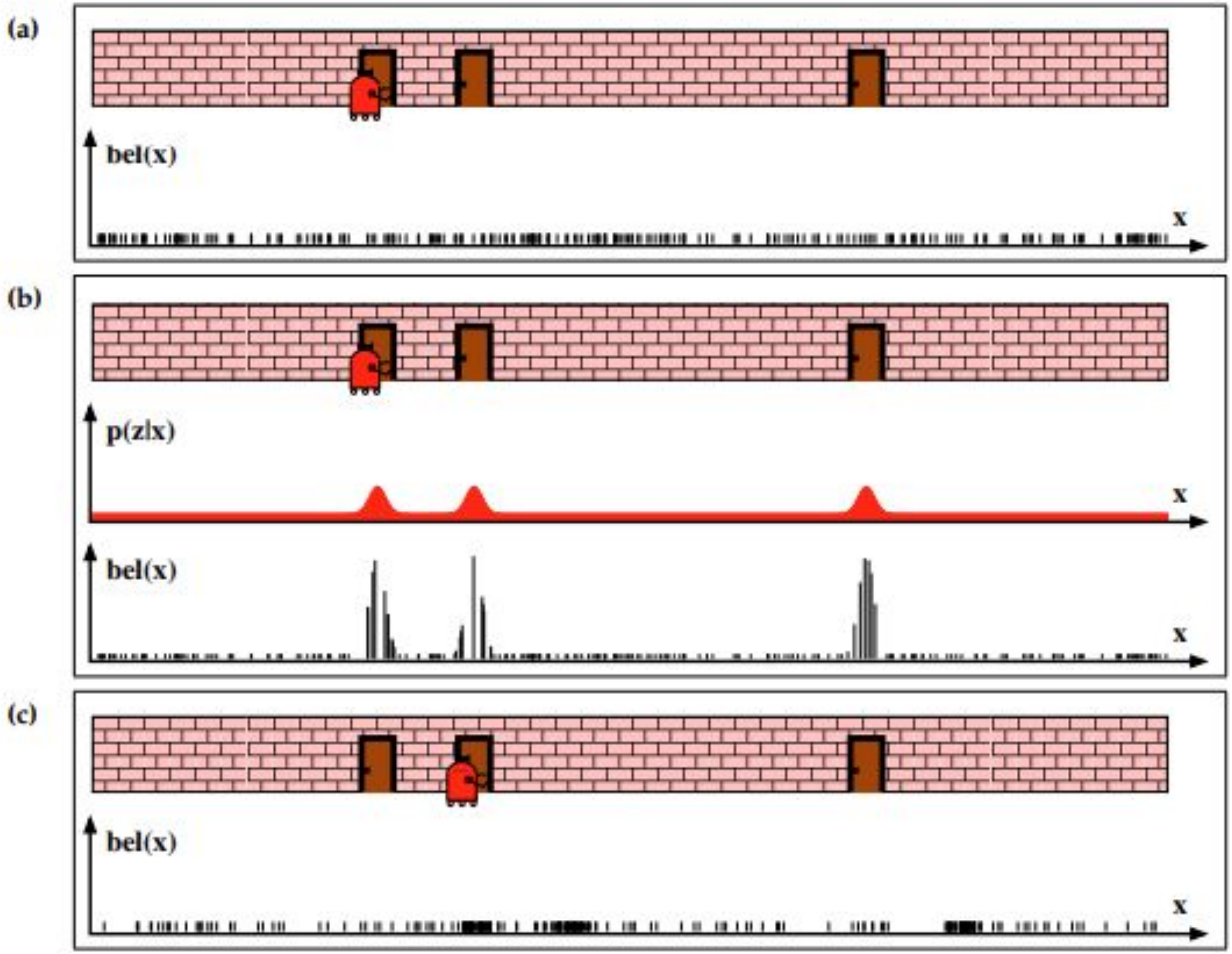


Image source: John-Olof Nilsson, Amit K Gupta, Peter Handel, "Foot mounted inertial navigation made easy", Proc Indoor Positioning & Indoor Navigation (IPIN), 2014

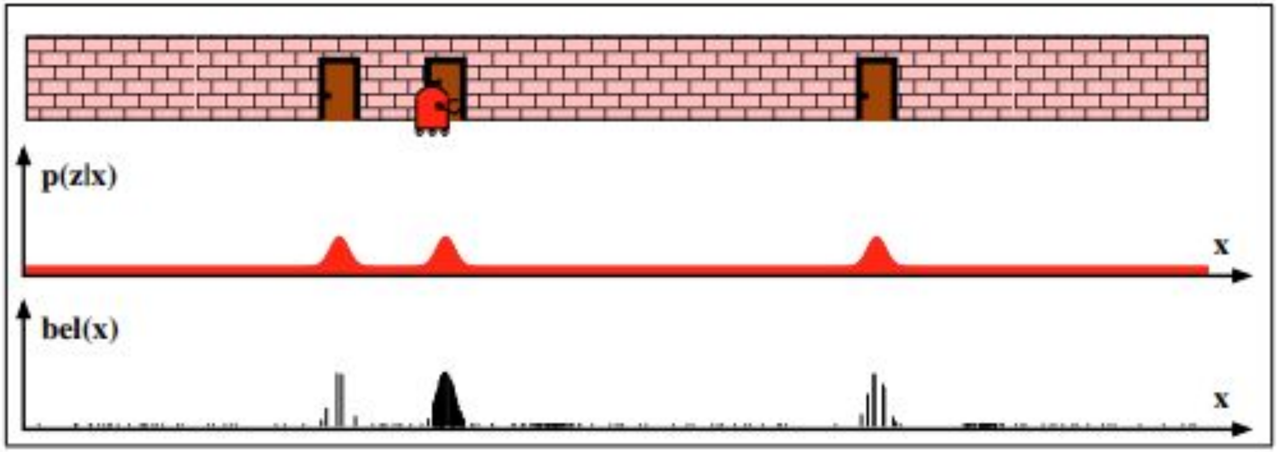
01 Localization Introduction

- 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 수렴시킨다.

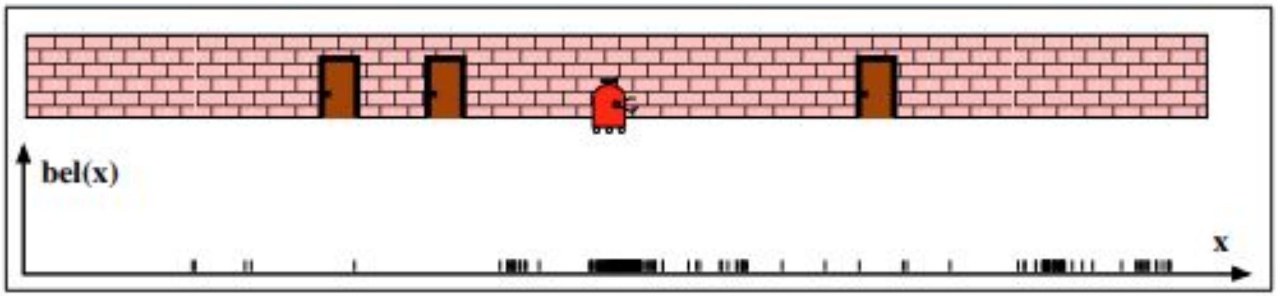
01 Localization Introduction



(d)



(e)



02 Localization Package

02 Localization Package

- amcl - adaptive monte carlo localization
- Published Topics
  - amcl_pose (geometry_msgs/PoseWithCovarianceStamped) - Robot's estimated pose in the map, with covariance.
  - particlecloud (geometry_msgs/PoseArray) - The set of pose estimates being maintained by the filter.
  - tf (tf/tfMessage) - Publishes the transform from odom (~odom_frame_id parameter) to map.
- Subscribed Topics
  - scan (sensor_msgs/LaserScan) - Lidar Laser scan data.
  - tf (tf/tfMessage) - Transforms.
  - initialpose (geometry_msgs/PoseWithCovarianceStamped) - Mean and covariance with which to (re-)initialize the PF.
  - map (nav_msgs/OccupancyGrid) - When the use_map_topic parameter is set, AMCL subscribes to map topic to retrieve the map used for laser-based localization

- Parameters(Default) - Overall filter parameters
  - ~min_particles (100) - Minimum allowed number of particles.
  - ~max_particles (5000) - Maximum allowed number of particles.
  - ~kld_err (0.01) - Maximum error between the true distribution and the estimated distribution.
  - ~kld_z (0.99) - Upper standard normal quantile for (1 - p), where p is the probability that the error on the estimated distrubition will be less than kld_err.
  - ~update_min_d (0.2 meters) - Translational movement required before performing a filter update.
  - ~update_min_a (π/6.0 radians) - Rotational movement required before performing a filter update.
  - ~resample_interval (2) - Number of filter updates required before resampling.
  - ~transform_tolerance (0.1 seconds) - Time with which to post-date the transform that is published, to indicate that this transform is valid into the future.
  - ~recovery_alpha_slow (0.0 (disabled)) - 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.
  - ~recovery_alpha_fast (0.0 (disabled)) - 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.
  - ~initial_pose_x (0.0 meters) - Initial pose mean (x), used to initialize filter with Gaussian distribution.
  - ~initial_pose_y (0.0 meters) - Initial pose mean (y), used to initialize filter with Gaussian distribution.
  - ~initial_pose_a (0.0 radians) - Initial pose mean (yaw), used to initialize filter with Gaussian distribution.

02 Localization Package

- Parameters(Default) – Overall filter parameters
  - ~initial_cov_xx (0.5*0.5 meters) – Initial pose covariance (x*x), used to initialize filter with Gaussian distribution.
  - ~initial_cov_yy (0.5*0.5 meters) – Initial pose covariance (y*y), used to initialize filter with Gaussian distribution.
  - ~initial_cov_aa ((π/12)*(π/12) radian) – Initial pose covariance (yaw*yaw), used to initialize filter with Gaussian distribution.
  - ~gui_publish_rate (double, default: -1.0 Hz) – Maximum rate (Hz) at which scans and paths are published for visualization, -1.0 to disable.
  - ~save_pose_rate (double, default: 0.5 Hz) – Maximum rate (Hz) at which to store the last estimated pose and covariance to the parameter server, in the variables ~initial_pose_* and ~initial_cov_*. This saved pose will be used on subsequent runs to initialize the filter. -1.0 to disable.
  - ~use_map_topic (bool, default: false) – When set to true, AMCL will subscribe to the map topic rather than making a service call to receive its map. New in navigation 1.4.2
  - ~first_map_only (bool, default: 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. New in navigation 1.4.2
  - ~selective_resampling (bool, default: false) – When set to true, will reduce the resampling rate when not needed and help avoid particle deprivation. The resampling will only happen if the effective number of particles (N_eff = 1/(sum(k_i^2))) is lower than half the current number of particles. Reference: Grisetti, Giorgio, Cyrill Stachniss, and Wolfram Burgard. "Improved techniques for grid mapping with rao-blackwellized particle filters." IEEE transactions on Robotics 23.1 (2007): 34.

02 Localization Package

- Parameters(Default) - Laser model parameters
  - ~laser_min_range (-1.0) - Minimum scan range to be considered. -1.0 will cause the laser's reported minimum range to be used.
  - ~laser_max_range (-1.0) - Maximum scan range to be considered; -1.0 will cause the laser's reported maximum range to be used.
  - ~laser_max_beams (30) - How many evenly-spaced beams in each scan to be used when updating the filter.
  - ~laser_z_hit (0.95) - Mixture weight for the z_hit part of the model.
  - ~laser_z_short (0.1) - Mixture weight for the z_short part of the model.
  - ~laser_z_max (0.05) - Mixture weight for the z_max part of the model.
  - ~laser_z_rand (0.05) - Mixture weight for the z_rand part of the model.
  - ~laser_sigma_hit (0.2 meters) - Standard deviation for Gaussian model used in z_hit part of the model.
  - ~laser_lambda_short (0.1) - Exponential decay parameter for z_short part of model.
  - ~laser_likelihood_max_dist (2.0 meters) - Maximum distance to do obstacle inflation on map, for use in likelihood_field model.
  - ~laser_model_type ("likelihood_field") - Which model to use, either beam, likelihood_field, or likelihood_field_prob (same as likelihood_field but incorporates the beamskip feature, if enabled).

02 Localization Package

- Parameters(Default) – Odometry model parameters
  - ~odom_model_type ("diff") – Which model to use, either "diff", "omni", "diff-corrected" or "omni-corrected".
  - ~odom_alpha1 (0.2) – Specifies the expected noise in odometry's rotation estimate from rotational component of robot's motion.
  - ~odom_alpha2 (0.2) – Specifies the expected noise in odometry's rotation estimate from translational component of robot's motion.
  - ~odom_alpha3 (0.2) – Specifies the expected noise in odometry's translation estimate from translational component of robot's motion.
  - ~odom_alpha4 (0.2) – Specifies the expected noise in odometry's translation estimate from the rotational component of robot's motion.
  - ~odom_alpha5 (0.2) – Translation-related noise parameter (only used if model is "omni").
  - ~odom_frame_id ("odom") – Which frame to use for odometry.
  - ~base_frame_id ("base_link") – Which frame to use for the robot base
  - ~global_frame_id ("map") – The name of the coordinate frame published by the localization system
  - ~tf_broadcast (true) – Set this to false to prevent amcl from publishing the transform between the global frame and the odometry frame.

02 Localization Package

- particle filter - basic particle filter localization
- Published Topics
  - /pf/pose/odom (nav_msgs/Odometry) - Robot's estimated pose in the map. with covariance.
  - /pf/viz/fake_scan (sensor_msgs/LaserScan) - Fake Scan made from Map
  - /pf/viz/inferred_pose (geometry_msgs/PoseStamped) - Robot's estimated pose in the map
  - /pf/viz/particles (geometry_msgs/PoseArray) - Particle's position
- Subscribed Topics
  - /intialpose (geometry_msgs/PoseWithCovarianceStamped) - Recover Position
  - /scan (sensor_msgs/LaserScan) - 2D LiDAR Range Data
  - /vesc/odom (nav_msgs/Odometry) - Odometry of Vehicle

WeGo

02 Localization Package

- Parameters
  - ~angle_step
  - ~fine_timing
  - ~max_particles
  - ~max_range
  - ~max_viz_particles
  - ~motion_dispersion_theta
  - ~motion_dispersion_x
  - ~motion_dispersion_y
  - ~odometry_topic
  - ~publish_odom
  - ~range_method
  - ~rangelib_variant
  - ~scan_topic
  - ~sigma_hit
  - ~squash_factor
  - ~theta_discretization
  - ~viz
  - ~z_hit
  - ~z_max
  - ~z_rand
  - ~z_short

03 Localization Demo

WeGo

- amcl – adaptive monte carlo localization
  - Particle Filter 기반의 Localization 방법
  - Input : 플랫폼의 이동에 해당하는 Odometry, 2D LiDAR 거리 데이터
  - GMapping 또는 Hector SLAM을 통해서 생성한 Map 상의 로봇의 위치를 추정
  - Odometry의 정확도에 많이 의존하는 편이며, LiDAR를 이용하는 모든 알고리즘과 유사하게 직선 주행에서는 괜찮으나, 회전에서 많이 취약한 편
  - 수동으로 위치를 보정하는 기능 및 Global Localization 두 가지 방법을 모두 제공

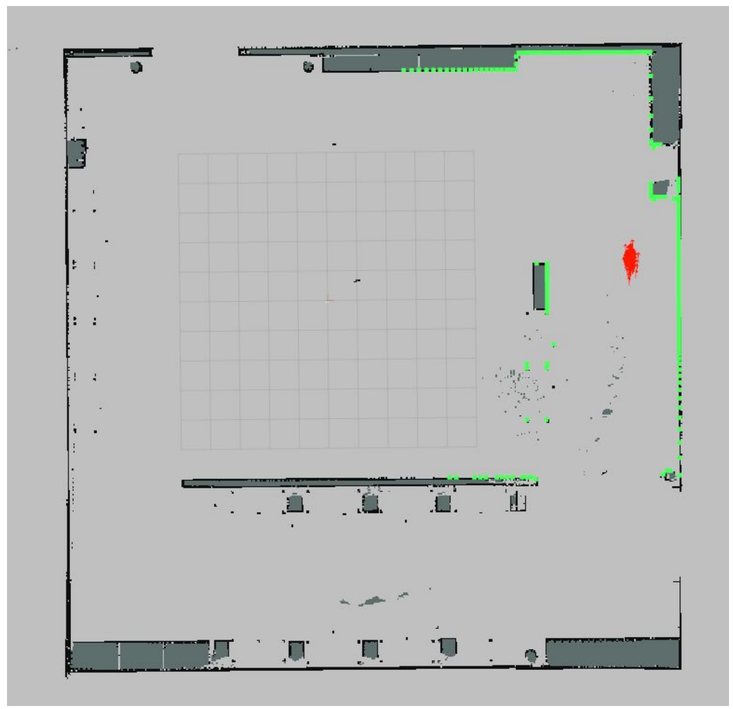
03 Localization Demo

- amcl - adaptive monte carlo localization
- AMCL ROS Wiki - http://wiki.ros.org/amcl
- AMCL Github - https://github.com/ros-planning/navigation

- Related Papers
  - KLD-Sampling: Adaptive Particle Filters - https://proceedings.neurips.cc/paper/2001/file/c5b2cebf15b205503560c4e8e6d1ea78-Paper.pdf
  - https://roboticsknowledgebase.com/wiki/state-estimation/adaptive-monte-carlo-localization/

03 Localization Demo

- amcl - adaptive monte carlo localization
  - $ roslaunch wecar teleop.launch
  - $ roscd wecar/map
  - $ rosrun map_server map_server "FILENAME.YAML"
  - $ roslaunch amcl amcl_diff.launch



03 Localization Demo

- particle filter – basic particle filter package
  - Particle Filter 기반의 Localization 방법
  - Input : 플랫폼의 이동에 해당하는 Odometry, 2D LiDAR 거리 데이터
  - GMapping 또는 Hector SLAM을 통해서 생성한 Map 상의 로봇의 위치를 추정
  - AMCL에 비해 수정 가능성 및 자유도가 높은편이지만, 기본 성능 자체가 많이 낮은편

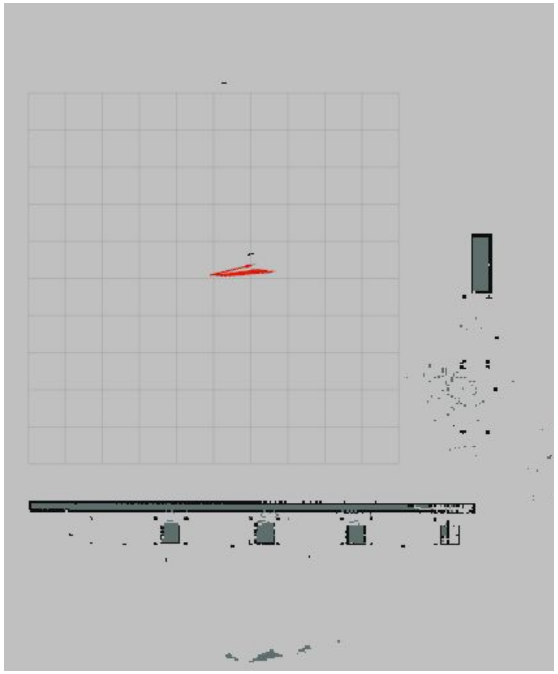
03 Localization Demo

- particle filter - basic particle filter localization
- particle filter Github - https://github.com/mit-racecar/particle_filter

- Related Papers
  - Lab5-Localization&Mapping - https://github.com/mit-racecar/particle_filter/blob/master/docs/Lab5.pdf
  - RangeLibC Usage and Information - https://github.com/mit-racecar/particle_filter/blob/master/docs/RangeLibcUsageandInformation.pdf

03 Localization Demo

- particle filter – basic particle filter localization
  - $ roslaunch wecar teleop.launch
  - $ roscd wecar/map
  - $ rosrun map_server map_server "FILENAME.YAML"
  - $ roslaunch particle_filter localize.launch



WeGo Robotics

Tel. 031 – 229 – 3553

Fax. 031 – 229 – 3554

제품 문의: go.sales@wego-robotics.com

기술 문의: go.support@wego-robotics.com