CSE-571
Probabilistic Robotics

SLAM: Simultaneous Localization and Mapping

Many slides courtesy of Ryan Eustice, Cyrill Stachniss, John Leonard

Today's Topic
- EKF Feature-Based SLAM
  - State Representation
  - Process / Observation Models
  - Landmark Initialization
  - Robot-Landmark Correlation

The SLAM Problem
A robot is exploring an unknown, static environment.

Given:
- The robot's controls
- Observations of nearby features

Estimate:
- Map of features
- Path of the robot

SLAM Applications

Indoors
Space
Undersea
Underground
Illustration of SLAM without Landmarks

With only dead reckoning, vehicle pose uncertainty grows without bound

Courtesy J. Leonard
With only dead reckoning, vehicle pose uncertainty grows without bound

Mapping with Raw Odometry

Repeat, with Measurements of Landmarks

First position: two features observed

Illustration of SLAM without Landmarks

Illustration of SLAM without Landmarks
Illustration of SLAM with Landmarks

- Second position: two new features observed
  Courtesy J. Leonard

Illustration of SLAM with Landmarks

- Re-observation of first two features results in improved estimates for both vehicle and feature
  Courtesy J. Leonard

Illustration of SLAM with Landmarks

- Third position: two additional features added to map
  Courtesy J. Leonard

Illustration of SLAM with Landmarks

- Re-observation of first four features results in improved location estimates for vehicle and all features
  Courtesy J. Leonard
Illustration of SLAM with Landmarks

Process continues as the vehicle moves through the environment

SLAM Using Landmarks

MIT Indoor Track

Test Environment (Point Landmarks)

View from Vehicle
SLAM Using Landmarks

1. Move
2. Sense
3. Associate measurements with known features
4. Update state estimates for robot and previously mapped features
5. Find new features from unassociated measurements
6. Initialize new features
7. Repeat

MIT Indoor Track

Comparison with Ground Truth

Simultaneous Localization and Mapping (SLAM)

- Building a map and locating the robot in the map at the same time
- Chicken-and-egg problem

Definition of the SLAM Problem

**Given**
- The robot’s controls
  \[ u_{1:T} = \{ u_1, u_2, u_3, \ldots, u_T \} \]
- Observations
  \[ z_{1:T} = \{ z_1, z_2, z_3, \ldots, z_T \} \]

**Wanted**
- Map of the environment
  \[ m \]
- Path of the robot
  \[ x_{0:T} = \{ x_0, x_1, x_2, \ldots, x_T \} \]
Three Main Paradigms

- **Kalman filter**
- **Particle filter**
- **Graph-based**

Bayes Filter

- Recursive filter with prediction and correction step

**Prediction**

\[
\overline{\text{bel}}(x_t) = \int p(x_t \mid u_{t-1}, x_{t-1}) \overline{\text{bel}}(x_{t-1}) \, dx_{t-1}
\]

**Correction**

\[
\text{bel}(x_t) = \eta \, p(z_t \mid x_t) \overline{\text{bel}}(x_t)
\]

EKF for Online SLAM

- We consider here the Kalman filter as a solution to the online SLAM problem

\[
p(x_t, m \mid z_{1:t}, u_{1:t})
\]

Extended Kalman Filter Algorithm

1. `Extended_Kalman_filter(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)`:
2. \[ \hat{\mu}_t = g(u_t, \mu_{t-1}) \]
3. \[ \Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t \]
4. \[ K_t = \Sigma_t H_t^T (H_t \Sigma_t H_t^T + Q_t)^{-1} \]
5. \[ \mu_t = \hat{\mu}_t + K_t (z_t - h(\hat{\mu}_t)) \]
6. \[ \Sigma_t = (I - K_t H_t) \Sigma_t \]
7. return \( \mu_t, \Sigma_t \)
EKF SLAM

- Application of the EKF to SLAM
- Estimate robot's pose and locations of landmarks in the environment
- Assumption: known correspondences
- State space (for the 2D plane) is

\[ x_t = \begin{pmatrix} x, y, \theta, m_{1,x}, m_{1,y}, \ldots, m_{n,x}, m_{n,y} \end{pmatrix}^T \]

robot's pose landmark 1 landmark n

EKF SLAM: State Representation

- Map with n landmarks: (3+2n)-dimensional Gaussian
- Belief is represented by

\[
\begin{pmatrix}
\begin{pmatrix}
\mathbf{x}_R \\
\mathbf{m}_1 \\
\vdots \\
\mathbf{m}_n
\end{pmatrix}

\begin{pmatrix}
\Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\
\Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n}
\end{pmatrix}

\end{pmatrix}
\]

EKF SLAM: State Representation

- More compactly

\[
\begin{pmatrix}
\mathbf{x}_R \\
\mathbf{m}
\end{pmatrix}
\]

\[
\begin{pmatrix}
\Sigma_{x x} & \Sigma_{x m} \\
\Sigma_{m x} & \Sigma_{m m}
\end{pmatrix}
\]

EKF SLAM: State Representation

- Even more compactly (note: \( x_R \rightarrow x \))

\[
\begin{pmatrix}
\mathbf{x} \\
\mathbf{m}
\end{pmatrix}
\]

\[
\begin{pmatrix}
\Sigma_{x x} & \Sigma_{x m} \\
\Sigma_{m x} & \Sigma_{m m}
\end{pmatrix}
\]
EKF SLAM: Filter Cycle

1. State prediction
2. Measurement prediction
3. Measurement
4. Data association
5. Update
EKF SLAM: Data Association and Difference Between $h(x)$ and $z$

$$\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_{R,i} m_i \\ \sum_{i=1}^{n} m_i x_{R,i} \\ \vdots \\ \sum_{i=1}^{n} m_i x_{R,i} \\ \sum_{i=1}^{n} m_i m_i \end{pmatrix}$$

EKF SLAM: Update Step

$$\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_{R,i} m_i \\ \sum_{i=1}^{n} m_i x_{R,i} \\ \vdots \\ \sum_{i=1}^{n} m_i x_{R,i} \\ \sum_{i=1}^{n} m_i m_i \end{pmatrix}$$

EKF SLAM: Concrete Example

**Setup**
- Robot moves in the 2D plane
- Velocity-based motion model
- Robot observes point landmarks
- Range-bearing sensor
- Known data association
- Known number of landmarks

Initialization
- Robot starts in its own reference frame (all landmarks unknown)
- $2N+3$ dimensions

$$\mu_0 = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \end{pmatrix}^T$$

$$\Sigma_0 = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots \end{pmatrix}$$
Extended Kalman Filter Algorithm

1: Extended_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$);
2: $\hat{\mu}_t = \hat{g}(u_t, \mu_{t-1})$
3: $\hat{\Sigma}_t = \hat{G}_t \Sigma_{t-1} \hat{G}_t^T + \hat{R}_t$
4: $K_t = \hat{\Sigma}_t H_t^T (H_t \hat{\Sigma}_t H_t^T + Q_t)^{-1}$
5: $\mu_t = \hat{\mu}_t + K_t (z_t - h(\hat{\mu}_t))$
6: $\Sigma_t = (I - K_t H_t) \hat{\Sigma}_t$
7: return $\mu_t, \Sigma_t$

Prediction Step (Motion)

- Goal: Update state space based on the robot's motion
- Robot motion in the plane

\[
\begin{pmatrix}
\dot{x} \\ \dot{y} \\ \dot{\theta}
\end{pmatrix} =
\begin{pmatrix}
x \\ y \\ \theta
\end{pmatrix} + \begin{pmatrix}
-\frac{2 \omega \sin \theta + 2 \omega \sin(\theta + \omega \Delta t)}{\omega^2} \\ \frac{2 \omega \cos \theta - 2 \omega \cos(\theta + \omega \Delta t)}{\omega^2}
\end{pmatrix} \omega \Delta t
\]

- How to map that to the 2N+3 dim space?

Update the State Space

- From the motion in the plane

\[
\begin{pmatrix}
\dot{x}' \\ \dot{y}' \\ \dot{\theta}'
\end{pmatrix} =
\begin{pmatrix}
x \\ y \\ \theta
\end{pmatrix} + \begin{pmatrix}
1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 1
\end{pmatrix} \begin{pmatrix}
-\frac{2 \omega \sin \theta + 2 \omega \sin(\theta + \omega \Delta t)}{\omega^2} \\ \frac{2 \omega \cos \theta - 2 \omega \cos(\theta + \omega \Delta t)}{\omega^2}
\end{pmatrix} \omega \Delta t
\]

- to the 2N+3 dimensional space

Extended Kalman Filter Algorithm

1: Extended_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$);
2: $\hat{\mu} = \hat{g}(u_t, \mu_{t-1})$ \text{ DONE}
3: $\hat{\Sigma} = \hat{G}_t \Sigma_{t-1} \hat{G}_t^T + \hat{R}_t$
4: $K_t = \hat{\Sigma} H_t^T (H_t \hat{\Sigma} H_t^T + Q_t)^{-1}$
5: $\mu_t = \hat{\mu} + K_t (z_t - h(\hat{\mu}))$
6: $\Sigma_t = (I - K_t H_t) \hat{\Sigma}$
7: return $\mu_t, \Sigma_t$
Update Covariance

- The function $\nu$ only affects the robot’s motion and not the landmarks

\[ G_t = \begin{pmatrix} G_t^x & 0 \\ 0 & I \end{pmatrix} \]

Identity $(2N \times 2N)$

This Leads to the Time Propagation

1. \texttt{Extended Kalman filter}\($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$): \texttt{DONE}
2. $\mu_t = g(\mu_{t-1}, u_t, \nu)$ \texttt{Apply \& DONE}
3. $\Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t$

\[ G_t = \begin{pmatrix} G_t^x & 0 \\ 0 & I \end{pmatrix}, \quad \Sigma_{t-1} = \begin{pmatrix} \Sigma_{t-1}^{xx} & \Sigma_{t-1}^{xz} \\ \Sigma_{t-1}^{zx} & \Sigma_{t-1}^{zm} \end{pmatrix}, \quad R_t = \begin{pmatrix} R_t^{xx} & 0 \\ 0 & R_t^{zz} \end{pmatrix} \]

Extended Kalman Filter Algorithm

1. \texttt{Extended Kalman filter}\($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$): \texttt{DONE}
2. $\mu_t = g(\mu_{t-1}, u_t, \nu)$ \texttt{DONE}
3. $\Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t$ \texttt{DONE}
4. $K_t = \Sigma_t H_t^T (H_t \Sigma_t H_t^T + Q_t)^{-1}$
5. $\mu_t = \hat{\mu}_t + K_t (z_t - h(\hat{\mu}_t))$
6. $\Sigma_t = (I - K_t H_t) \Sigma_t$
7. return $\mu_t, \Sigma_t$

EKF SLAM: Correction Step

- Known data association
- $z_t^j$: $i$-th measurement at time $t$ observes the landmark with index $j$
- Initialize landmark if unobserved
- Compute the expected observation
- Compute the Jacobian of $h$
- Proceed with computing the Kalman gain
Range-Bearing Observation

- **Range-Bearing observation** $z_i^r = (r_i^r, \phi_i^r)^T$
- If landmark has not been observed
  \[
  \begin{pmatrix}
  \hat{\mu}_{j,x} \\
  \hat{\mu}_{j,y}
  \end{pmatrix}
  = \begin{pmatrix}
  \hat{\mu}_{t,x} \\
  \hat{\mu}_{t,y}
  \end{pmatrix} + \begin{pmatrix}
  r_i^r \cos(\phi_i^r + \hat{\mu}_{t,\theta}) \\
  r_i^r \sin(\phi_i^r + \hat{\mu}_{t,\theta})
  \end{pmatrix}
  \]
  
Jacobian for the Observation

- Based on
  \[
  \delta = \begin{pmatrix}
  \delta_x \\
  \delta_y
  \end{pmatrix},
  q = \delta^T \delta
  \]
  \[
  \dot{z}_i^q = \begin{pmatrix}
  \tan^{-1}(\delta_y / \delta_x) - \hat{\mu}_{t,\theta}
  \end{pmatrix}
  \]
  
- Compute the Jacobian
  \[
  H^i_t = \frac{\partial h(\dot{\mu}_t)}{\partial \dot{\mu}_t} = 1/q \begin{pmatrix}
  -\sqrt{\delta_x} & -\sqrt{\delta_y} & 0 & +\sqrt{\delta_x} & \sqrt{\delta_y} \\
  \delta_y & -\delta_x & -q & -\delta_y & \delta_x
  \end{pmatrix}
  \]

- Use the computed Jacobian
  \[
  \text{low } H^i_t = \frac{1}{q} \begin{pmatrix}
  -\sqrt{\delta_x} & -\sqrt{\delta_y} & 0 & +\sqrt{\delta_x} & \sqrt{\delta_y} \\
  \delta_y & -\delta_x & -q & -\delta_y & \delta_x
  \end{pmatrix}
  \]

- Map it to the high dimensional space
  \[
  H^i_t = \text{low } H^i_t F_{x,i}
  \]

Next Steps as Specified...

1. **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:
   - $\dot{\mu}_t = g(u_t, \mu_{t-1})$ **DONE**
   - $\Sigma_t = G_t \Sigma_{t-1} \Sigma_t T + R_t$ **DONE**

4. **$K_t = \Sigma_t H^T_t (H_t \Sigma_t H^T_t + Q_t)^{-1}$**

5. $\mu_t = \mu_{t-1} + K_t (z_t - h(\mu_t))$

6. $\Sigma_t = (I - K_t H_t) \Sigma_t$

7. Return $\mu_t, \Sigma_t$
### Extended Kalman Filter Algorithm

1. \( \tilde{\mu}_t = \tilde{q}(\tilde{\mu}_{t-1}, \Sigma_{t-1}, u_t, z_t) \) DONE
2. \( \Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t \) DONE
3. \( K_t = \Sigma_t H_t^T (H_t \Sigma_t H_t^T + Q_t)^{-1} \) Apply & DONE
4. \( \tilde{\mu}_t = \tilde{\mu}_t + K_t (z_t - h(\tilde{\mu}_t)) \) Apply & DONE
5. \( \tilde{\Sigma}_t = (I - K_t H_t) \Sigma_t \) Apply & DONE
6. return \( \tilde{\mu}_t, \Sigma_t \)

### EKF SLAM – Correction (1/2)

1. \( \bar{Q}_t = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \)
2. for all observed features \( z_i = (r_i, \theta_i)^T \) do
3. \( j = i \)
4. if landmark \( j \) never seen before then
5. \( \bar{\mu}_{j,x} = \frac{\mu_{j,x} + \tilde{\mu}_{j,x}}{2} + \frac{v_j \cos(\theta_j + \tilde{\mu}_{j,y})}{2} \)
6. \( \bar{\mu}_{j,y} = \frac{\mu_{j,y} + \tilde{\mu}_{j,y}}{2} + \frac{v_j \sin(\theta_j + \tilde{\mu}_{j,y})}{2} \)
7. endif
8. \( \delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \tilde{\mu}_{j,x} \\ \bar{\mu}_{j,y} - \tilde{\mu}_{j,y} \end{pmatrix} \)
9. \( \gamma = \delta^T \delta \)
10. \( \tilde{z}_j = \sqrt{\gamma} \cdot \tan^{-1}(\frac{\delta_y}{\delta_x}) - \tilde{\mu}_{j,y} \)

### EKF SLAM Complexity

- Cubic complexity depends only on the measurement dimensionality
- Cost per step: dominated by the number of landmarks: \( O(n^2) \)
- Memory consumption: \( O(n^2) \)
- The EKF becomes computationally intractable for large maps!
In the limit, the landmark estimates become fully correlated.

Approximate the SLAM posterior with a high-dimensional Gaussian [Smith & Cheesman, 1986]...

Single hypothesis data association.
EKF SLAM Correlations

- Map
- Correlation matrix

EKF SLAM Uncertainties

- The determinant of any sub-matrix of the map covariance matrix decreases monotonically.
- New landmarks are initialized with maximum uncertainty.

Data Association in SLAM

- In the real world, the mapping between observations and landmarks is unknown.
- Picking wrong data associations can have catastrophic consequences.
- EKF SLAM is brittle in this regard.
- Pose error correlates data associations.

Data Association

- Given an environment map.
- And a set of sensor observations.
- Associate observations with map elements.

Vision

Laser
Difficulties: clutter
- Influence of the type, density, precision and robustness of features considered:

Laser scanner:
- Small amount of features ($n$)
- Small amount of measurements ($m$)
- Low spuriousness

Vertical Edge Monocular vision:
- Many features ($n$ large)
- Many measurements ($m$ large)
- No depth information
- Higher spuriousness

Difficulties: imprecision
- Both the sensor and the vehicle introduce imprecision

Vertical Edge Trinocular vision:
- Variable depth precision
- Good angular precision

Robot imprecision: introduces CORRELATED error

Loop-Closing
- Loop-closing means recognizing an already mapped area
- Data association under:
  - high ambiguity
  - possible environment symmetries
- Uncertainties collapse after a loop-closure (whether the closure was correct or not)

Courtesy: Cyril Stachniss
Before the Loop-Closure

After the Loop-Closure

Example: Victoria Park Dataset

Victoria Park: Data Acquisition
EKF SLAM Summary

- Quadratic in the number of landmarks: $O(n^2)$
- Convergence results for the linear case.
- Can diverge if nonlinearities are large!
- Have been applied successfully in large-scale environments.
- Approximations reduce the computational complexity.

Graph-SLAM

- Full SLAM technique
- Generates probabilistic links
- Computes map only occasionally
- Based on Information Filter form

Literature

**EKF SLAM**
- Thrun et al.: “Probabilistic Robotics”, Chapter 10
- Smith, Self, & Cheeseman: “Estimating Uncertain Spatial Relationships in Robotics”
- Dissanayake et al.: “A Solution to the Simultaneous Localization and Map Building (SLAM) Problem”
- Durrant-Whyte & Bailey: “SLAM Part 1” and “SLAM Part 2” tutorials

**Graph-SLAM**

- Full SLAM technique
- Generates probabilistic links
- Computes map only occasionally
- Based on Information Filter form
Graph-SLAM Idea

Information Form

- Represent posterior in canonical form
  \[ \Omega = \Sigma^{-1} \quad \text{Information matrix} \]
  \[ \xi = \Sigma^{-1} \mu \quad \text{Information vector} \]

- One-to-one transform between canonical and moment representation
  \[ \Sigma = \Omega^{-1} \]
  \[ \mu = \Omega^{-1} \xi \]
Graph-SLAM Inference (3)

Robot Poses and Scans [Lu and Milios 1997]

- Successive robot poses connected by odometry
- Sensor readings yield constraints between poses
- Constraints represented by Gaussians
  \[ D_i = \bar{D}_i + Q_i \]
- Globally optimal estimate
  \[ \arg\max_i \left[ P(D_i | \bar{D}_i) \right] \]

Loop Closure

- Use scan patches to detect loop closure
- Add new position constraints
- Deform the network based on covariances of matches

Mapping the Allen Center

Before loop closure

After loop closure
Mine Mapping

Efficient Map Recovery

• Information matrix inversion can be avoided if only best map estimate is required

• Minimize constraint function $J_{\text{GraphSLAM}}$ using standard optimization techniques (gradient descent, Levenberg Marquardt, conjugate gradient)

Mine Mapping: Data Associations

3D Outdoor Mapping

10^8 features, 10^5 poses, only few secs using cg.
**Graph-SLAM Summary**

- Adresses full SLAM problem
- Constructs link graph between poses and poses/landmarks
- Graph is sparse: number of edges linear in number of nodes
- Inference performed by building information matrix and vector (linearized form)
- Map recovered by reduction to robot poses, followed by conversion to moment representation, followed by estimation of landmark positions
- ML estimate by minimization of $J_{\text{GraphSLAM}}$
- Data association by iterative greedy search