Lecture Automotive Vision 2019

Institut für Mess- und Regelungstechnik, Karlsruher Institut für Technologie
Dr. Martin Lauer, M.Sc. Annika Meyer, M.Sc. Danial Kamran

This excercise sheet covers the *Simultaneous Localization and Mapping (SLAM)* problem to localize a moving robot in an unknown world, while – simultaneously – generating a map of the environment. The localization of the robot refers to the estimation of its position, while the mapping of the environment refers to the estimation of the positions of the landmarks in the world.

To estimate the position of the robot and the positions of the landmarks simultaneously, the *Extended Kalman Filter* – a variant of the *Kalman Filter* for non-linear state and observation equations – is applied in this task.

Nomenclature

The pose of the moving robot is represented on the two-dimensional ground plane by $(\vec{x}_t, \phi_t)$ with $\vec{x}_t$ being the position and ϕ_t the orientation (yaw angle). Each landmark in the environment is represented by its position $\vec{y}^{(i)}$. The moving robot is equipped with two sensors for (a) measuring the movement from one time step to the next one and (b) measuring the position of the landmark relative to the robot's current pose. The relative motion from time step t to time step $t+1$ is given by $(\Delta\vec{x}_t, \Delta\phi_t)$, where $\Delta\vec{x}_t$ denotes the translation vector and $\Delta\phi_t$ the rotation angle of the robot. The observed landmarks at time t are represented by pairs $(ID_t^{(m)}, \vec{p}_t^{(m)})$ where $ID_t^{(m)}$ provides the identifier (the index) of the m-th observed landmark and $\vec{p}_t^{(m)}$ is the sensed position. All variables are summarized in table 1.

Estimation of the Dynamic State with an EKF

The Extended Kalman Filter should be used in this task to simultaneously estimate the pose of the moving robot and the positions of the landmarks. Hence, the state to be

Table 1: Summary of the variables used throughout this assignment

	variable	coordinate system
time index	t	
robot pose at time t	$(\vec{x}_t, \phi_t)$	world frame
landmark position of i-th landmark	$\vec{y}^{(i)}$	world frame
movement of robot from time t to $t+1$	$(\Delta\vec{x}_t, \Delta\phi_t)$	vehicle frame at time t
m-th observation of a landmark at time t	$(ID_t^{(m)}, \vec{p}_t^{(m)})$	vehicle frame at time t

estimated is composed of the pose of the robot and the positions of the landmarks, i.e.

$$\vec{s}_t = (\underbrace{\vec{x}_t, \phi_t}_{\text{robot}}, \underbrace{\vec{y}_t^{(1)}}_{\text{landmark 1}}, \ldots, \underbrace{\vec{y}_t^{(N)}}_{\text{landmark N}})^T$$

Since the number of landmarks is not known beforehand, state vector and covariance matrix have to be extended dynamically whenever new landmarks are observed for the first time[1].

System Model

The system model describes the evolution of the system state by incorporating the movement of the robot provided by $(\Delta\vec{x}_t, \Delta\phi_t)$. Given the pose of the robot at time step t as $(\vec{x}_t, \phi_t)^T$, the pose at the next time step can be computed by:

$$\begin{pmatrix} \vec{x}_{t+1} \\ \phi_{t+1} \end{pmatrix} = \begin{pmatrix} \vec{x}_t \\ \phi_t \end{pmatrix} + \begin{pmatrix} \cos(\phi_t) & -\sin(\phi_t) & 0 \\ \sin(\phi_t) & \cos(\phi_t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \Delta\vec{x}_t \\ \Delta\phi_t \end{pmatrix}$$

Exercise (6.1):

Derive the system model equation for a system state $\vec{s}_t$ with N landmarks.

Exercise (6.2):

Given the system model, derive the prediction equation for the Extended Kalman Filter. Design an appropriate covariance matrix for the system uncertainty.

Observation Model

The observation model describes the relative position of the observed landmarks with respect to the present robot pose.

Exercise (6.3):

Derive the observation model equation for a system state $\vec{s}_t$ with N landmarks to landmark with $ID = m$ $(1 \leq m \leq N)$.

[1] Note that the landmarks are not necessarily observed in order of increasing IDs, e.g. the landmark with $ID = 5$ might be observed before the landmark with $ID = 1$ has been observed.

Exercise (6.4):

Given the observation model, derive the measurement equation for the Extended Kalman Filter for an arbitrary landmark with $ID = m$ as in excercise 6.3. Design an appropriate covariance matrix for the measurement uncertainty.

Matlab Implementation of EKF-SLAM

Based on the theoretical derivations of the system and observation equations of the Extended Kalman Filter, we will implement the EKF-SLAM algorithm in Matlab in this task. The data for this task is available in file *data_lecture.mat*. After loading the Matlab data file, the following variables are defined in the workspace.

`initial_state`	The initial robot pose $(\vec{x}_0, \phi_0)$. Note that landmarks have not yet been observed so that the initial state only contains the initial pose of the robot.
`odom_meas`	The movement of the robot from one point in time to the next one. The movements are stored as `3 x numT` matrix with the first and second column representing $\Delta\vec{x}$ and the third on representing $\Delta\phi$. Note that the first row describes the movement from the initial pose $(\vec{x}_0, \phi_0)$ to $(\vec{x}_1, \phi_1)$, the second row from $(\vec{x}_1, \phi_1)$ to $(\vec{x}_2, \phi_2)$, etc.
`noise_odom`	The covariance matrix of the uncertainty of the robot movement as a `3 x 3` matrix.
`noise_observation`	The variance σ^2 of the measurement uncertainty as a scalar value. This variance should be assumed for both components of the measurement vector.

A skeleton file for the implementation of the EKF-SLAM is provided in file *slam.m*. The skeleton loads the data and implements the high-level sequence of function calls to `slam_prediction` for the prediction of the system state and `slam_innovation` to perform the update of the robot's pose as well as the positions of the landmarks.

Exercise (6.5):

Implement the function `slam_prediction` to perform the EKF prediction step of the system state in file *slam_prediction.m*. Be aware that the state dimension varies over time according to the number of landmarks.

Exercise (6.6):

Implement the function `slam_innovation` to perform the EKF innovation step in file *slam_innovation.m*. It contains three parts to be implemented:

1. Extension of the state vector and covariance matrix if a landmark ID has been observed that is larger than the largest ID that is already represented in the state vector and covariance matrix.
2. Setup of the Jacobian of the observation matrix for all observed landmarks.
3. Computation of new state vector and covariance matrix.

The list of observations, as obtained from function `get_landmark_measurements` and provided as argument to the function `slam_innovation`, is represented as a list of Matlab structures. The number of observations is computed by `length(observations)`, while the ID and position of the landmark are accessed by `observations(index).id` and `observations(index).pos`, respectively. Note that the observations are randomly sampled each time when the *slam.m* function is started so that the results vary for each execution (see Figure 1 for one possible result).

For debugging purpose, a *step-wise* mode is available that allows to execute each prediction-innovation sequence step-by-step when pressing `Enter`. To enable the step-wise mode, the variable `stepwise` has to be set to `true`.

Exercise (6.7):

The function `get_landmark_measurements` accepts an argument called `radius` that specifies the search radius for landmarks in the vicinity of the vehicle. Perform experiments with varying radii to see the effects of loop closures.

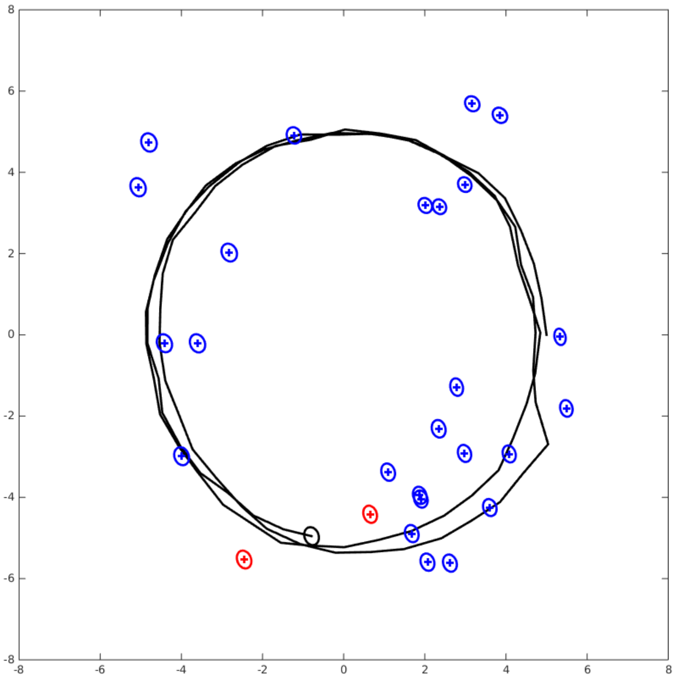

Figure 1: Result of EKF-SLAM