Simultaneous Localization and Mapping (SLAM)

EE468/CE468: Mobile Robotics

Dr. Basit Memon

Electrical and Computer Engineering
Habib University

November 22, 2023

Table of Contents

Table of Contents

What is SLAM?

- **Simultaneous Localization And Mapping** or **Concurrent Localization and Mapping**
- Estimate both the pose of robot and the map of **static** environment at the same time.
- **Localization:** Estimate pose of robot, given map of environment and measurements
- **Mapping:** Estimate map of environment, given pose of robot and measurements.

SLAM is a chicken or egg problem:

- Localization requires map and Mapping requires pose
- Considered fundamental problem for existence of autonomous robots.
- History of SLAM dates back to 1986 [1].
- Most algorithms use probabilistic formulation.

Central issue of SLAM:

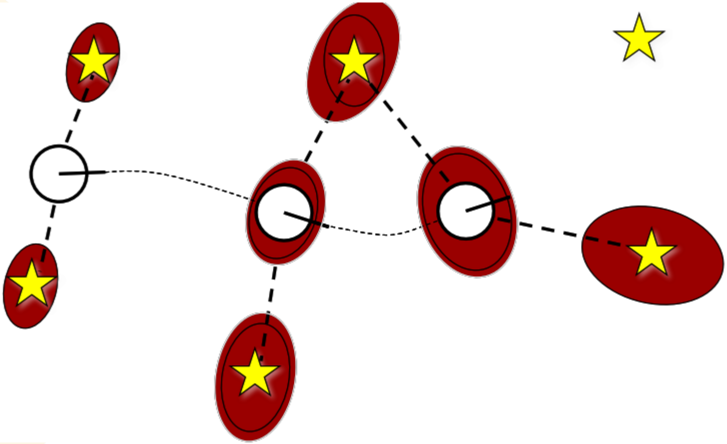


Figure: Errors in pose and map location are correlated. Image credit: [1]

Table of Contents

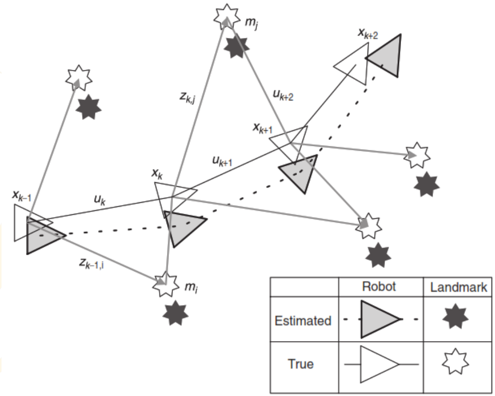


Figure: Constituents of SLAM problem. Image credit: [1]

- **Given:**
  - Control: Obtained from control algorithm or odometry

$$\mathbf{U}_{1:t} = \{u_1, u_2, \cdots, u_t\}$$

  - Observations: observations of landmarks relative to robot

$$\mathbf{Z}_{1:t} = \{z_1, z_2, \cdots, z_t\}$$

Observation at a given time may be multidimensional, e.g. readings from each beam of a LiDAR.

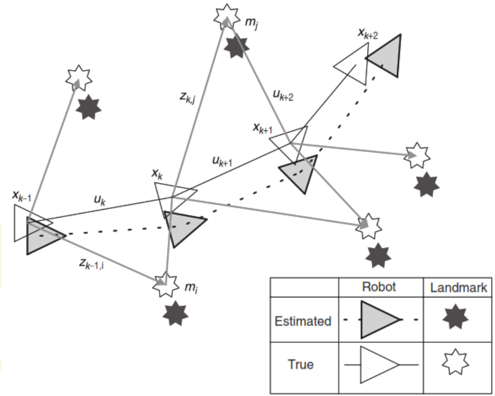


Figure: Constituents of SLAM problem. Image credit: [1]

- **Wanted:**
  - Map of features: Absolute positions of all landmarks

$$\mathbf{m} = \{m_1, m_2, \cdots, m_n\}$$

  - Path of robot: Absolute pose

$$\mathbf{X}_{1:t} = \{x_1, x_2, \cdots, x_t\} \text{ or } x_t$$

We want absolute positions from relative observations.

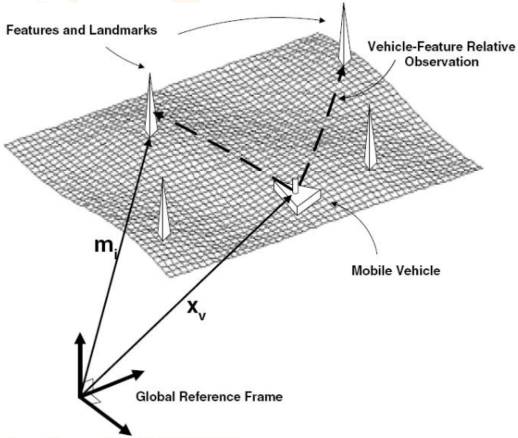
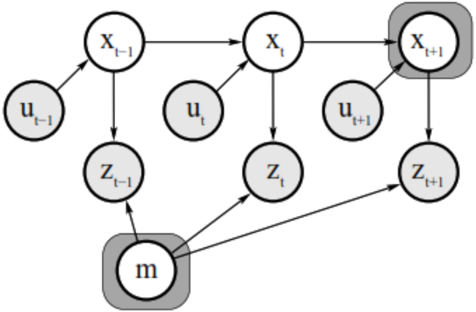


Figure: Image credit: W.Burgard

- Absolute poses
- Absolute landmark positions
- But we make only relative measurements of landmarks. Measurements are in robot frame.



- Online SLAM problem can be expressed as finding

$$p(x_t, m \mid z_{1:t}, u_{1:t}).$$

- The Full SLAM problem is that of finding

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t}).$$

KF can be used to solve feature-based SLAM problem.

Filter Cycle

1. State Prediction
2. Measurement Prediction
3. Measurement
4. State Update

Extended Kalman Filter

1. $\bar{\mu}_t = g(\mu_{t-1}, u_t)$
2. $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$
3. $K_t = \bar{\Sigma}_t \, H_t^T \, (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$
4. $\mu_t = \bar{\mu}_t + K_t \, (z_t - h(\bar{\mu}_t))$
5. $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

$$G_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}$$

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t}$$

EKF-SLAM extends state vector to include landmarks.

- Previously,

3 × 1 pose vector
$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

3 × 3 Covariance matrix
$$\Sigma_t = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{x\theta}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{y\theta}^2 \\ \sigma_{\theta x}^2 & \sigma_{\theta y}^2 & \sigma_{\theta\theta}^2 \end{bmatrix}$$

EKF-SLAM extends state vector to include landmarks.

- State vector and covariance matrix grows with landmarks

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_R \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}, \qquad \Sigma_t = \begin{bmatrix} \Sigma_R & \Sigma_{RM_1} & \Sigma_{RM_2} & \cdots & \Sigma_{RM_n} \\ \Sigma_{M_1R} & \Sigma_{M_1} & \Sigma_{M_1M_2} & \cdots & \Sigma_{M_1M_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M_nR} & \Sigma_{M_nM_1} & \Sigma_{M_nM_2} & \cdots & \Sigma_{M_n} \end{bmatrix}$$

- What is m_i?
  - It could be multidimensional vector describing feature, e.g. x and y coordinates of location of landmark.

- State is multi-dimensional Gaussian.

Table of Contents

A simple landmark SLAM problem with known correspondences

- Given
  - a robot that doesn't care about orientation
  - motion commands are steps in the x and y directions, but noisy
  - robot can measure all of the landmarks at all times in a fixed order
  - robot measures relative offset from its position to each landmark
- Wanted
  - robot's position, (x_t, y_t)
  - landmark locations, $(m_{i,x}, m_{i,y})$

Motion Model

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ m_{1,x} \\ m_{1,y} \\ m_{2,x} \\ m_{2,y} \\ \vdots \\ m_{n,x} \\ m_{n,y} \end{bmatrix} = \begin{bmatrix} x_{k} \\ y_{k} \\ m_{1,x} \\ m_{1,y} \\ m_{2,x} \\ m_{2,y} \\ \vdots \\ m_{n,x} \\ m_{n,y} \end{bmatrix} + \begin{bmatrix} \Delta x_{k} \\ \Delta y_{k} \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \epsilon_{k,x} \\ \epsilon_{k,y} \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + Bu_k + w_k,$$

where $A = B = I_{2n+2}$.

Measurement Model

$$z_k = \begin{bmatrix} m_{1,x} - x_k \\ m_{1,y} - y_k \\ m_{2,x} - x_k \\ m_{2,y} - y_k \\ \vdots \\ m_{n,x} - x_k \\ m_{n,y} - y_k \end{bmatrix} + \begin{bmatrix} \nu_{k,1x} \\ \nu_{k,1y} \\ \nu_{k,2x} \\ \nu_{k,2y} \\ \vdots \\ \nu_{k,nx} \\ \nu_{k,ny} \end{bmatrix}$$

$$\mathbf{z}_k = C\mathbf{x}_k + n_k,$$

where

$$C = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

Go to MATLAB

Live Script: `ekfslam.mlx`

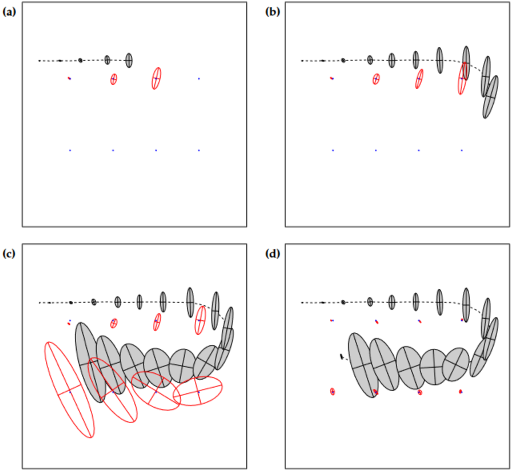


Figure 10.3 EKF applied to the online SLAM problem. The robot's path is a dotted line, and its estimates of its own position are shaded ellipses. Eight distinguishable landmarks of unknown location are shown as small dots, and their location estimates are shown as white ellipses. In (a)–(c) the robot's positional uncertainty is increasing, as is its uncertainty about the landmarks it encounters. In (d) the robot senses the first landmark again, and the uncertainty of *all* landmarks decreases, as does the uncertainty of its current pose. Image courtesy of Michael Montemerlo, Stanford University.

EKF-SLAM Application

Figure: courtesy by John Leonard

EKF-SLAM Application

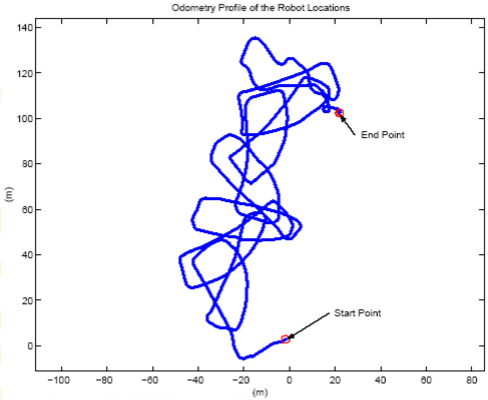


Figure: Path through odometry only [courtesy of John Leonard]

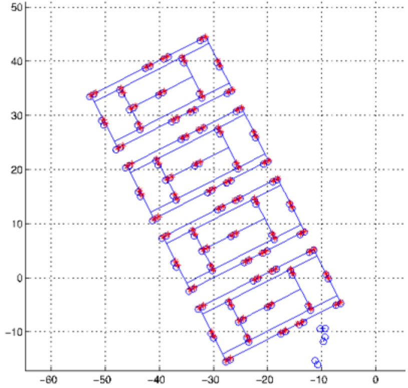


Figure: SLAM path with landmarks [courtesy of John Leonard]

Table of Contents

Filter Cycle

1. State Prediction
2. Measurement Prediction
3. Measurement
4. Data Association
5. State Update
6. Addition of new landmarks

Extended Kalman Filter

1. $\bar{\mu}_t = g(\mu_{t-1}, u_t)$
2. $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$
3. $K_t = \bar{\Sigma}_t \, H_t^T \, (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$
4. $\mu_t = \bar{\mu}_t + K_t \, (z_t - h(\bar{\mu}_t))$
5. $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

$$G_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}$$

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t}$$

EKF SLAM: State Prediction

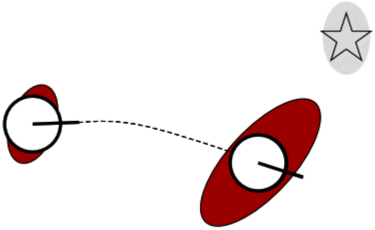


Figure: Image credit: W.Burgard

- $\bar{\mathbf{x}}_R = g(\mathbf{x}_R, u)$
- Landmarks are assumed static and will not be updated, i.e.

$$\dot{m}_i = 0 \text{ or } m_i^t = m_i^{t-1}$$

- $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$

EKF SLAM: Measurement Prediction

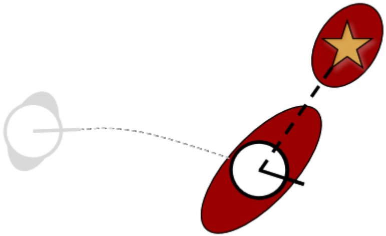


Figure: Image credit: W.Burgard

- Use predicted state $\bar{\mathbf{x}}_t$ to predict the measurement at the current pose.

$$\hat{z}_t = h(\hat{\mathbf{x}}_t)$$

- Recall that $\bar{x}$ contains landmark locations in global frame, and function h converts them to robot frame.

EKF SLAM: Obtain Measurements

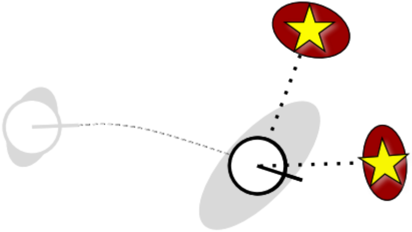


Figure: Image credit: W.Burgard

- (x, y) of landmarks in perception field

$$\mathbf{z_t} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} x_{l1} \\ y_{l1} \\ x_{l2} \\ y_{l2} \end{bmatrix}$$

- We also obtain Q_t, covariance matrix for measurements.

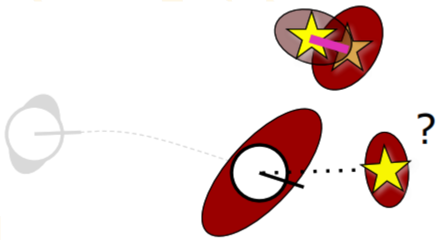


Figure: Image credit: W.Burgard

- At each time t, for each measurement z^i,

$$\nu_k = z_t^i - \bar{z}_t^k, \quad k \in \{1, \cdots, \text{Number of landmarks}\}$$

$$S_k = Q_t + H_t^k \bar{\Sigma}_t {H_t^k}^T$$

- Associate most likely landmark with measurement

$$\pi_k = \nu_k^T S_k^{-1} \nu_k$$

$$j(i) = \arg\min_k \pi_k$$

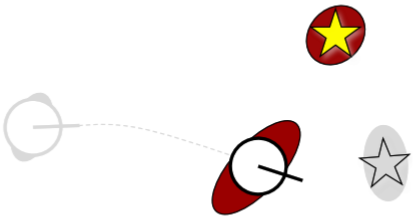


Figure: Image credit: W.Burgard

- Same old KF equations
- $K_t = \bar{\Sigma}_t \, H_t^T \, (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$
- $\mu_t = \bar{\mu}_t + K_t \, (z_t - h(\bar{\mu}_t))$
- $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

EKF SLAM: New Landmark

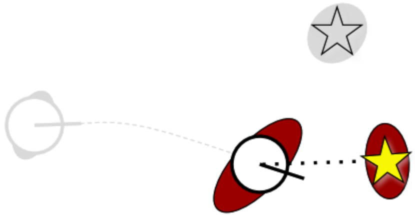


Figure: Image credit: W.Burgard

- If π_k is not below some threshold for any k, then add new landmark.

$$\mathbf{x} = \begin{bmatrix} x_R \\ m_1 \\ \vdots \\ m_n \\ m_{n+1} \end{bmatrix}$$

- Mean state (belief) for this new landmark is updated as:

$$\begin{bmatrix} \mu_{n+1,x} \\ \mu_{n+1,y} \end{bmatrix} = \begin{bmatrix} \mu_{t,x} + r_t^i \cos(\phi_t^i + \mu_{t,\theta}) \\ \mu_{t,y} + r_t^i \sin(\phi_t^i + \mu_{t,\theta}) \end{bmatrix},$$

where r_t^i, ϕ_t^i is measurement.

Map Management

- Number of landmarks is usually smaller than 1000.
- Gaussian noise assumption results in spurious measurements in tail end, and create fake landmarks.
- Fake landmarks adversely affect localization.
- Maintain a provisional landmark list and move landmark to main list only if it is consistently observed or log odds ratio for each landmark.

EKF SLAM is sensitive to data association.

- It is not robust to landmark confusion.
- Choose landmarks far from each other. Tradeoff with localization efficacy.
- Assign signatures. Maximize perceptual distinctiveness of landmarks, e.g. different colors.
- Few features make data association problem harder.

Table of Contents

[1] **Hugh Durrant-Whyte and Tim Bailey.**
Simultaneous localization and mapping: part i.
IEEE robotics & automation magazine, 13(2):99–110, 2006.