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

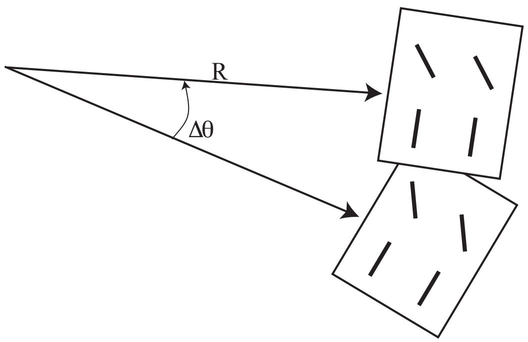
Position and orientation of the robot

$$\mathbf{x} = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \dots \\ \mathcal{L}_n \end{bmatrix}$$

Landmark 1 position in OCF

All landmark positions in original coordinate frame

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 = velocity
omega_t = rotational velocity

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{v_t}{\omega_t}(\sin(\theta + \omega_t \Delta t) - \sin(\theta)) \\ -\frac{v_t}{\omega_t}(\cos(\theta + \omega_t \Delta t) - \cos(\theta)) \\ \omega_t \Delta t \end{bmatrix}$$

State update

$$\mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{n})$$

$$\mathbf{x} = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \dots \\ \mathcal{L}_n \end{bmatrix}$$

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

$$\begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \dots \\ \mathcal{L}_n \end{bmatrix} \rightarrow \begin{bmatrix} h(\mathcal{R}) + \xi \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \dots \\ \mathcal{L}_n \end{bmatrix}$$

Recall: The extended Kalman filter

- Linearize:

$$\mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{n})$$

$$\mathcal{F}_x = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \cdots \\ \cdots & \frac{\partial f_i}{\partial x_j} & \cdots \end{bmatrix}$$

$$\mathcal{F}_n = \begin{bmatrix} \frac{\partial f_1}{\partial n_1} & \cdots & \cdots \\ \cdots & \frac{\partial f_i}{\partial n_j} & \cdots \end{bmatrix}$$

Posterior covariance of x_{i-1}

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

Noise covariance

Measuring position

- Landmark is at: $\left[\begin{array}{c} u \\ v \end{array}\right]$
  - in world coordinate system
- We record position in vehicle's frame:

Observation

$$\left[\begin{array}{c} x_v \\ y_v \end{array}\right] = \mathcal{R}_{-\theta} \left[\begin{array}{c} (u-x) \\ (v-y) \end{array}\right]$$

vehicle orientation in world coords

point posn in world coords

point posn in vehicle coords

vehicle posn in world coords

THIS ISN'T LINEAR!

Recall: The extended Kalman filter

- Linearize:

$$\mathbf{y}_i = g(\mathbf{x}_i, \mathbf{n})$$

$$\mathcal{G}_x = \begin{bmatrix} \frac{\partial g}{\partial x_1} & \dots & \dots \\ \dots & \frac{\partial g}{\partial x_1} & \dots \end{bmatrix}$$

$$\mathcal{G}_n = \begin{bmatrix} \frac{\partial g}{\partial n_1} & \dots & \dots \\ \dots & \frac{\partial g}{\partial n_1} & \dots \end{bmatrix}$$

$$\mathbf{y}_i \approx \mathcal{N}(g(\mathbf{x}_i, \mathbf{0}), \mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}_n^T)$$

Correction!

Dynamic Model:

$$\mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{n})$$

$$\mathbf{y}_i = g(\mathbf{x}_i, \mathbf{n})$$

Start Assumptions: $\overline{\boldsymbol{x}}_0^-$ and Σ_0^- are known

Update Equations: Prediction

$\overline{x}_i^-$ Σ_i^-

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

Update Equations: Correction

$$\begin{aligned} \mathcal{K}_i &= \Sigma_i^- \mathcal{G}_x^T \left[\mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}_n^T\right]^{-1} \\ \overline{\boldsymbol{x}}_i^+ &= \overline{\boldsymbol{x}}_i^- + \mathcal{K}_i \left[\mathbf{y}_i - g(\mathbf{x}_i^-, \mathbf{0})\right] \\ \Sigma_i^+ &= [Id - \mathcal{K}_i \mathcal{G}_x] \Sigma_i^- \end{aligned}$$

The extended kalman filter

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

$$\mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{n})$$

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

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

$$\begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \ldots \\ \mathcal{L}_n \end{bmatrix} \rightarrow \begin{bmatrix} h(\mathcal{R}) + \xi \\ \mathcal{L}_1 \\ \mathcal{L}_2 \\ \ldots \\ \mathcal{L}_n \end{bmatrix}$$

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

N=Number of landmarks

$$\mathcal{F}_x = \begin{bmatrix} \overbrace{\frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}}}^{3} & \overbrace{0}^{N} \\ 0 & \mathcal{I} \end{bmatrix}$$

$$\mathcal{F}_n = \begin{bmatrix} \frac{\partial f_{\mathcal{R}}}{\partial \mathbf{n}} & 0 \\ 0 & 0 \end{bmatrix}$$

State update, III

- Imagine we have 2 landmarks

Recall EKF: $$\mathbf{x}_i \sim \mathcal{N}(f(\mathbf{x}_{i-1}, \mathbf{0}), \mathcal{F}_x \Sigma^+_{i-1} \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T$$

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

$$\Sigma^+_{i-1} = \begin{bmatrix} \mathcal{A} & \mathcal{B} & \mathcal{C} \\ \mathcal{B}^T & \mathcal{D} & \mathcal{E} \\ \mathcal{C}^T & \mathcal{E}^T & \mathcal{F} \end{bmatrix}$$

$$\mathcal{F}_x \Sigma^+_{i-1} \mathcal{F}_x^T = \begin{bmatrix} \mathcal{W}\mathcal{A}\mathcal{W}^T & \mathcal{W}\mathcal{A} & \mathcal{W}\mathcal{B} \\ \mathcal{B}^T\mathcal{W}^T & \mathcal{D} & \mathcal{E} \\ \mathcal{C}^T\mathcal{W} & \mathcal{E}^T & \mathcal{F} \end{bmatrix}$$

Notice fewer matrix multiplies!

State update, IV

- Imagine we have 2 landmarks

Recall EKF: $\mathbf{x}_i \sim \mathcal{N}(f(\mathbf{x}_{i-1}, \mathbf{0}), \mathcal{F}_x \Sigma^+_{i-1} \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T$

$$\mathcal{F}_n = \begin{bmatrix} \mathcal{V} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \Sigma_{n,i} = \begin{bmatrix} \mathcal{H} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

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

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!

N N=Number of landmarks

3 2

$$\mathcal{G}_x = \begin{bmatrix} \frac{\partial \mathcal{O}_1}{\partial \mathcal{R}} & \frac{\partial \mathcal{O}_1}{\partial \mathcal{L}_1} & 0 & 0 & 0 & 0 \\ \frac{\partial \mathcal{O}_2}{\partial \mathcal{R}} & 0 & \frac{\partial \mathcal{O}_2}{\partial \mathcal{L}_2} & 0 & 0 & 0 \\ \dots & & & & & \\ \frac{\partial \mathcal{O}_N}{\partial \mathcal{R}} & 0 & 0 & 0 & 0 & \frac{\partial \mathcal{O}_N}{\partial \mathcal{L}_N} \end{bmatrix}$$

2 N

More simplifications

- BUT
  - G_n is usually much simpler than it might look
    - noise is usually additive normal noise
    - This means that the term: $\mathcal{G}_n \Sigma_{n,i} \mathcal{G}_n^T$
      - is actually a block diagonal matrix

Big simplification

- The nasty bit…

$$\left[\mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}_n^T\right]^{-1}$$

- 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

$$\mathcal{K}_i = \Sigma_i^- \mathcal{G}_x^T \left[\mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}_n^T \right]^{-1}$$

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

$$\Sigma_i^+ = \left[\mathcal{I} - \mathcal{K}_i \mathcal{G}_x \right] \Sigma_i^-$$

One measurement from one landmark!

Steps in EKF

3+2Nx2   3+2N x 2   2 x 2

$$\mathcal{K}_i = \Sigma_i^- \mathcal{G}_x^T \left[\mathcal{G}_x \Sigma_i^- \mathcal{G}_x^T + \mathcal{G}_n \Sigma_{m,i} \mathcal{G}_n^T\right]^{-1}$$

Notice:
Inverting only a small matrix

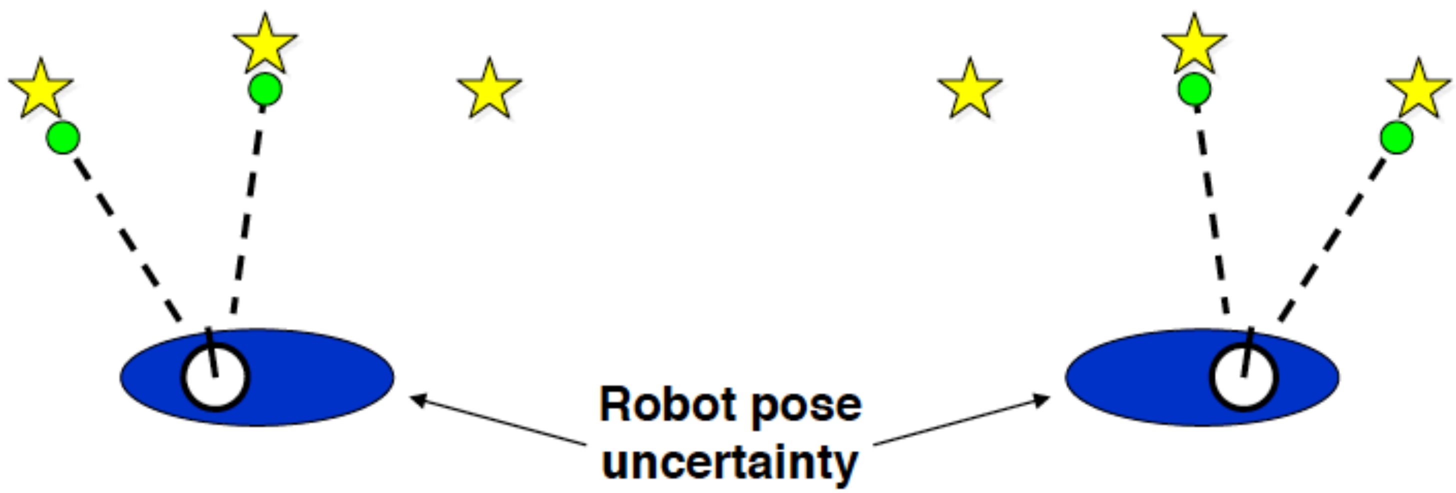
3+2Nx2   2 x 1

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

Notice:
But affecting the whole state!

$$\Sigma_i^+ = \left[\mathcal{I} - \mathcal{K}_i \mathcal{G}_x\right] \Sigma_i^-$$

Why is SLAM a hard problem?



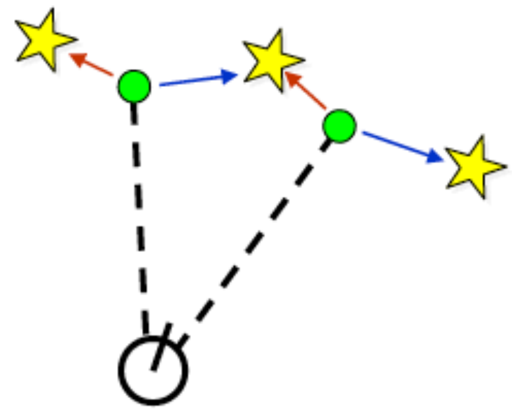
- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
- Pose error correlates data associations

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: $\overline{\mathbf{x}}_i^-$
        - 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

$$\begin{bmatrix} u \\ v \end{bmatrix}$$

- We record distance and heading:
  - measurement

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

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 coords

vehicle posn in world coords

Observation

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

vehicle orientation in world coords

Range and bearing

Landmark position

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

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