Preliminary progress on application of SLAM algorithm

Offline application on Ground Vehicle

Agraj Jain
Hamed Ali Yaghini Bonabi

Control/Robotics Research Laboratory

April 25, 2015

- Localization:
  - Finding the robot's position in the environment.
- Mapping:
  - Creating a map of the environment.
- Goal:
  - Making the robots autonomous.
  - Creating map of the environment for different applications.

Two general approaches:

- Traditional approach: Metrical SLAM.
- New approaches : Semantic SLAM.

Motion Model

- Motion model is the relation between the previous state of the robot, control inputs and the robot's current state.

In general:

$$X_t = g(X_{t-1}, U_t)$$

where, X_t is the robot's current state.X_{t-1} is the robot's previous state, and U_t is the control input.

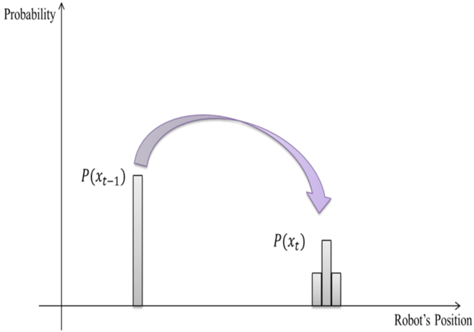
Simplest Localization Case: If we know the robots initial position $\longrightarrow$ use the motion model for finding robot's position.! However, uncertainty in parameters, measurements, control. $\longrightarrow$ We cannot.

General approach:

- Motion model for Prediction.
- Landmarks for correction.

Bayes Filter

- Prediction by using motion model.
- Correction by landmark detection.



Landmark detection

- Spark method: uses the step change in beams distance for finding landmarks.
- RANSAC: is used for finding walls by fitting a line to the points that are along the wall.

Bayes filter's problems:

- Requires lots of calculation.
- It is discrete $\longrightarrow$ The required memory increases by time.

Solutions:

- Selecting larger cells. $\longrightarrow$ accuracy decreases
- Kalman filter.

- Is obtained from the Bayes filter. By assuming that robot's position has a normal distribution.
- Only requires calculation of mean and covariance.

Motion model

If r $\neq$ l:

$$\alpha = \frac{r - l}{w} \qquad R = \frac{l}{\alpha}$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \left(R + \frac{w}{2}\right)(\sin(\theta + \alpha) - \sin(\theta)) \\ \left(R + \frac{w}{2}\right)(-\cos(\theta + \alpha) - \cos(\theta)) \\ \alpha \end{bmatrix}$$

If r $=$ l:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} l.\cos(\theta) \\ l.\sin(\theta) \\ 0 \end{bmatrix}$$

In general: $x' = g(x, u)$ Where:

$$x = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad u = \begin{bmatrix} l \\ r \end{bmatrix}$$

Predicting the measurement:

$$\bar{z}_t = h(x_m, y_m, \theta)$$

$$\bar{z}_t = \begin{bmatrix} r \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{(x_l - x_m)^2 + (y_l - y_m)^2} \\ \arctan\left(\frac{y_l - y_m}{x_l - x_m}\right) - \theta \end{bmatrix}$$

Where, x_m, y_m are the co-ordinates of each landmark already on the map.

Extended Kalman filter

EKF's problems:

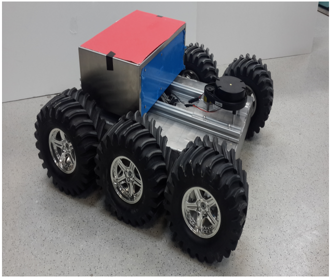
- What will happen if we dont know the robot's initial position ? What if the positions distribution is not normal?

$\longrightarrow$ Particle filter The main idea: using points and their density as a tool for representing distribution

Simultaneous localization and mapping

- Put the world coordinates origin in the robot's starting point.
- Start observing the environment.
- While moving, add the landmarks as detected to the systems state, with uncertainty.
- Use prediction and correction methods of localization to finding the correct location of both robot and landmarks.

$$\mu = \begin{bmatrix} x \\ y \\ \theta \\ x_1 \\ y_1 \\ x_2 \\ y_2 \\ . \\ . \\ . \end{bmatrix}$$



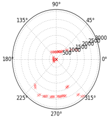
- Odometry:
  - 2 Encoders sampled at 1000 Hz
  - Velocity logged at a rate of 200 Hz
- Piccolo Laser Distance Sensor:
  - 6 m range
  - 5 Hz Rotation Speed
  - 1deg Angular Resolution

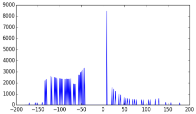
Implementation

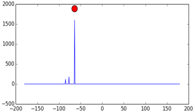
Enviornment

Landmark Extraction methods

Spike Landmark Extraction







- Derivative Based Detection
- Very susceptible to noise
- Dependent on environment and shape of landmark

Traditional Algorithm:[1]

1. Select a random subset of the original data. Call this subset the hypothetical inliers.
2. A model is fitted to the set of hypothetical inliers.
3. All other data are then tested against the fitted model. Those points that fit the estimated model well, according to some model-specific loss function, are considered as part of the consensus set.
4. The estimated model is reasonably good if sufficiently many points have been classified as part of the consensus set.
5. Afterwards, the model may be improved by reestimating it using all members of the consensus set.

[1] From wikipedia

Landmark Extraction methods

RANSAC

Algorithm used:

while

- there are still unassociated laser readings,
- and the number of readings is larger than the consensus,
- and we have done less than N trials.

do

- Select a random laser data reading.
- Randomly sample S data readings within D degrees of this laser data reading
- Using these S samples and the original reading calculate a least squares best fit line.
- Determine how many laser data readings lie within X distance from this best fit line
- If the number of laser data readings on the line is above some consensus C do the following:
  - calculate new least squares best fit line based on all the laser readings determined to lie on the old best fit line.
  - Add this best fit line to the lines we have extracted.
  - Remove the number of readings lying on the line from the total set of unassociated readings.

end while

Landmark Extraction methods

RANSAC

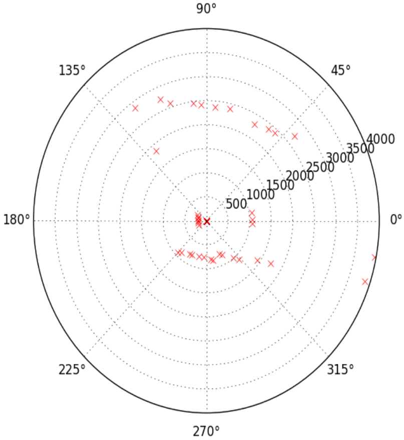


Figure : The LIDAR Scan

Landmark Extraction methods

RANSAC

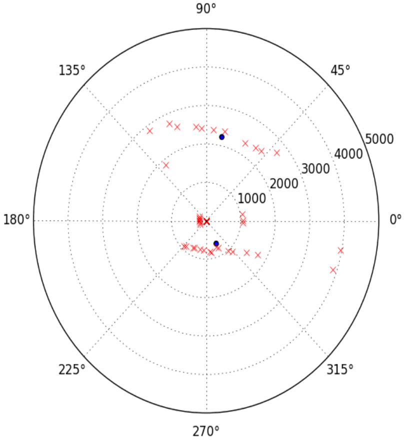


Figure : Detected walls

Landmark Extraction methods

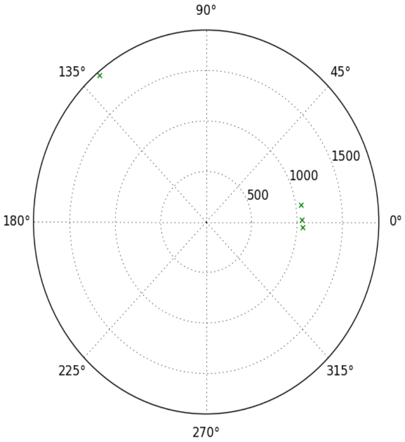


Figure : Remaining points

Landmark Extraction methods

Mean Shift Clustering

- Centroid based algorithm
- Works by updating candidates for centroids to be the mean of the points within a given region

Given a candidate centroid x_i for iteration t, the candidate is updated according to the following equation:

$$x_i^{t+1} = x_i^t + m(x_i^t)$$

m is the mean shift vector that is computed for each centroid that points towards a region of the maximum increase in the density of points.

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}$$

Landmark Extraction methods

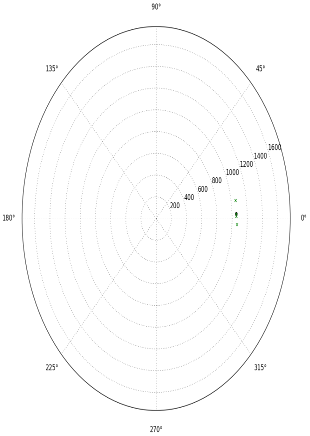
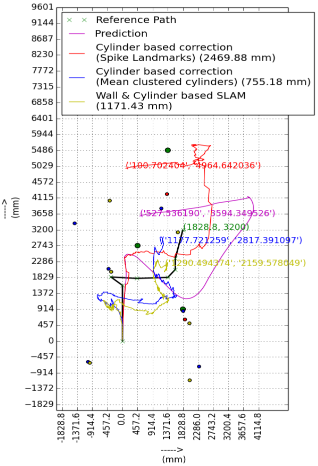


Figure : Detected Cylinders

Results


Reference Path
Prediction
Cylinder based correction (Spike Landmarks) (2469.88 mm)
Cylinder based correction (Mean clustered cylinders) (755.18 mm)
Wall & Cylinder based SLAM (1171.43 mm)
('100.702404', '4964.642036')
('527.536190', '3594.349526')
(1828.8, 3200)
('1177.721259', '2817.391097')
('1290.494374', '2159.578649')
9601
9144
8687
8230
7772
7315
6858
6401
5944
5486
5029
4572
4115
3658
3200
2743
2286
1829
1372
914
457
0
−457
−914
−1372
−1829
-1828.8
-1371.6
-914.4
-457.2
0.0
457.2
914.4
1371.6
1828.8
2286.0
2743.2
3200.4
3657.6
4114.8
----->
(mm)

Future Landmark Extraction methods

Feature detection

Figure : Typical on-board camera frame

Semantic SLAM

Difference with metrical SLAM

- Metrical SLAM: looks at geographical position of the landmarks $==$ sees them as obstacle, does not interpret sensor information other than the geometric level.
- Semantic SLAM: uses other features of environment for producing a meaningful representation.

Semantic SLAM

Applications:

- Possibility of information sharing between robots, and robots and humans $\longrightarrow$
  - Human robot collaboration.
  - Collaboration of robots with each other.

The richer the environment model, and the higher its semantic level $\longrightarrow$ the more useful it becomes for a robot in order to perform autonomous tasks.

Semantic SLAM

Examples

Meaningful representation $\longrightarrow$ distinguishing navigable and non navigable areas.

Collaboration between robots.

- Aerial Robots
- Ground Robots

Thank you