INFO0948
Localization

Renaud Detry

University of Liège, Belgium

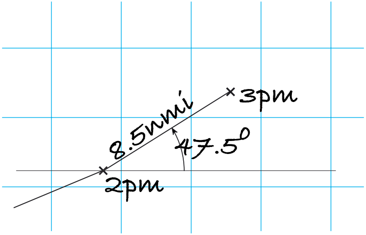
Updated by L. Wehenkel on March 1, 2017

These slides are based on Chapter 6 and Appendix H of the textbook *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* by Peter Corke, published by Springer in 2011.

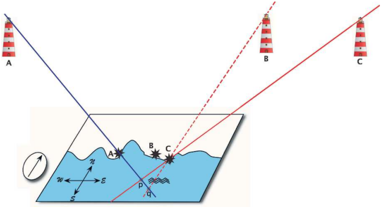
Plan

Dead Reckoning, Maps and Landmarks

Dead Reckoning:
Estimation of location based on estimated speed, direction and time of travel.

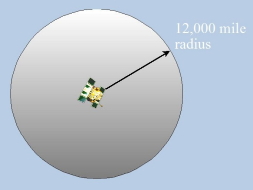


Using a compass, a map, and landmarks: (flashing) lighthouses, stars, islands, ...

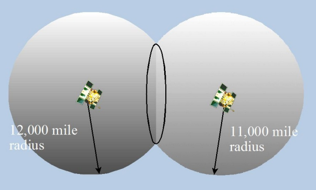


GPS

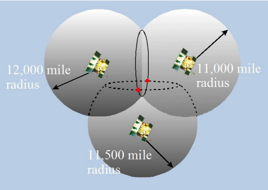
Distance to one satellite: sphere.



Distance to two satellites: circle.



Distance to three satellites: point.



A fourth satellite is required for clock sync.

The European Galileo system has 5 satellites up and running. The remaining 24 or 25 are expected to launch before 2019.

http://www.montana.edu/gps/understd.html

Means Of Robot Localization

GPS-like systems (GPS, ultrasound beacons, ...): useful when available, but not always available, and sometimes not sufficiently precise.

Odometry: for instance, integrating the rotation speed of a wheel.

Using a map and observing landmarks.

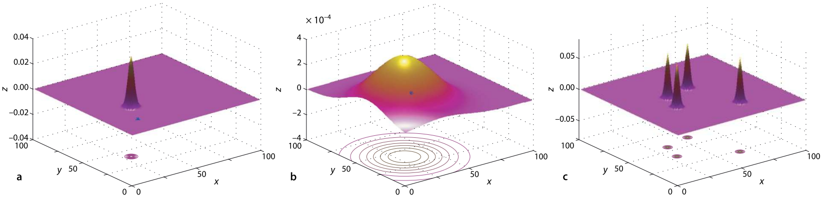
Plan

The Localization Problem

We denote by x the true but unknown robot position.

$\hat{x}$ is our estimate of the robot's position.

Need for a formalization of the uncertainty of $\hat{x}$. For instance: stdev of a Gaussian.



How do we integrate the observation of new landmarks or odometry into this representation? $\rightarrow$ Via the Kalman Filter.

The Kalman Filter

Example use: tracking the position of a robot, tracking the ball in a football game, tracking people moving on the platforms of a train station.

Mathematically:

- We track (over time) the value of a state vector x that cannot be observed directly.
- We can predict a future value of x via a dynamic model.
- We can obtain observations (measurements) that are a function of the state.
- The Kalman filter merges the estimates obtained from the dynamic model and the observations.

Discrete-time Linear Time-invariant System Model

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{F}\boldsymbol{x}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle + \boldsymbol{v}\langle k\rangle$$
$$\boldsymbol{z}\langle k+1\rangle = \boldsymbol{H}\boldsymbol{x}\langle k\rangle + \boldsymbol{w}\langle k\rangle$$

- $\boldsymbol{x}$ is the state vector
- $\boldsymbol{u}$ is the system input
- $\boldsymbol{z}$ are the sensor measurements
- $\boldsymbol{F}$ describes the dynamics of the system
- $\boldsymbol{G}$ describes the coupling between the inputs and the state
- $\boldsymbol{H}$ describes how the state is mapped to the sensory channels
- $\boldsymbol{v}$ is the Gaussian and zero-mean process noise (cf. $\boldsymbol{F}$ and $\boldsymbol{G}$)
- $\boldsymbol{w}$ is the Gaussian and zero-mean measurement noise (cf. $\boldsymbol{H}$)

Discrete-time Linear Time-invariant System Model

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{F}\boldsymbol{x}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle + \boldsymbol{v}\langle k\rangle$$
$$\boldsymbol{z}\langle k+1\rangle = \boldsymbol{H}\boldsymbol{x}\langle k\rangle + \boldsymbol{w}\langle k\rangle$$

Our problem

Given a model of the system, the known inputs $\boldsymbol{u}$ and some noisy sensor measurements $\boldsymbol{z}$, estimate $\boldsymbol{x}$.

Example: Omnidirectional Robot

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{F}\boldsymbol{x}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle + \boldsymbol{v}\langle k\rangle$$
$$\boldsymbol{z}\langle k+1\rangle = \boldsymbol{H}\boldsymbol{x}\langle k\rangle + \boldsymbol{w}\langle k\rangle$$

The input $\boldsymbol{u}$ controls the velocity along $\mathbf{x}$ and $\mathbf{y}$.
The measurement $\boldsymbol{z}$ give the position of the robot (ultrasound beacons).

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{x}\langle k\rangle = \begin{pmatrix} x_0\langle k\rangle \\ x_1\langle k\rangle \end{pmatrix}$$

$$\boldsymbol{G} = \begin{pmatrix} \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \qquad \boldsymbol{u}\langle k\rangle = \begin{pmatrix} u_0\langle k\rangle \\ u_1\langle k\rangle \end{pmatrix}$$

$$\boldsymbol{H} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{z}\langle k\rangle = \begin{pmatrix} z_0\langle k\rangle \\ z_1\langle k\rangle \end{pmatrix}$$

Localizing The Robot

At every timestep:

- Predict the current state from the previous state.
- Read the sensor data.
- Correct the current state in light of measurements.

The Kalman Filter: Prediction and Correction

System model:

$$\boldsymbol{x}\langle k{+}1\rangle = \boldsymbol{F}\boldsymbol{x}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle + \boldsymbol{v}\langle k\rangle$$
$$\boldsymbol{z}\langle k{+}1\rangle = \boldsymbol{H}\boldsymbol{x}\langle k\rangle + \boldsymbol{w}\langle k\rangle$$

Prediction

$$\hat{\boldsymbol{x}}\left\langle k{+}1|k\right\rangle = \boldsymbol{F}\hat{\boldsymbol{x}}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle$$

$$\hat{\boldsymbol{P}}\left\langle k{+}1|k\right\rangle = \boldsymbol{F}\hat{\boldsymbol{P}}\left\langle k|k\right\rangle \boldsymbol{F}^T + \hat{\boldsymbol{V}}$$

Correction

$$\boldsymbol{\nu}\langle k{+}1\rangle = \boldsymbol{z}\langle k{+}1\rangle - \boldsymbol{H}\hat{\boldsymbol{x}}\left\langle k{+}1|k\right\rangle$$
$$\boldsymbol{K}\langle k{+}1\rangle = \hat{\boldsymbol{P}}\left\langle k{+}1|k\right\rangle \boldsymbol{H}^T \underbrace{\left(\boldsymbol{H}\hat{\boldsymbol{P}}\left\langle k{+}1|k\right\rangle \boldsymbol{H}^T + \hat{\boldsymbol{W}}\right)}_{\boldsymbol{S}}^{-1}$$
$$\hat{\boldsymbol{x}}\left\langle k{+}1|k{+}1\right\rangle = \hat{\boldsymbol{x}}\left\langle k{+}1|k\right\rangle + \boldsymbol{K}\langle k{+}1\rangle \boldsymbol{\nu}\langle k{+}1\rangle$$
$$\hat{\boldsymbol{P}}\left\langle k{+}1|k{+}1\right\rangle = \hat{\boldsymbol{P}}\left\langle k{+}1|k\right\rangle - \boldsymbol{K}\langle k{+}1\rangle \boldsymbol{H}\hat{\boldsymbol{P}}\left\langle k{+}1|k\right\rangle$$

Rather: $\hat{\boldsymbol{x}}\langle k{+}1|k\rangle = \boldsymbol{F}\hat{\boldsymbol{x}}\langle k|k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle$

Initial conditions: $\hat{\boldsymbol{x}}\langle 0|0\rangle$, $\hat{\boldsymbol{P}}\langle 0|0\rangle$

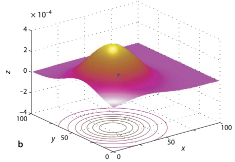
The Kalman Filter: Prediction

$$\hat{\boldsymbol{x}}\langle k+1|k\rangle = \boldsymbol{F}\hat{\boldsymbol{x}}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle$$

$$\hat{\boldsymbol{P}}\langle k+1|k\rangle = \boldsymbol{F}\hat{\boldsymbol{P}}\langle k|k\rangle\boldsymbol{F}^T + \hat{\boldsymbol{V}}$$

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{F}\boldsymbol{x}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle + \boldsymbol{v}\langle k\rangle$$

$$\boldsymbol{z}\langle k+1\rangle = \boldsymbol{H}\boldsymbol{x}\langle k\rangle + \boldsymbol{w}\langle k\rangle$$



- The covariance matrix $\hat{\boldsymbol{P}}$ models the state uncertainty.
- With only prediction, the Kalman filter is open-loop (relies on the accuracy of the model).
- During prediction, $\hat{\boldsymbol{P}}$ (typically) increases.

The Kalman Filter: Correction

$$\boldsymbol{\nu}\langle k+1\rangle = \boldsymbol{z}\langle k+1\rangle - \boldsymbol{H}\hat{\boldsymbol{x}}\langle k+1|k\rangle$$

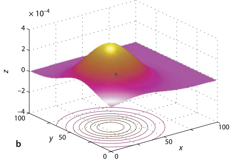
$$\boldsymbol{K}\langle k+1\rangle = \hat{\boldsymbol{P}}\langle k+1|k\rangle \boldsymbol{H}^T \underbrace{\left(\boldsymbol{H}\hat{\boldsymbol{P}}\langle k+1|k\rangle \boldsymbol{H}^T + \hat{\boldsymbol{W}}\right)}_{S}{}^{-1}$$

$$\hat{\boldsymbol{x}}\langle k+1|k+1\rangle = \hat{\boldsymbol{x}}\langle k+1|k\rangle + \boldsymbol{K}\langle k+1\rangle \boldsymbol{\nu}\langle k+1\rangle$$

$$\hat{\boldsymbol{P}}\langle k+1|k+1\rangle = \hat{\boldsymbol{P}}\langle k+1|k\rangle - \boldsymbol{K}\langle k+1\rangle \boldsymbol{H}\hat{\boldsymbol{P}}\langle k+1|k\rangle$$

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{F}\boldsymbol{x}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle + \boldsymbol{v}\langle k\rangle$$

$$\boldsymbol{z}\langle k+1\rangle = \boldsymbol{H}\boldsymbol{x}\langle k\rangle + \boldsymbol{w}\langle k\rangle$$



- During correction, $\hat{\boldsymbol{P}}$ decreases.
- The Kalman Gain specifies how strongly
  - $\boldsymbol{x}$ is drawn towards the measurement
  - The state uncertainty is reduced

Example: Walking In The Mountains

The input $\boldsymbol{u}$ controls the size of a step.
The measurement $\boldsymbol{z}$ gives my position along the path when I see a mountaintop.

$$\boldsymbol{F} = 1 \qquad \boldsymbol{x}\langle k\rangle = x\langle k\rangle$$
$$\boldsymbol{G} = 1 \qquad \boldsymbol{u}\langle k\rangle = u\langle k\rangle \qquad \hat{\boldsymbol{V}} = 0.1^2$$
$$\boldsymbol{H} = 1 \qquad \boldsymbol{z}\langle k\rangle = z\langle k\rangle \qquad \hat{\boldsymbol{W}} = 50^2$$

Notes On The Kalman Filter

- Recursive and Asynchronous
- Requires a starting estimate of the state and state uncertainty.
- Requires a reasonable estimate of the process and measurement noise.

- Theoretical properties/justifications (in a nutshell)
  - Under the linear dynamical and observation models, and with the gaussian noise assumptions, all states and observations form actually a jointly gaussian random vector
  - The Kalman filter computes exactly the conditional expectation and the conditional covariance matrix of states given observations, hence the state estimate is also a 'least mean-square error' estimate.
  - The recursive 'left-right' formulas are justified by fact that the process is actually a Hidden-Markov process.

Plan

Odometry

Odometer: sensor that measures the distance travelled by the robot – typically: wheel rotation angles.

Keeping track of the direction of movement: differential odometry, compass, gyro.

Affected by systematic error (wheel radius slightly off) and random errors (slippage).

Modeling A Non-holonomic Vehicle

Discrete-time Model Of The Robot's Configuration

Odometry reading for the preceding time interval: $\delta\langle k\rangle = (\delta_d, \delta_\theta)$

Pose at time k:

$$\xi\langle k\rangle \sim \begin{pmatrix} \cos\theta\langle k\rangle & -\sin\theta\langle k\rangle & x\langle k\rangle \\ \sin\theta\langle k\rangle & \cos\theta\langle k\rangle & y\langle k\rangle \\ 0 & 0 & 1 \end{pmatrix}$$

Pose at time $k+1$:

$$\xi\langle k+1\rangle \sim \begin{pmatrix} \cos\theta\langle k\rangle & -\sin\theta\langle k\rangle & x\langle k\rangle \\ \sin\theta\langle k\rangle & \cos\theta\langle k\rangle & y\langle k\rangle \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\delta_\theta & -\sin\delta_\theta & 0 \\ \sin\delta_\theta & \cos\delta_\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \delta_d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\sim \begin{pmatrix} \cos(\theta\langle k\rangle+\delta_\theta) & -\sin(\theta\langle k\rangle+\delta_\theta) & x\langle k\rangle+\delta_d\cos(\theta\langle k\rangle+\delta_\theta) \\ \sin(\theta\langle k\rangle+\delta_\theta) & \cos(\theta\langle k\rangle+\delta_\theta) & y\langle k\rangle+\delta_d\sin(\theta\langle k\rangle+\delta_\theta) \\ 0 & 0 & 1 \end{pmatrix}$$

Modeling A Non-holonomic Vehicle With Noisy Odometry

Error model: continuous random variables v_d and v_θ.

$$\xi\langle k+1\rangle = \begin{pmatrix} x\langle k\rangle + \left(\delta_d\langle k\rangle + \nu_d\right)\cos\left(\theta\langle k\rangle + \delta_\theta + \nu_\theta\right) \\ y\langle k\rangle + \left(\delta_d\langle k\rangle + \nu_d\right)\sin\left(\theta\langle k\rangle + \delta_\theta + \nu_\theta\right) \\ \theta\langle k\rangle + \delta_\theta + \nu_\theta \end{pmatrix}$$

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{f}\left(\boldsymbol{x}\langle k\rangle, \delta\langle k\rangle, \boldsymbol{v}\langle k\rangle\right)$$

Compare with

$$\begin{aligned} \boldsymbol{x}\langle k+1\rangle &= \boldsymbol{F}\boldsymbol{x}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle + \boldsymbol{v}\langle k\rangle \\ \boldsymbol{z}\langle k+1\rangle &= \boldsymbol{H}\boldsymbol{x}\langle k\rangle + \boldsymbol{w}\langle k\rangle \end{aligned}$$

EKF: The Extended Kalman Filter

System Model

$$\xi\langle k{+}1\rangle = \begin{pmatrix} x\langle k\rangle + \big(\delta_d\langle k\rangle + v_d\big)\cos\big(\theta\langle k\rangle + \delta_\theta + v_\theta\big) \\ y\langle k\rangle + \big(\delta_d\langle k\rangle + v_d\big)\sin\big(\theta\langle k\rangle + \delta_\theta + v_\theta\big) \\ \theta\langle k\rangle + \delta_\theta + v_\theta \end{pmatrix}$$

$$\boldsymbol{x}\langle k{+}1\rangle = \boldsymbol{f}\big(\boldsymbol{x}\langle k\rangle,\, \delta\langle k\rangle,\, \boldsymbol{v}\langle k\rangle\big)$$

Local linear approximation:

$$\hat{\boldsymbol{x}}\langle k{+}1\rangle = \hat{\boldsymbol{x}}\langle k\rangle + \boldsymbol{F}_x\big(\boldsymbol{x}\langle k\rangle - \hat{\boldsymbol{x}}\langle k\rangle\big) + \boldsymbol{F}_v\boldsymbol{v}\langle k\rangle$$

$$\boldsymbol{F}_x = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{v=0} = \begin{pmatrix} 1 & 0 & -\delta_d\langle k\rangle - \sin\big(\theta\langle k\rangle + \delta_\theta\big) \\ 0 & 1 & \delta_d\langle k\rangle\cos\big(\theta\langle k\rangle + \delta_\theta\big) \\ 0 & 0 & 1 \end{pmatrix} \qquad \boldsymbol{F}_v = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}\right|_{v=0} = \begin{pmatrix} \cos\big(\theta\langle k\rangle + \delta_\theta\big) & -\delta_d\langle k\rangle\sin\big(\theta\langle k\rangle + \delta_\theta\big) \\ \sin\big(\theta\langle k\rangle + \delta_\theta\big) & \delta_d\langle k\rangle\cos\big(\theta\langle k\rangle + \delta_\theta\big) \\ 0 & 1 \end{pmatrix}$$

EKF: Precition

$$\hat{\boldsymbol{x}}\langle k+1|k\rangle = \boldsymbol{f}\big(\hat{\boldsymbol{x}}\langle k\rangle, \delta\langle k\rangle, 0\big)$$

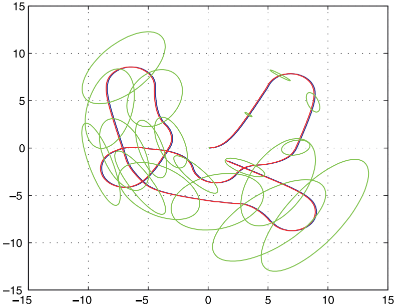
$$\hat{\boldsymbol{P}}\langle k+1|k\rangle = \boldsymbol{F}_x\langle k\rangle \hat{\boldsymbol{P}}\langle k|k\rangle \boldsymbol{F}_x\langle k\rangle^T + \boldsymbol{F}_v\langle k\rangle \hat{\boldsymbol{V}} \boldsymbol{F}_v\langle k\rangle^T$$

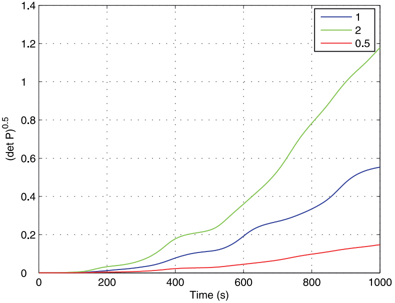
Compare to the regular KF prediction:

$$\hat{\boldsymbol{x}}\langle k+1|k\rangle = \boldsymbol{F}\hat{\boldsymbol{x}}\langle k\rangle + \boldsymbol{G}\boldsymbol{u}\langle k\rangle$$

$$\hat{\boldsymbol{P}}\langle k+1|k\rangle = \boldsymbol{F}\hat{\boldsymbol{P}}\langle k|k\rangle \boldsymbol{F}^T + \hat{\boldsymbol{V}}$$

Open-loop EKF: Illustration





Left: trajectory "guessing" via odometry (with uncertainty ellipses)

Right: uncertainty growth with time (more quickly if sensors are less accurate)

Plan

Using A Map

With Dead-reckoning, position uncertainty constantly grows.

Solution:

- bring a map of prominent landmarks
- look around for landmarks
- incorporate this information into our position estimate via the Kalman correction stage:

$$\boldsymbol{\nu}\langle k+1\rangle = \boldsymbol{z}\langle k+1\rangle - \boldsymbol{H}\hat{\boldsymbol{x}}\langle k+1|k\rangle$$

$$\boldsymbol{K}\langle k+1\rangle = \hat{\boldsymbol{P}}\langle k+1|k\rangle \boldsymbol{H}^T \underbrace{\left(\boldsymbol{H}\hat{\boldsymbol{P}}\langle k+1|k\rangle \boldsymbol{H}^T + \hat{\boldsymbol{W}}\right)}_{S}^{-1}$$

$$\hat{\boldsymbol{x}}\langle k+1|k+1\rangle = \hat{\boldsymbol{x}}\langle k+1|k\rangle + \boldsymbol{K}\langle k+1\rangle \boldsymbol{\nu}\langle k+1\rangle$$

$$\hat{\boldsymbol{P}}\langle k+1|k+1\rangle = \hat{\boldsymbol{P}}\langle k+1|k\rangle - \boldsymbol{K}\langle k+1\rangle \boldsymbol{H}\hat{\boldsymbol{P}}\langle k+1|k\rangle$$

Working With A Nonlinear Observation Model

Let us assume a general expression for the observation model:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}_v, \boldsymbol{x}_f, \boldsymbol{w})$$

where $\boldsymbol{x}_v$ is the world vehicle coordinates, $\boldsymbol{x}_f$ is the world coordinates of an observed feature.

With a range/bearing sensor:

$$\boldsymbol{z} = \begin{pmatrix} \sqrt{(y_i - y_v)^2 + (x_i - x_v)^2} \\ \tan^{-1}(y_i - y_v)/(x_i - x_v) - \theta_v \end{pmatrix} + \begin{pmatrix} w_r \\ w_\beta \end{pmatrix}$$

with $\boldsymbol{z} = (r, \beta)$, and

$$\begin{pmatrix} w_r \\ w_\beta \end{pmatrix} \sim N(0, W), \;\; W = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\beta^2 \end{pmatrix}$$

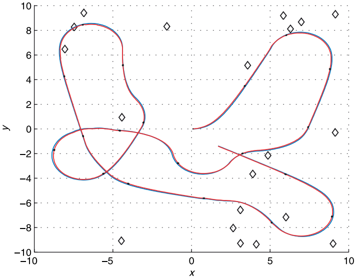
Working With A Nonlinear Observation Model

The innovation is written as

$$\boldsymbol{\nu}\langle k+1\rangle = \boldsymbol{z}\langle k+1\rangle - \boldsymbol{h}\Big(\hat{\boldsymbol{x}}(k+1\big|k), x_f, 0\Big)$$

which requires to linearize the observation model

$$\boldsymbol{z}\langle k\rangle = \hat{\boldsymbol{h}} + \boldsymbol{H}_x\big(\boldsymbol{x}\langle k\rangle - \hat{\boldsymbol{x}}\langle k\rangle\big) + \boldsymbol{H}_w\boldsymbol{w}\langle k\rangle$$

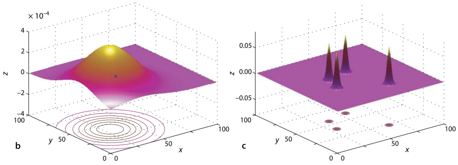


The Extended Kalman Filter

The EKF linearizes the dynamic and measurement models about the current state estimate, then applies the linear Kalman filter.

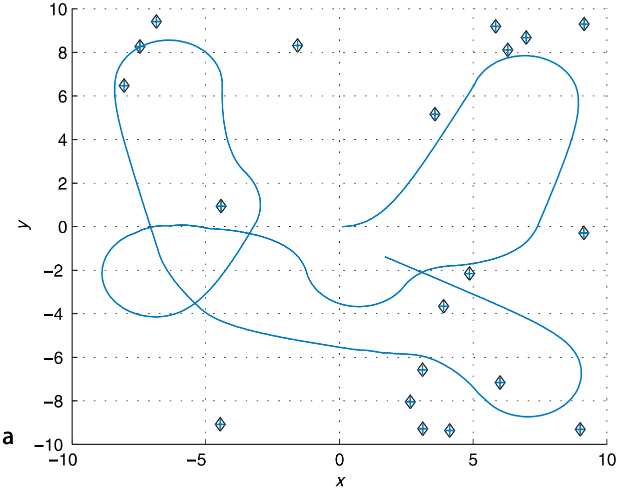
Risk of quick divergence if the initial conditions or dynamic/measurement models are incorrect.

The state is modeled with a Gaussian distribution. Result: can only hold one hypothesis, problem with data association in the measurement stage. For multi-hypothesis: particle filters.



Still de facto standard in navigation systems such as car GPS.

Creating A Map



Creating A Map

Assuming a perfect odometry, creating a map is straightforward.

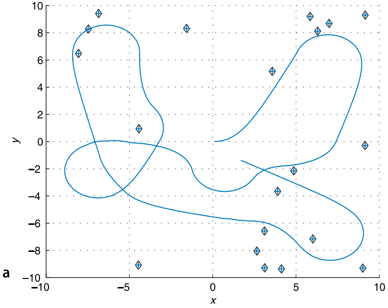
- Kalman filter state: landmark positions

$$\hat{\boldsymbol{x}} = (x_1, y_1, x_2, y_2, \cdots x_M, y_M)^T$$

- Kalman prediction:

$$\hat{\boldsymbol{x}}\langle k+1|k\rangle = \hat{\boldsymbol{x}}\langle k|k\rangle$$

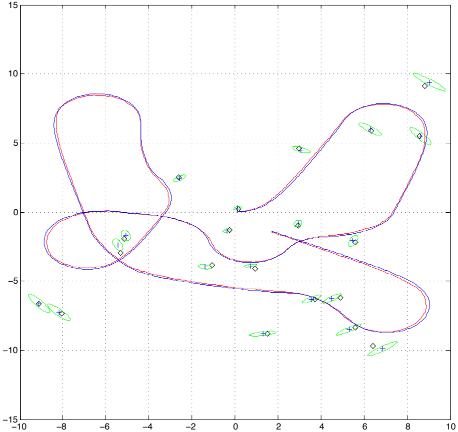
$$\hat{\boldsymbol{P}}\langle k+1|k\rangle = \hat{\boldsymbol{P}}\langle k|k\rangle$$



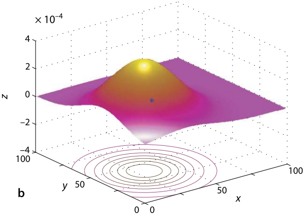
Simultaneous Localization And Mapping (SLAM)

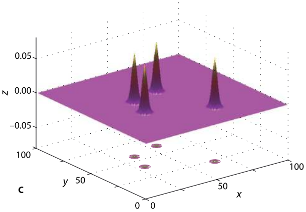
SLAM: a chicken-and-egg problem

$$\hat{\boldsymbol{x}} = (x_v, y_v, \theta_v, x_1, y_1, x_2, y_2, \cdots x_M, y_M)$$



Particle Filtering





By contrast to the (Extended) Kalman Filter: handles non-Gaussian state representations (Multiple hypotheses, non-Gaussian uncertainty), does not care about model linearity.

- Maintain several states concurrently
- Randomly perturb states when updating them with the dynamics model
- Get rid of particles that do not explain new measurements well

Particle Filtering For Localization

Goal: track the vehicle's pose (x, y, θ).

The filter maintains N particles $\boldsymbol{x}_{v,i}$.

Loop:

1. Use the dynamic model to update all particles:

$$\boldsymbol{x}_{\nu,i}\langle k+1\rangle = \boldsymbol{f}\Big(\boldsymbol{x}_{\nu,i}\langle k\rangle, \delta\langle k\rangle\Big) + \boldsymbol{q}\langle k\rangle$$

2. Weight all particles in light of new observation $\boldsymbol{z}$:

$$w_i = e^{-\frac{1}{2}\nu_i^T \boldsymbol{L} \nu_i} + a, \qquad \nu_i = \boldsymbol{h}(\boldsymbol{x}_{v,i}, \boldsymbol{x}_f) - \boldsymbol{z}$$

3. Resample particle set: draw N times from the particle set, where at each draw the probability of selecting particle i is proportional to w_i.

Pose estimate: mean of all particles.
Pose uncertainty: standard deviation around mean.