EKF SLAM

D.A. Forsyth, UIUC
The SLAM Problem

- SLAM stands for simultaneous localization and mapping
- The task of building a map while estimating the pose of the robot relative to this map
- Why is SLAM hard?
  Chicken-or-egg problem:
  - a map is needed to localize the robot and a pose estimate is needed to build a map

From Burgard et al slides
Alternative view of SLAM

• We already know we can do it
  • for example
    • do the matrix factorization stuff incrementally
    • visual odometry then triangulate
  • BUT
    • that doesn’t take uncertainty into account

• What we’re doing now is
  • wrapping an EKF (other filter) around ideas we’ve seen before
Simplest case

• Vehicle moves in 2D
• Each measurement is
  • a 2D measurement
  • of position of a known beacon in vehicle coords
    • (i.e. we know which measurement corresponds to which 3D point)
State

\[ \mathbf{x} = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \ldots \\ \mathcal{L}_n \end{bmatrix} \]

All landmark positions in original coordinate frame

Position and orientation of the robot

Landmark 1 position in OCF
A general movement model

\[
\begin{bmatrix}
  x \\
y \\
\theta
\end{bmatrix}
\rightarrow
\begin{bmatrix}
x + R(\sin(\theta + \Delta\theta) - \sin(\theta)) \\
y - R(\cos(\theta + \Delta\theta) - \cos(\theta)) \\
\theta + \Delta\theta
\end{bmatrix}
\]

THIS ISN’T LINEAR!

\[v_t \text{ = velocity}\]
\[\omega_t \text{ = rotational velocity}\]
State update

\[ x_i = f(x_{i-1}, n) \]

\[
\begin{bmatrix}
R \\
L_1 \\
L_2 \\
\vdots \\
L_n
\end{bmatrix}
= \begin{bmatrix}
R \\
L_1 \\
C \ldots \\
L_1 \\
L_n
\end{bmatrix}
\]

- The vehicle moves, as above;
  - but the landmarks don’t move
  - and there isn’t any noise

\[
\begin{bmatrix}
R \\
L_1 \\
L_2 \\
\vdots \\
L_n
\end{bmatrix}
\rightarrow
\begin{bmatrix}
h(R) + \xi \\
L_1 \\
L_2 \\
\vdots \\
L_n
\end{bmatrix}
\]
Recall: The extended Kalman filter

- Linearize:

\[ x_i = f(x_{i-1}, n) \]

\[ F_x = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \cdots \\ \cdots & \frac{\partial f_i}{\partial x_j} & \cdots \\ \cdots & \cdots & \frac{\partial f_i}{\partial x_j} & \cdots \end{bmatrix} \]

\[ F_n = \begin{bmatrix} \frac{\partial f_1}{\partial n_1} & \cdots & \cdots \\ \cdots & \frac{\partial f_i}{\partial n_j} & \cdots \\ \cdots & \cdots & \frac{\partial f_i}{\partial n_j} & \cdots \end{bmatrix} \]

Posterior covariance of \( x_{i-1} \)

\[ x_i \sim N(f(\bar{x}_{i-1}^+, 0), F_x \Sigma_{i-1}^{+} F_x^T + F_n \Sigma_{n,i} F_n^T) \]

Noise covariance
Measuring position

- Landmark is at:
  - in world coordinate system

- We record position in vehicle’s frame:

\[
\begin{bmatrix}
u \\
v
\end{bmatrix}
\]

\[
\begin{bmatrix}
x_v \\
y_v
\end{bmatrix} = \mathcal{R}_{-\theta} \begin{bmatrix}
(u - x) \\
(v - y)
\end{bmatrix}
\]

\textbf{Observation}

\textbf{THIS ISN’T LINEAR!}
Recall: The extended Kalman filter

- Linearize:

\[ y_i = g(x_i, n) \]

\[ G_x = \begin{bmatrix}
\frac{\partial g}{\partial x_1} & \cdots & \cdots \\
\cdots & \frac{\partial g}{\partial x_1} & \cdots \\
\cdots & \cdots & \frac{\partial g}{\partial x_1} & \cdots
\end{bmatrix} \]

\[ G_n = \begin{bmatrix}
\frac{\partial g}{\partial n_1} & \cdots & \cdots \\
\cdots & \frac{\partial g}{\partial n_1} & \cdots \\
\cdots & \cdots & \frac{\partial g}{\partial n_1} & \cdots
\end{bmatrix} \]

\[ y_i \approx \mathcal{N}(g(x_i, 0), G_x \Sigma_i G_x^T + G_n \Sigma_{m,i} G_n^T) \]
The extended kalman filter

Dynamic Model:

\[ x_i = f(x_{i-1}, n) \]
\[ y_i = g(x_i, n) \]

Start Assumptions: \( \bar{x}_0^- \) and \( \Sigma_0^- \) are known

Update Equations: Prediction

\[ x_i \sim \mathcal{N}(f(x_{i-1}, 0), \mathcal{F}_x \Sigma_{i-1}^+ \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T) \]

Update Equations: Correction

\[ \mathcal{K}_i = \Sigma_i^- \mathcal{G}_x^T \left[ \mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{n,i} \mathcal{G}_n^T \right]^{-1} \]
\[ \bar{x}_i^+ = \bar{x}_i^- + \mathcal{K}_i \left[ y_i - g(x_i^-, 0) \right] \]
\[ \Sigma_i^+ = [\text{Id} - \mathcal{K}_i \mathcal{G}_x] \Sigma_i^- \]
In principle, now easy

- Rather horrid from the point of view of complexity
  - looks like we have to invert a 3+N by 3+N matrix!

- **BUT**
  - F_x is much simpler than it might look
    - the landmarks do not move!
  - F_n ditto
    - there is no noise in the landmark updates - the landmarks are fixed

- **Outcome:**
  - We can deal with landmarks one by one
    - and so do many small matrix inversions rather than one large one
State update

\[ x_i = f(x_{i-1}, n) \]

\[ x = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \vdots \\ \mathcal{L}_n \end{bmatrix} \]

- The vehicle moves, as above;
  - but the landmarks don’t move
  - and there isn’t any noise

\[ \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \vdots \\ \mathcal{L}_n \end{bmatrix} \rightarrow \begin{bmatrix} h(\mathcal{R}) + \xi \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \vdots \\ \mathcal{L}_n \end{bmatrix} \]
State update, II

• BUT
  • $F_x$ is much simpler than it might look
    • the landmarks do not move!
  • $F_n$ ditto
    • there is no noise in the landmark updates - the landmarks are fixed

\[
F_x = \begin{bmatrix}
\frac{\partial f_R}{\partial R} & 0 \\
0 & \mathcal{I}
\end{bmatrix}
\]

\[
F_n = \begin{bmatrix}
\frac{\partial f_R}{\partial n} & 0 \\
0 & 0
\end{bmatrix}
\]

$N=$Number of landmarks
State update, III

- Imagine we have 2 landmarks

Recall EKF:

\[ x_i \sim \mathcal{N}(f(x_{i-1}, 0), \mathcal{F}_x \Sigma_{i-1}^+ \mathcal{F}_x^T + \mathcal{F}_n \Sigma_n, i \mathcal{F}_n^T) \]

\[
\mathcal{F}_x = \begin{bmatrix}
\mathcal{W} & 0 & 0 \\
0 & \mathcal{I} & 0 \\
0 & 0 & \mathcal{I}
\end{bmatrix}
\]

\[
\Sigma_{i-1}^+ = \begin{bmatrix}
A & B & C \\
B^T & D & E \\
C^T & E^T & \mathcal{F}
\end{bmatrix}
\]

\[
\mathcal{F}_x \Sigma_{i-1}^+ \mathcal{F}_x^T = \begin{bmatrix}
\mathcal{W} A \mathcal{W}^T & \mathcal{W} A & \mathcal{W} B \\
B^T \mathcal{W}^T & D & E \\
C^T \mathcal{W} & E^T & \mathcal{F}
\end{bmatrix}
\]

Notice fewer matrix multiplies!
State update, IV

- Imagine we have 2 landmarks

Recall EKF:

\[ \mathbf{x}_i \sim \mathcal{N}(f(\mathbf{x}_{i-1}, 0), \mathcal{F}_x \Sigma_{i-1}^+ \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T) \]

\[
\mathcal{F}_n = \begin{bmatrix}
\mathcal{V} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\Sigma_{n,i} = \begin{bmatrix}
\mathcal{H} & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
\mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T = \begin{bmatrix}
\mathcal{V} \mathcal{H} \mathcal{V}^T & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

Notice fewer matrix multiplies!
More simplifications

BUT
- \( G_x \) is much simpler than it might look
  - each set of measurements affected by only one landmark!

\[
\begin{array}{c|cc|c}
\multicolumn{1}{c}{} & \multicolumn{2}{|c|}{N} & N=\text{Number of landmarks} \\
\hline
3 & 2 & \hline
\frac{\partial O_1}{\partial R} & \frac{\partial O_1}{\partial L_1} & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial O_2}{\partial R} & 0 & \frac{\partial O_2}{\partial L_2} & 0 & 0 & 0 & 0 \\
\ldots & \frac{\partial O_N}{\partial R} & 0 & 0 & 0 & 0 & \frac{\partial O_N}{\partial L_N} \\
\end{array}
\]

\[ G_x = \begin{bmatrix}
\frac{\partial O_1}{\partial R} & \frac{\partial O_1}{\partial L_1} & 0 & 0 & 0 & 0 \\
\frac{\partial O_2}{\partial R} & 0 & \frac{\partial O_2}{\partial L_2} & 0 & 0 & 0 \\
\ldots \\
\frac{\partial O_N}{\partial R} & 0 & 0 & 0 & 0 & \frac{\partial O_N}{\partial L_N}
\end{bmatrix} \]
More simplifications

• BUT
  • $G_n$ is usually much simpler than it might look
    • noise is usually additive normal noise

• This means that the term: $G_n \sum_{n,i} G_n^T$
  • is actually a block diagonal matrix
Big simplification

• The nasty bit…

\[
\begin{bmatrix}
  \sum_i g_x^T g_x + \sum_{m,i} g_n^T g_n
\end{bmatrix}^{-1}
\]

• But notice **key point**
  • measurements interact only through the position/orientation of the vehicle
  • each measurement depends on only one landmark and pose of v.
  • OR measurements are conditionally independent conditioned on pose of v.
  • OR you could subdivide time and update measurements one by one
  • OR matrix $G_x$ has the sparsity structure above

• (the same point, manifesting in different ways)
Subdividing time...

- We receive measurements of landmarks in some order
  - a measurement of the position of landmark $i$ affects the whole state
    - because it changes your estimate of the pose of the vehicle
    - and that affects your estimate of state of every landmark
  - BUT
    - the change in estimate of pose depends ONLY on
      - pose
      - landmark $i$
Subdividing time…

- **Sequence**
  - repeat
    - move (so make predictions)
    - landmark 1 measurement arrives (update 1 and pose)
    - …
    - landmark N measurement arrives (update N and pose)
Steps in EKF

\[ K_i = \Sigma_i^{-} g_x^T \left[ g_x \Sigma_i^{-} g_x^T + g_n \Sigma_m, i g_n^T \right]^{-1} \]

\[ x_i^{+} = x_i^{-} + K_i \left[ y_i - g(x_i^{-}, 0) \right] \]

\[ \Sigma_i^{+} = \left[ I - K_i g_x \right] \Sigma_i^{-} \]
Steps in EKF

One measurement from one landmark!

\[ \mathcal{K}_i = \sum_i -G_x^T \left[ G_x \sum_i -G_x^T + G_n \sum_{m,i} G_n^T \right]^{-1} \]

\[ \mathbf{x}_i^+ = \mathbf{x}_i^- + \mathcal{K}_i \left[ y_i - g(\mathbf{x}_i^-, 0) \right] \]

\[ \Sigma_i^+ = [I - \mathcal{K}_i G_x] \Sigma_i^- \]
Why is SLAM a hard problem?

- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
- Pose error correlates data associations

From Burgard et al slides
In factorization language

- Which point in image i goes into which row of the matrix?
  - get that wrong enough often enough and you’re in trouble
- Obvious we can do something about this
  - eg assume we have OK reconstruction from frame 1..N-1
    - in frame N, estimate camera motion from
      - small number of reliable point correspondences + VO
      - shaft encoders, etc.
    - now sort out all other observations
      - eg map to the point that appears closest in predicted camera
Data Association Problem

- A data association is an assignment of observations to landmarks.
- In general there are more than $\binom{n}{m}$ (n observations, m landmarks) possible associations.
- Also called “assignment problem.”

From Burgard et al slides
Landmarks

• Which measurement comes from which landmark?
  • data association -
    • use some form of bipartite graph matching

• Idea:
  \[ \overline{X_i} \]
  • predicts landmark positions, vehicle position before obs
  • compute distances between all pairs of
    • predicted obs, real obs
    • bipartite graph matcher
    • OR greedy
Landmarks

• No measurement from a landmark?
  • structure of EKF means you can process landmarks one by one
    • that’s what all the matrix surgery was about
    • so don’t update that landmark

• How do we know no measurement from a landmark?
  • refuse to match if distance in greedy/bipartite is too big
  • other kinds of matching problem (color, features, etc)
Measuring distance and orientation

• Landmark is at:
  • in global coordinate system

• We record distance and heading:
  • measurement

\[
\begin{bmatrix}
  d \\
  \phi
\end{bmatrix} = \begin{bmatrix}
  \sqrt{(x-u)^2 + (y-v)^2} \\
  \text{atan}2(y-u, x-v) - \theta
\end{bmatrix}
\]

THIS ISN’T LINEAR!
A further trick: inverting measurement

- Example: measure distance and orientation to point

\[
\begin{bmatrix}
    u \\
    v
\end{bmatrix}
\]

point posn in world coods

\[
\begin{bmatrix}
    \phi \\
    d
\end{bmatrix}
= \begin{bmatrix}
    \sqrt{(x - u)^2 + (y - v)^2} \\
    \text{atan2}(y - u, x - v) - \theta
\end{bmatrix}
\]

vehicle posn in world coods

vehicle orientation in world coods

Observation
Range and bearing

Observation \[
\begin{bmatrix}
d \\
\phi
\end{bmatrix}
= \begin{bmatrix}
\sqrt{(x-u)^2 + (y-v)^2} \\
\text{atan2}(y-u, x-v) - \theta
\end{bmatrix}
\]

Landmark position

Vehicle state

\[
\begin{bmatrix}
u \\
v
\end{bmatrix} = \begin{bmatrix}
x + (d + \xi) \sin(\phi + \zeta + \theta) \\
y + (d + \xi) \cos(\phi + \zeta + \theta)
\end{bmatrix}
\]

These are measurements of landmark ONLY

Noise affecting measurements

Here use the current estimate of vehicle state
Bearing only (sketch)

- Cannot determine landmark in 2D from measurement
  - it’s on a line!
  - you must come up with a prior
    - after that, it’s easy
      - find mean posterior location, covariance
      - plug in
  - Big Issue
    - True prior should have infinite covariance
      - can’t work with that
      - so linearization may fail