ME 597/747 - Lecture 7
Autonomous Mobile Robots

Instructor: Chris Clark
Term: Fall 2004

Figures courtesy of Siegwart & Nourbakhsh
Navigation Control Loop

- Prior Knowledge
  - Perception
  - Localization
- Operator Commands
  - Cognition
  - Motion Control
Localization: Outline

1. The Kalman Filter
   1. KF for fusing multiple measurements
   2. KF for fusing measurements over time
   3. KF for Robot Motion Estimation

2. Kalman Filter Localization
3. SLAM
4. Stochastic Maps
The Kalman Filter

- The Kalman Filter is used by engineers to fuse different measurements optimally, assuming they are gaussian distributions.
- For localization, it is used to fuse the encoder measurements (Prediction Step) with range sensors (Correction Step) in an optimal manner.
Consider the fusion of two measurements

- \( q_1 \) with variance \( \sigma_1^2 \)
- \( q_2 \) with variance \( \sigma_2^2 \)

The fusion is framed as a weighted least squares optimization problem with cost

\[
S = \sum_i w_i (\hat{q} - q_i)^2
\]

Minimize cost with \( \frac{\delta S}{\delta \hat{q}} = 0 \) to obtain:

\[
\hat{q} = q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} (q_2 - q_1)
\]
The Kalman Filter: Multiple Measurement Fusion

- This equation can be rewritten as a function of the Kalman Gain $K$:

$$\hat{q} = q_1 + K (q_2 - q_1)$$

- The Kalman gain in this case (1D) is:

$$K = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$
The Kalman Filter: Multiple Measurement Fusion

- The variance of $q$ can be calculated from two measurement variances $\sigma_1$ and $\sigma_2$

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

- Note that the new variance is less than previous two measurement variances.
The Kalman Filter: Multiple Measurement Fusion

\[ f(q) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left( -\frac{(q - \mu)^2}{2\sigma^2} \right) \]
Localization: Outline

1. The Kalman Filter
   1. KF for fusing multiple measurements
   2. KF for fusing measurements over time
   3. KF for Robot Motion Estimation

2. Kalman Filter Localization

3. SLAM

4. Stochastic Maps
The Kalman Filter: Measurement Fusion over Time

- The Kalman Filter can be used to fuse measurements over time.
- The measurements are used to estimate some state $x$.
- The new measurement $z_{k+1}$ at time step $k+1$ is fused with the previous state estimate $\hat{x}_k$ at step $k$.
- The first measurement $(q_1, \sigma_1)$ is replaced by the previous state estimate measurement $(x_k, \sigma_k)$.
- The second measurement $(q_2, \sigma_2)$ is replace by the new state measurement $(z_{k+1}, \sigma_z)$. 
The Kalman Filter: Measurement Fusion over Time

- The notation for state estimate update is:
  \[
  \hat{x}_{k+1} = \hat{x}_k + K_{k+1} \left( z_{k+1} - \hat{x}_k \right)
  \]

- Where the Kalman Gain is updated as:
  \[
  K_{k+1} = \frac{\sigma_k^2}{\sigma_k^2 + \sigma_z^2}
  \]
Localization: Outline

1. The Kalman Filter
   1. KF for fusing multiple measurements
   2. KF for fusing measurements over time
   3. KF for Robot Motion Estimation

2. Kalman Filter Localization

3. SLAM

4. Stochastic Maps
The Kalman Filter: Robot Motion Estimation

- For robot motion estimation, we don’t just fuse measurements over time.
- At each time step $k+1$, we fuse the robots predicted motion estimate $\hat{\mathbf{x}}_{k+1}'$ with new measurements $\mathbf{z}_{k+1}$ to obtain an “optimal” estimate $x_{k+1}$. 
The Kalman Filter: Robot Motion Estimation

- This is an iterative process.
The Kalman Filter: Robot Motion Estimation

- **Motion Prediction:**
  - In predicting motion of a robot, consider ODE:
    \[
    \frac{dx}{dt} = u + w \quad u = \text{velocity} \\
    \sigma^2 \quad w = \text{noise}
    \]

- The state estimate can be updated based on the last robot position \((\hat{x}_k, \sigma_k^2)\)
  \[
  \hat{x}'_{k+1} = \hat{x}_k + u [t_{k+1} - t] \\
  \sigma'_{k+1}^2 = \sigma_k^2 + \sigma_w^2 [t_{k+1} - t]
  \]
The Kalman Filter: Robot Motion Estimation

- The straight addition of variances increases overall variance
The Kalman Filter: Robot Motion Estimation

- Motion Correction
  - Can fuse sensor measurements $z_{k+1}$ with motion prediction $\hat{x}'_{k+1}$:
    $$\hat{x}_{k+1} = \hat{x}'_{k+1} + K_{k+1} \left( z_{k+1} - \hat{x}'_{k+1} \right)$$
  - Where the Kalman Gain is updated as:
    $$K_{k+1} = \frac{\sigma'_{k+1}^2}{\sigma'_{k+1}^2 + \sigma_z^2}$$
Localization: Outline

1. The Kalman Filter
2. Kalman Filter Localization
3. SLAM
4. Stochastic Maps
Kalman Filter Localization

- Algorithm:
  1. Robot Position Prediction
  2. Observation
  3. Measurement Prediction
  4. Matching
  5. Estimation: Applying the KF
Kalman Filter Localization

1. Prediction

- The robot’s position at time step \( k+1 \) is predicted based on its old location (time step \( k \) ) and its movement due to the control input \( u_k \):

\[
\hat{x}'_{k+1} = f (\hat{x}_k, u_k)
\]

\[
\Sigma'_{x,k+1} = \nabla_x f \Sigma_{x,k} \nabla_x f^T + \nabla_u f \Sigma_{u,k} \nabla_u f^T
\]

- Where \( f \) is the odometry function
Kalman Filter Localization: 1. Prediction

- For example:
  \[ \hat{x}'_{k+1} = \hat{x}_k + u_k \]
Kalman Filter Localization: 2. Observation

- The second step is to obtain the observation $Z_{k+1}$ (measurements) from the robot’s sensors at the new location at time $k+1$.
- The observation usually consists of a set $n_0$ of single observations $z_{j,k+1}$ extracted from the different sensors signals. It can represent raw data scans as well as features like lines, doors or any kind of landmarks.
Kalman Filter Localization: 2. Observation

- The parameters of the targets are usually observed in the sensor frame \( \{S\} \).
  - *Therefore the observations have to be transformed to the world frame \( \{W\} \)*
  - Or
  - *The measurement prediction have to be transformed to the sensor frame \( \{S\} \).*
Kalman Filter Localization: 2. Observation

\[ z_{j,k+1} = \sum R_{k+1} \]

\[ \Sigma_{R,k+1} = \begin{bmatrix} \sigma_{\alpha \alpha} & \sigma_{\alpha r} \\ \sigma_{r \alpha} & \sigma_{rr} \end{bmatrix} \]
Kalman Filter Localization: 3. Measurement Prediction

- We use the predicted robot position and the map $M(k)$ to generate multiple predicted observations $z_t$.
- They are transformed into the sensor frame
  \[ \hat{z}_{i,k+1} = h_i (z_t, \hat{x}'_{k+1}) \]
- We can define the measurement prediction as the set containing all $n_i$ predicted observations
  \[ Z_{k+1} = \{ \hat{z}_{i,k+1} \mid (1 \leq i \leq n_i) \} \]
Kalman Filter Localization: 3. Measurement Prediction

- For prediction, only the walls that are in the field of view of the robot are selected.
Kalman Filter Localization: 3. Measurement Prediction

- The generated measurement predictions have to be transformed to the robot frame \( \{R\} \)
- According to the figure in previous slide the transformation is given by

\[
\hat{Z}_{i,k+1} = R \begin{bmatrix} \alpha_{t,i} \\ r_{t,i} \end{bmatrix} \\
= h_i (z_t, \hat{x}'_{k+1}) \\
= \begin{bmatrix} wa_{t,i} - w\theta_{k+1} \\ wr_{t,i} - wx_{k+1} \cos(wa_{t,i}) - wy_{k+1} \sin(wa_{t,i}) \end{bmatrix}
\]
Kalman Filter Localization: 4. Matching

- Assignment from observations $z_{j,k+1}$ (gained by sensors) to the targets $z_t$ (stored in map)
- For each measurement prediction for which an corresponding observation is found we calculate the innovation:

$$v_{ij,k+1} = z_{j,k+1} - \hat{z}_{i,k+1}$$
$$= z_{j,k+1} - h_i (z_t, \hat{x}_{k+1})$$
Kalman Filter Localization: 4. Matching

- The innovation covariance found by applying the error propagation law with the covariance of the measurement $\Sigma_{R,i}$:
  \[ \Sigma_{IN,ij,k+1} = \nabla h_i \Sigma'_{x,k+1} \nabla h_i^T + \Sigma_{R,i,k+1} \]

- The validity of the correspondence between measurement and prediction can e.g. be evaluated through the Mahalanobis distance:
  \[ v_{ij,k+1}^T \Sigma_{IN,ij,k+1}^{-1} v_{ij,k+1} \leq g^2 \]
Kalman Filter Localization: 4. Matching
Kalman Filter Localization: 5. Estimation

- Update of robot’s position estimate
  \[ \hat{x}_{k+1} = \hat{x}'_{k+1} + K_{k+1} v_{k+1} \]

- The associated variance
  \[ \Sigma_{x,k+1} = \Sigma'_{x,k+1} - K_{k+1} \Sigma_{lN,K+1} K^T_{k+1} \]

- Kalman Gain:
  \[ K_{k+1} = \Sigma'_{x,k+1} \nabla h^T \Sigma_{lN,k+1}^{-1} \]
Kalman Filter Localization: 5. Estimation

- Kalman filter estimation of the new robot position:
  - By fusing the prediction of robot position (magenta) with the innovation gained by the measurements (green) we get the updated estimate of the robot position (red)
Localization: Outline

1. The Kalman Filter
2. Kalman Filter Localization
3. SLAM
4. Stochastic Maps
SLAM: Simultaneous Localization And Mapping

- Starting from an arbitrary initial point, a mobile robot should be able to autonomously explore the environment with its on board sensors, gain knowledge about it, interpret the scene, build an appropriate map and localize itself relative to this map.
SLAM: Problems

- Trying to reduce uncertainty

position of robot $\rightarrow$ position of wall

position of wall $\rightarrow$ position of robot
SLAM: Problems

- Cyclic Environment
  - Small local error accumulates to arbitrary large global errors.
  - This is usually irrelevant for navigation, but global error does matter when closing loops.
SLAM: Problems

- Dynamic Environments

e.g. disappearing cupboard
SLAM: Localization Only
SLAM: Localization and Mapping
SLAM

- Map Representation
  - $M$ is a set $n$ of probabilistic feature locations
  - Each feature is represented by the covariance matrix $\Sigma_t$ and an associated credibility factor $c_t$

$$M = \{ \hat{Z}_t, \Sigma_t, c_t \mid (1 \leq t \leq n) \}$$
Map representation

- $c_t$ is between 0 and 1 and quantifies the belief in the existence of the feature in the environment

\[ c_t(k) = 1 - \exp(- \frac{n_s}{a} + \frac{n_u}{b}) \]

- $a$ and $b$ define the learning and forgetting rate and $n_s$ and $n_u$ are the number of matched and unobserved predictions up to time $k$, respectively.
Localization: Outline

1. The Kalman Filter
2. Kalman Filter Localization
3. SLAM
4. Stochastic Maps
   1. Representation
   2. Example
   3. Spatial Relationships
   4. Map Building
SLAM: Stochastic Maps

- Use probability distributions to model all objects in the environment
  - Map features
  - Robot states

- Refer to:
  - Smith, Self and Cheeseman’s paper on “Estimating Uncertain Spatial Relationships in Robotics”
SLAM: Stochastic Maps

- Representation
  - For single object (e.g. mobile robot), represent state $x$ of object as first two moments of probability distribution
    
    Mean: $\hat{x} = E(x)$
    
    Variance: $C(x) = E( (x - \hat{x})(x - \hat{x})^T)$
  
  - For a mobile robot:
    $$\hat{x} = \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \end{pmatrix}, \quad C(x) = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 \end{pmatrix}$$
SLAM: Stochastic Maps

- Representation
  - For many objects, represent states in the stacked vector $X$:
    \[
    X = \begin{pmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n
    \end{pmatrix}, \quad
    C(X) = \begin{pmatrix}
    C(x_1)^2 & C(x_1, x_2) & \cdots \\
    C(x_1, x_2) & C(x_2)^2 & \cdots \\
    \vdots & \vdots & \ddots \\
    C(x_1, x_n) & C(x_2, x_n) & \cdots
    \end{pmatrix}
    \]
SLAM: Example – 1. Sense Object 1

Obj1_i
Robot_i
WORLD_i

Obj1_R
Robot_R
WORLD_R
SLAM: Example – 2. Move

Obj1 \_i \quad \text{Robot}_i \quad \text{Obj1}_R

\text{WORLD}_i \quad \text{WORLD}_R
SLAM: Example – 3. Sense Object 2
SLAM: Example – 4. Sense Object 1 again
SLAM: Example – 5. Update Estimate
SLAM: Stochastic Maps

- We can use Spatial Relationships to update the stochastic map
  - Compounding
  - Inverse
  - Composites
SLAM: Stochastic Maps

- **Compound Relationship**
  - Given two spatial relationships, $x_{ij}$ and $x_{jk}$, we can compute the resultant relationship $x_{ik}$
    
    $$x_{ik} = x_{ij} \oplus x_{jk} = \begin{pmatrix}
    x_{jk} \cos \theta_{ij} - y_{jk} \sin \theta_{ij} + x_{ij} \\
    x_{jk} \sin \theta_{ij} + y_{jk} \cos \theta_{ij} + y_{ij} \\
    \theta_{ij} + \theta_{jk}
    \end{pmatrix}$$

- Use first order estimate of mean:
  
  $$\hat{x}_{ik} \approx \hat{x}_{ij} \oplus \hat{x}_{jk}$$
SLAM: Stochastic Maps

- Compound Relationship
  - The first order estimate of the covariance is
    \[ C(x_{ik}) \approx J_\otimes \begin{pmatrix} C(x_{ij}) & C(x_{ij}, x_{jk}) \\ C(x_{ij}, x_{jk}) & C(x_{jk}) \end{pmatrix} J_\otimes^T \]
  
    \[ J_\otimes = \frac{\delta x_{ik}}{\delta(x_{ij}, x_{jk})} \]
SLAM: Stochastic Maps

- Inverse Relationship
  - Given the spatial relationship $x_{ij}$, we can compute the inverse relationship $x_{ji}$
    
    $$x_{ji} = \Theta x_{ij} = \begin{pmatrix}
    -x_{ij} \cos \theta_{ij} - y_{ij} \sin \theta_{ij} \\
    x_{ij} \sin \theta_{ij} - y_{ij} \cos \theta_{ij} \\
    -\theta_{ij}
    \end{pmatrix}$$

- Use first order estimate of mean:
  
  $$\hat{x}_{ji} \approx \Theta \hat{x}_{ij}$$
SLAM: Stochastic Maps

- Inverse Relationship
  - The first order estimate of the covariance is
  \[
  C(x_{ji}) \approx J_\theta \ C(x_{ij}) \ J_\theta^T
  \]

\[
J_\theta = \frac{\delta x_{ji}}{\delta x_{ij}}
\]
SLAM: Stochastic Maps

- Composite Relations
  - Given the spatial relationship $x_{ij}$, we can compute the inverse relationship $x_{ji}$

$$x_{il} = x_{ij} \oplus x_{jl} = x_{ij} \oplus (x_{jk} \oplus x_{kl}) = x_{ik} \oplus x_{kl} = (x_{ij} \oplus x_{jk}) \oplus x_{kl}$$
SLAM: Building Stochastic Maps

- Iterate on
  - Movement
  - New Spatial Information
- Use Kalman Filter at each iteration
Movement

The robot makes an uncertain move $y_R$

$$y_{Rob} = u_{Rob} + w_{Rob}$$

This motion results in a new world state

$$x'_{Rob} = x_{Rob} \oplus y_{Rob}$$

This requires only modifying a small portion of the map (e.g. the element of X corresponding to the robot).
New Spatial Information

- There are two categories of new information: a new object or a new constraint on an existing object.
- If a new object is added to the map, the additional entries must be added to $X$ and $C(X)$.
- If a new measurement is gained for an existing object, we can model measurement $z$ as:

$$z = h(x) + v$$
SLAM: Building Stochastic Maps

- New Spatial Information
  - Looking at the first moments
    \[ \hat{z} \approx h(\hat{x}) \]
    \[ C(z) = \nabla h_x C(x) \nabla h_x + C(v) \]
  - A simple function \( h(x) \) might be:
    \[ h(x) = x_{ij} = \otimes x_i \oplus x_j \]
SLAM: Building Stochastic Maps

- Kalman Filter
  - We can use the KF to update the state estimates
    \[
    \hat{x}_k = \hat{x}'_k + K_k \left[ z_k - h_k (\hat{x}'_k) \right]
    \]
    \[
    C(x_k) = C(x'_k) - K_k \nabla h_x C(x'_k)
    \]
SLAM: Example

- Time $t_0$
  - Before the robot does anything, the initial position in both frames of reference are equated at 0.
  - The covariance is also 0 representing no uncertainty

\[
\hat{x} = [\hat{x}_R] = [0] \\
C(x) = [C(x_R)] = [0]
\]
SLAM: Example

- Time $t_1$
  - The robot senses Object 1, and the object mean is added to the state vectors.

\[
\hat{x} = \begin{pmatrix} \hat{x}_R \\ \hat{x}_1 \end{pmatrix} = \begin{pmatrix} 0 \\ \hat{z}_1 \end{pmatrix}
\]

\[
C(x) = \begin{bmatrix} C(x_R) & C(x_R, x_1) \\ C(x_1, x_R) & C(x_1) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & C(z_1) \end{bmatrix}
\]
SLAM: Example

- Time $t_2$
  - The robot moves to another location, where the relative motion is given by $y_R$

$$\hat{x} = \begin{pmatrix} \hat{x}_R \\ \hat{x}_1 \end{pmatrix} = \begin{pmatrix} \hat{y}_R \\ \hat{z}_1 \end{pmatrix}$$

$$C(x) = \begin{bmatrix} C(x_R) & C(x_R, x_1) \\ C(x_1, x_R) & C(x_1) \end{bmatrix} = \begin{bmatrix} C(y_R) & 0 \\ 0 & C(z_1) \end{bmatrix}$$
**SLAM: Example**

- **Time** $t_3$
  - The robot now senses a new object from the new location

\[
\hat{x} = \begin{pmatrix}
\hat{x}_R \\
\hat{x}_1 \\
\hat{x}_2
\end{pmatrix} = \begin{pmatrix}
\hat{y}_R \\
\hat{z}_1 \\
\hat{y}_R \oplus \hat{z}_2
\end{pmatrix}
\]

\[
C(x) = \begin{bmatrix}
C(x_R) & C(x_R, x_1) & C(x_R, x_2) \\
C(x_1, x_R) & C(x_1) & C(x_1, x_2) \\
C(x_2, x_R) & C(x_2, x_1) & C(x_2)
\end{bmatrix} = \begin{bmatrix}
C(y_R) & 0 & C(y_R)J_1^T \\
0 & C(z_1) & 0 \\
J_1 \oplus C(y_R) & 0 & C(x_2)
\end{bmatrix}
\]
SLAM: Example

- Time $t_4$
  - The robot now senses the first object again from the new location
  - Use the Kalman Filter to update the estimate of Object 1
SLAM: Mining

Courtesy of S. Thrun