Robot Navigation: Lecture 6
Lecture Notes

Victor Risager

September 26, 2023

1 Recap

Behavior based navigation. Simple, similar to the Bug algorithms.
Map-based navigation (or model-based) it is more which goes through the steps betweenperception and actuation.

Markov assumption: Determine the state, based on only the previous state, and nothing else.

Bayes filter: Recursive, which updates the current state based on the previous state, and improves the estimation of the robot localization.

Assumptions:

- The system is linear.
- The robot motion model and measurement model is affected by white gaussian noise.

1.0.1 Kalman filter

Proporties of KF's: We know the Kalman filter is linear, and the EKF is used for nonlinear systems.
Optimality of EKF is not gurarанteed.
Steps of kalman filtering:

1. Prediction based on previous estimate and odometry
2. Observation with on-board sensors
3. Predicted observation based on prediction and map
4. Matching of observation and map
5. Estimation $\rightarrow$ position update (Posteriori position)

Data association is done with the innovation filter. The innovation is covariance is the difference between the measured state, and the estimated state.
Kalman filter can solve global localisation and kidnapped robot problem given using Multi-Hyposthesis:

- Belief is represented by multiple hypothesis
- Each hypothesis is tracked by a Kalman filter.

2 SLAM

Based on 3 different algorithms:

- EKF
- Particle filter SLAM

Both the map and the path is affected by noise.

2.1 Map building

1. It is important to maintain the map, if changes occur.
2. Representing and propagating Uncertainty.
3.
4.
5.
6.

At $t = 0$ we are certain where the robot is relative to the landmarks. The further it moves, the more uncertain it becomes. Both in the localization and also the position of the landmarks in the map. The uncertainty increases until it sees a known landmark.

Loop closure detection: Percieving a previously known landmark, with high certainty. This can recalibrate the localisation, and thus make it more certain of where it is.
In order to obtain loop closure detection, we need data association. The loop closure detection is bounded by the certainty of the previously known landmark.

Mathematical definition

- Robot pose at time t is defined as x_t
- The path $X_T = \{x_0, x_1, ..., x_T\}$
-
-
-
-
-

2.2 SLAM

Full SLAM is recovering the model of the map M and the path X_T given the Observations Z_T and the input U_T

$$P(X_T, M|Z_T, U_T) \tag{1}$$

Online SLAM is only focussing on the next robot position.
we need

- The probabilistic motion model: $P(x_t, ...)$

2.3 EKF SLAM

The EKF SLAM is slow, because of the computation of the covariance matrix, and since the state vector is extended.
The output is a sparse vector which only contains the robot state, because we assume that the landmarks does not move.
We also start by assuming that the correlation matrix is diagonal, this entails the landmarks are independent.
But as the robot moves along, and aquires more information, the correlation matrix gets nonsparse.
It is **Regularily covisible** and it is highly correlated to convergence.
It should be robust to new landmarks. This is done by increasing the dimension of the correlation matrix by 1.
When the loop closure concept occurs, it reduced the uncertainty across all the landmarks. This is because of the update of the correlation matrix. The closer landmarks are to the known landmarks, the more certain they become when the loop closes.
You tune the validation gate to reduce the change of wrong landmark association