The SLAM Problem

SLAM is the process by which a robot builds a map of the environment and, at the same time, uses this map to compute its location.

- **Localization**: inferring location given a map
- **Mapping**: inferring a map given a location
- **SLAM**: learning a map and locating the robot simultaneously
The SLAM Problem

- SLAM is a **chicken-or-egg problem**:  
  - A map is needed for localizing a robot  
  - A pose estimate is needed to build a map  

- Thus, SLAM is (regarded as) a **hard problem** in robotics
The SLAM Problem

- SLAM is considered one of the most fundamental problems for robots to become truly autonomous.

- A variety of different approaches to address the SLAM problem have been presented.

- **Probabilistic methods** rule.

- History of SLAM dates back to the mid-eighties (stone-age of mobile robotics).
The SLAM Problem

**Given:**

- The robot’s controls
  \[ U_{0:k} = \{u_1, u_2, \cdots, u_k\} \]
- Relative observations
  \[ Z_{0:k} = \{z_1, z_2, \cdots, z_k\} \]

**Wanted:**

- Map of features
  \[ m = \{m_1, m_2, \cdots, m_n\} \]
- Path of the robot
  \[ X_{0:k} = \{x_0, x_1, \cdots, x_k\} \]
The SLAM Problem

- **Absolute** robot pose
- **Absolute** landmark positions
- But only **relative** measurements of landmarks
SLAM Applications

SLAM is central to a range of indoor, outdoor, in-air and underwater applications for both manned and autonomous vehicles.

Examples:

• At home: vacuum cleaner, lawn mower
• Air: surveillance with unmanned air vehicles
• Underwater: reef monitoring
• Underground: exploration of abandoned mines
• Space: terrain mapping for localization
SLAM Applications

Indoors

Space

Undersea

Underground
Map Representations

Examples:
Subway map, city map, landmark-based map

Maps are **topological** and/or **metric models** of the environment
Map Representations

• Grid maps or scans, 2d, 3d

[Lu & Milios, 97; Gutmann, 98; Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;...]

• Landmark-based

[Leonard et al., 98; Castelanos et al., 99; Dissanayake et al., 2001; Montemerlo et al., 2002;...]
Why is SLAM a hard problem?

1. Robot path and map are both **unknown**

2. Errors in map and pose estimates correlated
Why is SLAM a hard problem?

- In the real world, the mapping between observations and landmarks is unknown (origin uncertainty of measurements)
- **Data Association**: picking wrong data associations can have catastrophic consequences (divergence)
**SLAM:**
**Simultaneous Localization And Mapping**

- **Full SLAM:**
  
  \[ p(x_{0:t}, m | z_{1:t}, u_{1:t}) \]

  *Estimates entire path and map!*

- **Online SLAM:**

  \[ p(x_t, m | z_{1:t}, u_{1:t}) = \int \int ... \int p(x_{1:t}, m | z_{1:t}, u_{1:t}) \, dx_1 \, dx_2 \, ... \, dx_{t-1} \]

  Integrations (marginalization) typically done recursively, one at a time

  *Estimates most recent pose and map!*
Graphical Model of Full SLAM

\[ p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \]
Graphical Model of Online SLAM

\[
p(x_t, m \mid z_{1:t}, u_{1:t}) = \int \int \ldots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \, dx_1 \, dx_2 \ldots dx_{t-1}
\]
Graphical Model: Models

\[ x_k = f(x_{k-1}, u_k) \]

Motion model

\[ z_k = h(x_k, m) \]

Observation model
Remember? KF Algorithm

1. Algorithm \texttt{Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):

2. Prediction:
3. \( \mu_t = A_t \mu_{t-1} + B_t u_t \)
4. \( \Sigma_t = A_t \Sigma_{t-1} A_t^T + R_t \)

5. Correction:
6. \( K_t = \Sigma_t C_t^T (C_t \Sigma_t C_t^T + Q_t)^{-1} \)
7. \( \mu_t = \mu_t + K_t (z_t - C_t \mu_t) \)
8. \( \Sigma_t = (I - K_t C_t) \Sigma_t \)

9. Return \( \mu_{tr}, \Sigma_t \)
EKF SLAM: State representation

- **Localization**

  3x1 pose vector

  3x3 cov. matrix

  \[
  x_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}, \quad C_k = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta}^2 \end{bmatrix}
  \]

- **SLAM**

  Landmarks are **simply added** to the state.

  **Growing** state vector and covariance matrix!

  \[
  x_k = \begin{bmatrix} x_R \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}_k, \quad C_k = \begin{bmatrix} C_R & C_{RM_1} & C_{RM_2} & \cdots & C_{RM_n} \\ C_{M_1 R} & C_{M_1} & C_{M_1 M_2} & \cdots & C_{M_1 M_n} \\ C_{M_2 R} & C_{M_2 M_1} & C_{M_2} & \cdots & C_{M_2 M_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{M_n R} & C_{M_n M_1} & C_{M_n M_2} & \cdots & C_{M_n} \end{bmatrix}_k
  \]
**EKF SLAM: State representation**

- **Map with** $n$ **landmarks:** $(3 + 2n)$-dimensional Gaussian

\[
\text{Bel}(x_t, m_t) = \begin{bmatrix}
    x \\
    y \\
    \theta \\
    l_1 \\
    l_2 \\
    \vdots \\
    l_N
\end{bmatrix},
\begin{bmatrix}
    \sigma^2_x & \sigma_{xy} & \sigma_{x\theta} \\
    \sigma_{xy} & \sigma^2_y & \sigma_{y\theta} \\
    \sigma_{x\theta} & \sigma_{y\theta} & \sigma^2_{\theta} \\
    \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_N} \\
    \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_N} \\
    \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_N} \\
    \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_N} \\
    \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_N} \\
    \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_N} \\
    \vdots & \vdots & \ddots & \vdots \\
    \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma^2_{xl_N} \\
    \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma^2_{yl_N} \\
    \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma^2_{\theta l_N}
\end{bmatrix}
\]

- **Can handle hundreds of dimensions**
EKF SLAM: Building the Map

Filter Cycle, Overview:

1. State prediction (odometry)
2. Measurement prediction
3. Observation
4. Data Association
5. Update
6. Integration of new landmarks

NEW
EKF SLAM: Building the Map

- State Prediction

Odometry:
\[ \hat{x}_R = f(x_R, u) \]
\[ \hat{C}_R = F_x C_R F_x^T + F_u U F_u^T \]

Robot-landmark cross-covariance prediction:
\[ \hat{C}_{RM_i} = F_x C_{RM_i} \]

(skipping time index \( k \))
EKF SLAM: Building the Map

• Measurement Prediction

Global-to-local frame transform $h$

$$\hat{z}_k = h(\hat{x}_k)$$

$$x_k = \begin{bmatrix} x_R \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}_k$$

$$C_k = \begin{bmatrix} C_R & C_{RM_1} & C_{RM_2} & \cdots & C_{RM_n} \\ C_{M_1 R} & C_{M_1} & C_{M_1 M_2} & \cdots & C_{M_1 M_n} \\ C_{M_2 R} & C_{M_2 M_1} & C_{M_2} & \cdots & C_{M_2 M_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{M_n R} & C_{M_n M_1} & C_{M_n M_2} & \cdots & C_{M_n} \end{bmatrix}_k$$
EKF SLAM: Building the Map

- Observation

\[ x_k = \begin{bmatrix} x_R \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}_k \]

\[ C_k = \begin{bmatrix} C_R & C_{RM_1} & C_{RM_2} & \cdots & C_{RM_n} \\ C_{M_1 R} & C_{M_1} & C_{M_1 M_2} & \cdots & C_{M_1 M_n} \\ C_{M_2 R} & C_{M_2 M_1} & C_{M_2} & \cdots & C_{M_2 M_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{M_n R} & C_{M_n M_1} & C_{M_n M_2} & \cdots & C_{M_n} \end{bmatrix}_k \]

\[ z_k = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \]

\[ R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \]
EKF SLAM: Building the Map

- Data Association

Associates predicted measurements $\mathbf{\hat{z}}_k^i$ with observation $\mathbf{z}_k^j$

\[
\nu_k^{ij} = \mathbf{z}_k^j - \mathbf{\hat{z}}_k^i \\
S_k^{ij} = R_k^j + H^i \mathbf{\hat{C}}_k H^i T
\]

(Gating)

\[
\mathbf{x}_k = \begin{bmatrix}
\mathbf{x}_R \\
\mathbf{m}_1 \\
\mathbf{m}_2 \\
\vdots \\
\mathbf{m}_n
\end{bmatrix}_k \\
\mathbf{C}_k = \begin{bmatrix}
\mathbf{C}_R & \mathbf{C}_{RM_1} & \mathbf{C}_{RM_2} & \cdots & \mathbf{C}_{RM_n} \\
\mathbf{C}_{M_1 R} & \mathbf{C}_{M_1} & \mathbf{C}_{M_2 M_1} & \cdots & \mathbf{C}_{M_1 M_n} \\
\mathbf{C}_{M_2 R} & \mathbf{C}_{M_2 M_1} & \mathbf{C}_{M_2} & \cdots & \mathbf{C}_{M_2 M_n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{C}_{M_n R} & \mathbf{C}_{M_n M_1} & \mathbf{C}_{M_n M_2} & \cdots & \mathbf{C}_{M_n}
\end{bmatrix}_k
\]
EKF SLAM: Building the Map

- Filter Update

The usual Kalman filter expressions

\[ K_k = \hat{C}_k H^T S_k^{-1} \]
\[ x_k = \hat{x}_k + K_k \nu_k \]
\[ C_k = (I - K_k H) \hat{C}_k \]
EKF SLAM: Building the Map

- Integrating New Landmarks

State augmented by
\[ \mathbf{m}_{n+1} = g(\mathbf{x}_R, z_j) \]
\[ C_{M_{n+1}} = G R C_R G_R^T + G_z R_j G_z^T \]

Cross-covariances:
\[ C_{M_{n+1} M_i} = G_R C_{RM_i} \]
\[ C_{M_{n+1} R} = G_R C_R \]
EKF SLAM

Map

Correlation matrix
EKF SLAM

Map

Correlation matrix
EKF SLAM

Map

Correlation matrix
EKF SLAM: Correlations Matter

- What if we **neglected** correlations?

\[
C_k = \begin{bmatrix}
C_R & 0 & \cdots & 0 \\
0 & C_{M_1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & C_{M_n}
\end{bmatrix}
\]

\[
C_{RM_i} = \begin{bmatrix} 0_{3\times 2} \end{bmatrix}
\]

\[
C_{M_i M_{i+1}} = \begin{bmatrix} 0_{2\times 2} \end{bmatrix}
\]

→ Landmark and robot uncertainties would become overly optimistic
→ Validation gates for matching too small
→ Data association would fail
→ Multiple map entries of the same landmark
→ Inconsistent map
SLAM: Loop Closure

- Loop closure is the problem of recognizing an already mapped area, typically after a long exploration path (the robot "closes a loop")

- Structurally identical to data association, but
  - high levels of ambiguity
  - possibly useless validation gates
  - environment symmetry

- Uncertainties collapse after a loop closure (whether the closure was correct or not)
SLAM: Loop Closure

- Before loop closure
SLAM: Loop Closure

• After loop closure
SLAM: Loop Closure

• By revisiting already mapped areas, uncertainties in robot and landmark estimates can be **reduced**

• This can be exploited to "**optimally**" explore an environment for the sake of better (e.g. more accurate) maps

• Exploration: the problem of **where to acquire new information** (e.g. depth-first vs. breadth first)

→ See separate chapter on exploration
**KF-SLAM Properties (Linear Case)**

- The **determinant** of any sub-matrix of the map covariance matrix **decreases monotonically** as successive observations are made.

- When a new landmark is initialized, its **uncertainty** is maximal.

- Landmark uncertainty **decreases monotonically** with each new observation.

[Dissanayake et al., 2001]
KF-SLAM Properties (Linear Case)

- In the limit, the landmark estimates become **fully correlated**

[Dissanayake et al., 2001]
In the limit, the covariance associated with any single landmark location estimate is determined only by the initial covariance in the vehicle location estimate.

[Disanayake et al., 2001]
EKF SLAM Example: Victoria Park

Sydney, Australia
Victoria Park: Data Acquisition

[courtesy by E. Nebot]
Victoria Park: Estimated Trajectory

[courtesy by E. Nebot]
Victoria Park: Landmarks

[courtesy by E. Nebot]
EKF SLAM Example: Tennis Court

[courtesy by J. Leonard]
EKF SLAM Example: Tennis Court

odometry

estimated trajectory

[courtesy by John Leonard]
EKF SLAM Example: Line Features

• KTH Bakery Data Set

[Wulf, Arras et al., ICRA 04]
EKF SLAM Example: AGV

- Pick-and-Place AGV at Geiger AG, Ludwigsburg (Project by LogObject/Nurobot)

[courtesy by LogObject/Nurobot]
EKF SLAM Example: AGV

- Pick-and-Place AGV at Geiger AG, Ludwigsburg (Project by LogObject/Nurobot)

[courtesy by LogObject/Nurobot]
SLAM Techniques

- EKF SLAM
- FastSLAM (PF)
- Graphical SLAM a.k.a Network-Based SLAM
- Topological SLAM (mainly place recognition)
- Scan Matching / Visual Odometry (only locally consistent maps)
EKF-SLAM: Complexity

- **Cost per step**: quadratic in $n$, the number of landmarks: $O(n^2)$
- **Total cost** to build a map with $n$ landmarks: $O(n^3)$
- **Memory**: $O(n^2)$

**Problem**: becomes computationally intractable for large maps!

→ Approaches exist that make EKF-SLAM amortized $O(n) / O(n^2) / O(n^2)$

D&C SLAM [Paz et al., 2006]
EKF-SLAM: Summary

• Convergence proof for linear case!

• Can diverge if nonlinearities are large (and the reality is nonlinear...)

• However, has been applied successfully in large-scale environments

• Approximations reduce the computational complexity
Approximations for SLAM

- Local submaps
  [Leonard et al.99, Bosse et al. 02, Newman et al. 03]

- Sparse links (correlations)
  [Lu & Milios 97, Guivant & Nebot 01]

- Sparse extended information filters
  [Frese et al. 01, Thrun et al. 02]

- Thin junction tree filters
  [Paskin 03]

- Rao-Blackwellisation (FastSLAM)
  [Murphy 99, Montemerlo et al. 02, Eliazar et al. 03, Haehnel et al. 03]