EKF-SLAM

D.A. Forsyth,

University of Illinois at Urbana Champaign

Core idea

- Observe the world from a moving platform
- Build:
  - a map of the world
  - an estimate of pose in that map
- Online:
  - notice that new observations update:
    - estimate of pose
    - estimate of map
- Filtering problem

Some more details

- With two cameras and some calibration:
  - we can recover the position of 3D points
    - in the vehicle's coordinate system
      - see triangulation movie, etc.

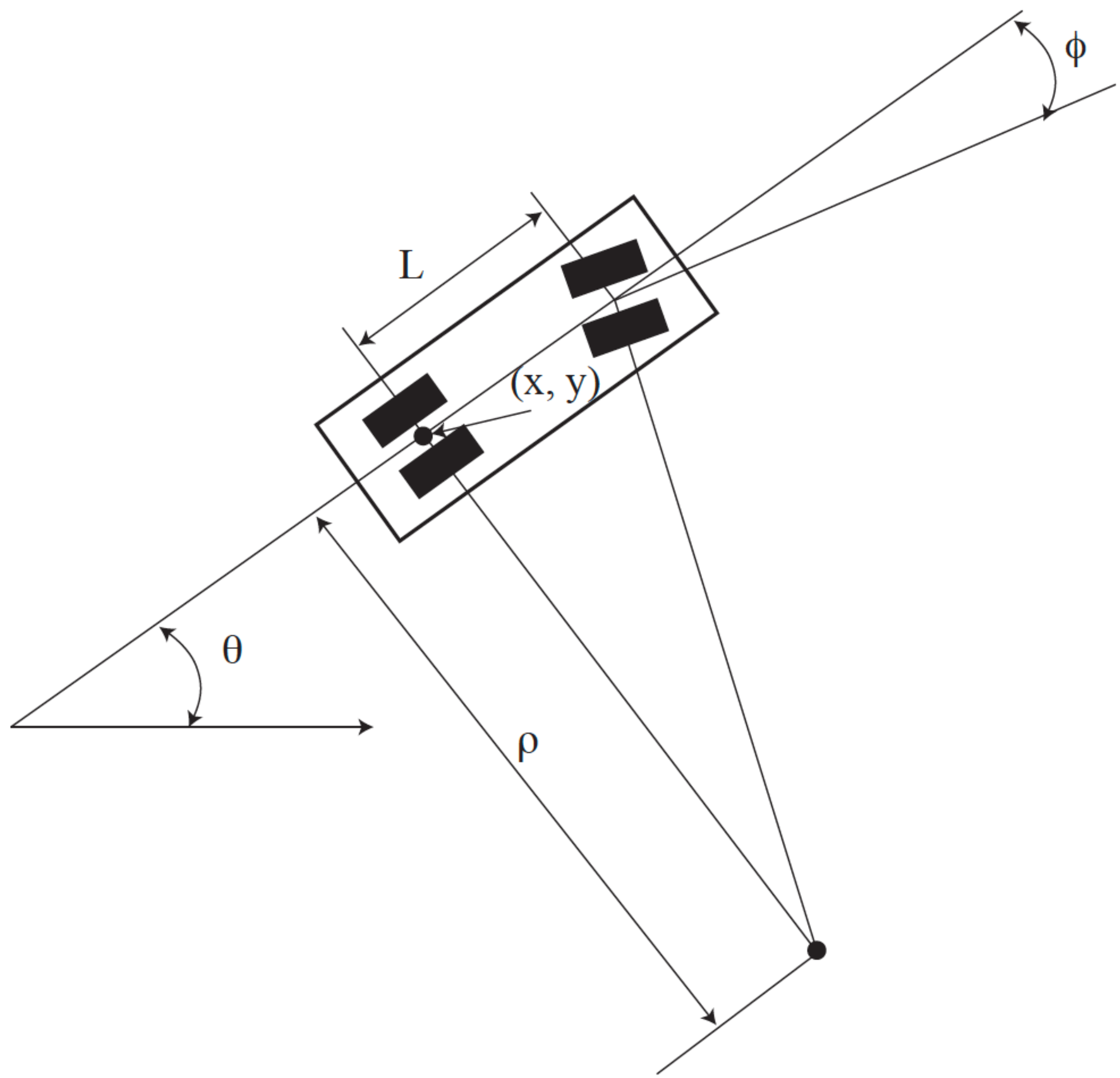
- Together with an EKF, we can use this to recover
  - points in world coordinates (a map)
  - vehicle location

- In fact, we can do all this with one camera
  - with some minor care

Simple case

- Vehicle is car, 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)

Notation for dynamics



Dynamics

- Write
  - s – speed along the midline
  - r - rate of change of steering angle
- Get:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} s\cos\theta \\ s\sin\theta \\ \frac{s}{L}\tan r \end{bmatrix}$$

Derivation at: https://msl.cs.uiuc.edu/planning/node658.html

State

Position and orientation of the robot; speed and rate of change of steering

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

Landmark 1 position in OCF

All landmark positions in original coordinate frame

State update

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

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

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

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

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:
  - in world coordinate system

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

- We record position in vehicle's frame:

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

Observation

vehicle orientation in world coords

point posn in world coords

point posn in vehicle coords

vehicle posn in world coords

THIS ISN'T LINEAR!

The steps, EKF:

Have:

$$\overline{X}^+_{i-1} \qquad \Sigma^+_{i-1}$$

Construct:

$$\bar{X}_i^- = f_i(\bar{\mathbf{x}}^+_{i-1}, \mathbf{0}) \qquad \Sigma_i^- = \mathcal{F}_x \Sigma^+_{i-1} \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T$$

Measurement arrives: $\mathbf{y}_i = g_i(\mathbf{x}_i, \mathbf{n})$

Now construct:

$$\bar{X}_i^+ = \bar{X}_i^- + \mathcal{K}_i \left[\mathbf{y}_i - g_i(\bar{X}_i^-, \mathbf{0})\right] \qquad \Sigma_i^+ = \left[Id - \mathcal{K}_i \mathcal{G}_x\right] \Sigma_i^-$$

Where:

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

In principle, now easy

- Rather horrid from the point of view of complexity
    - looks like we have to invert a 3+2N by 3+2N 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

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

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

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

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

State update, II

$$\Sigma_i^- = \mathcal{F}_x \Sigma_{i-1}^+ \mathcal{F}_x^T + \mathcal{F}_n \Sigma_{n,i} \mathcal{F}_n^T$$

- 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

3 2 N

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

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

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

$$\Sigma_i^+ = [Id - \mathcal{K}_i \mathcal{G}_x]\, \Sigma_i^-$$

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

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

N (N=Number of landmarks)

Column widths: 3, 2. Rows: 2 N.

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

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…

- Receive observations in some order
  - 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 pose and so all based on 1)
    - …
    - landmark N measurement arrives (update pose and so all based on N)

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!

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

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

New landmarks

- Sequence
  - repeat
    - move (so make predictions)
    - landmark 1 measurement arrives (update pose and so all based on 1)
    - ...
    - landmark N measurement arrives (update pose and so all based on N)
    - check if there is a new landmark
      - if so
        - initialize landmark position and covariance
          - conditioned on current state and measurement
        - process from now on (we have N+1 landmarks)

Variants

- More complex vehicle dynamics
- Different measurement process
    - one camera
    - LIDAR
    - SONAR (v. hard)
    - photo-consistency constraints (direct methods)
- Multiple processes at different timescales
    - fast:
        - estimate configuration, landmarks
    - slow:
        - refine estimates (something like bundle adjustment)
        - take more local image data into account

Resources

- Very nice collection of links at:

https://github.com/Taeyoung96/SLAM-Resources-for-Beginner?tab=readme-ov-file

- Challenging drone racing dataset at:

https://fpv.ifi.uzh.ch

- Drone racing team at:

https://adr.utias.utoronto.ca

- Many cases at:

https://silenceoverflow.github.io/Awesome-SLAM/#LSurvey