Lecture 9
Approximate nonlinear filtering

- Nonlinear filtering
- Extended Kalman filter
- Monte Carlo based updates

Nonlinear filtering

- nonlinear Markov model:

$$x_{t+1} = f(x_t, w_t), \qquad y_t = g(x_t, v_t)$$

  - f is (possibly nonlinear) dynamics function
  - g is (possibly nonlinear) measurement or output function
  - $w_0, w_1, \ldots, v_0, v_1, \ldots$ are independent, zero mean
  - even if w, v Gaussian, x and y need not be

- nonlinear filtering problem: find, $e.g.$,

$$\hat{x}_{t|t-1} = \mathbf{E}(x_t | y_0, \ldots, y_{t-1}), \qquad \hat{x}_{t|t} = \mathbf{E}(x_t | y_0, \ldots, y_t)$$

Nonlinear filtering

- general solution requires propagating (infinite dimensional) conditional densities of $(x_t|y_0,\ldots,y_{t-1})$ or $(x_t|y_0,\ldots,y_t)$; not practical except for $n=1$ or 2

- *extended Kalman filter*

  - replaces f, g with affine approximations at current estimate
  - uses KF formulas for measurement, time updates

- other methods ($e.g.$, 'particle filters') based on Monte Carlo methods

Extended Kalman filter

- extended Kalman filter (EKF) is *heuristic* for nonlinear filtering problem
- often works well (when tuned properly), but sometimes not
- widely used in practice
- based on
  - linearizing dynamics and output functions at current estimate
  - propagating an approximation of the conditional expectation and covariance

Extended Kalman filter — time update

- linearize dynamics function at $x = \hat{x}_{t|t}$:

$$x_{t+1} = f(x_t, w_t) \approx f(\hat{x}_{t|t}, 0) + A(x_t - \hat{x}_{t|t}) + Bw_t$$

where

$$A = \frac{\partial f}{\partial x}(\hat{x}_{t|t}, 0), \qquad B = \frac{\partial f}{\partial w}(\hat{x}_{t|t}, 0)$$

- using $\mathbf{E}\, w_t = 0$ and assuming $\mathbf{E}(x_t - \hat{x}_{t|t}) \approx 0$ (we'd have equality if $\hat{x}_{t|t}$ is the conditional mean),

$$\hat{x}_{t+1|t} \approx f(\hat{x}_{t|t}, 0), \qquad \Sigma_{t+1|t} \approx A\Sigma_{t|t}A^T + B\Sigma_w B^T$$

- EKF uses these formulas for time update

Extended Kalman filter — measurement update

- linearize output function at $x = \hat{x}_{t|t-1}$:

$$y_t = g(x_t, v_t) \approx g(\hat{x}_{t|t-1}, 0) + C(x_t - \hat{x}_{t|t-1}) + Dv_t$$

where

$$C = \frac{\partial g}{\partial x}(\hat{x}_{t|t-1}, 0), \qquad D = \frac{\partial g}{\partial v}(\hat{x}_{t|t-1}, 0)$$

- using $\mathbf{E}\, v_t = 0$ and assuming $\mathbf{E}(x_t - \hat{x}_{t|t-1}) \approx 0$ (we'd have equality if $\hat{x}_{t|t-1}$ is the conditional mean),

$$\begin{aligned} \hat{x}_{t|t} &= \hat{x}_{t|t-1} + \Sigma_{t|t-1} C^T \left(C \Sigma_{t|t-1} C^T + V \right)^{-1} (y_t - g(\hat{x}_{t|t-1}, 0)) \\ \Sigma_{t|t} &= \Sigma_{t|t-1} - \Sigma_{t|t-1} C^T \left(C \Sigma_{t|t-1} C^T + V \right)^{-1} C \Sigma_{t|t-1} \end{aligned}$$

Example

- p_t, $u_t \in \mathbf{R}^2$ are position and velocity of vehicle, with $(p_0, u_0) \sim \mathcal{N}(0, I)$
- vehicle dynamics:

$$p_{t+1} = p_t + 0.1 u_t, \qquad u_{t+1} = \begin{bmatrix} 0.85 & 0.15 \\ -0.1 & 0.85 \end{bmatrix} u_t + w_t$$

  w_t are IID $\mathcal{N}(0, I)$

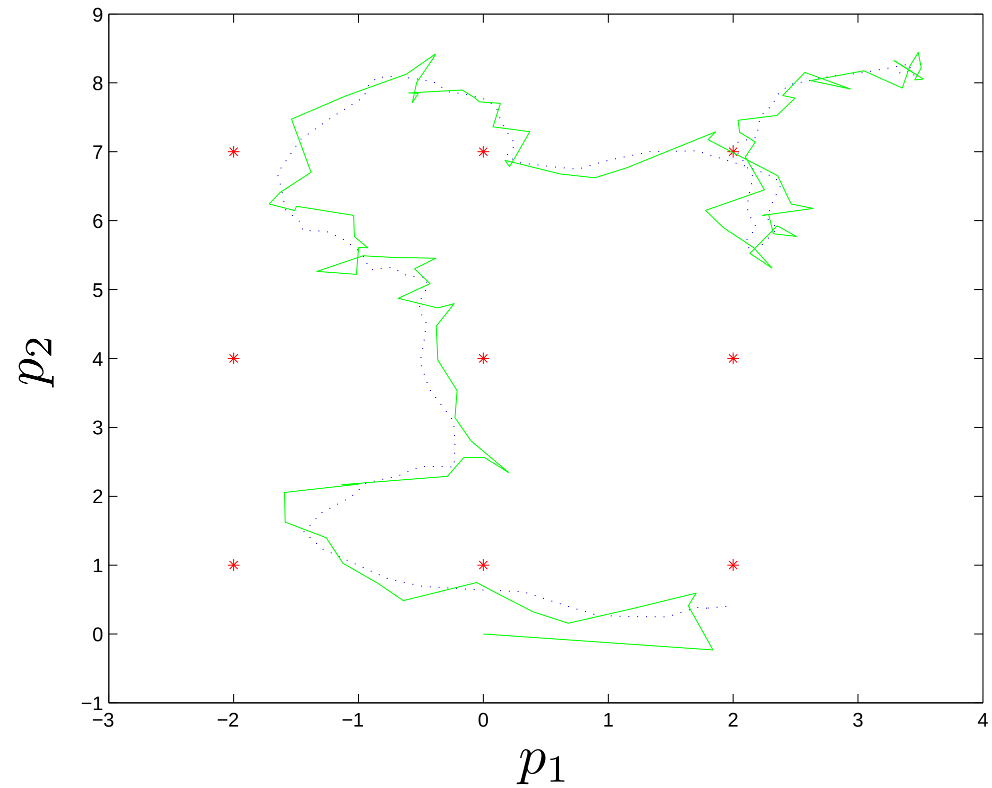
- measurements: noisy measurements of distance to 9 points $p_i \in \mathbf{R}^2$

$$(y_t)_i = \|p_t - p_i\| + (v_t)_i, \quad i = 1, \ldots, 9,$$

  $(v_t)_i$ are IID $\mathcal{N}(0, 0.3^2)$

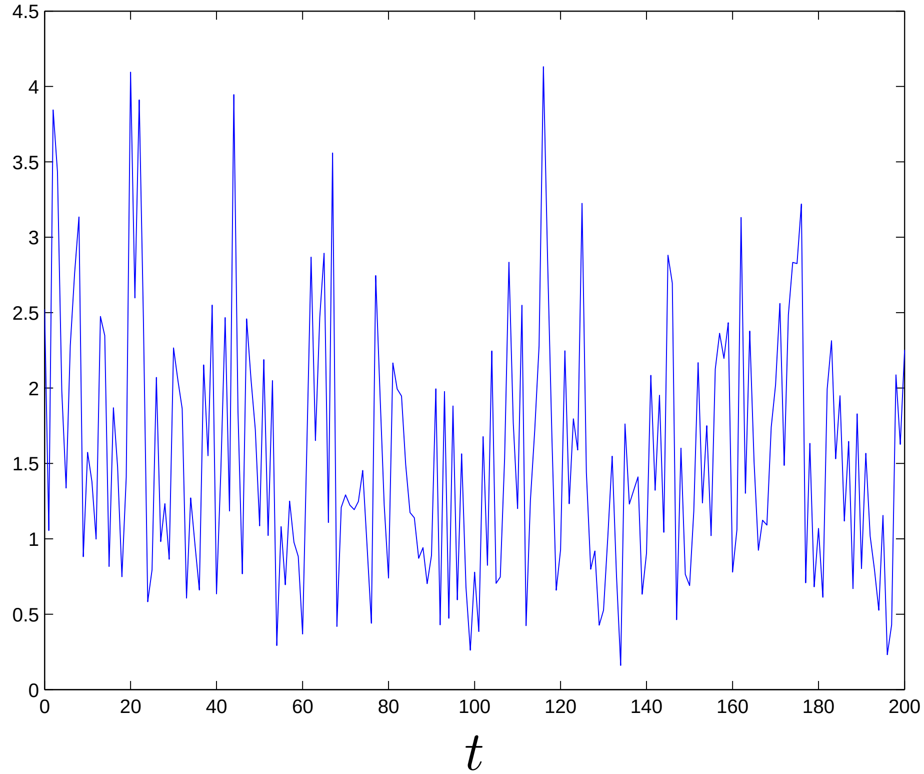
EKF results

- EKF initialized with $\hat{x}_{0|-1} = 0$, $\Sigma(0|-1) = I$, where $x = (p, u)$
- p_i shown as stars; p_t as dotted curve; $\hat{p}_{t|t}$ as solid curve



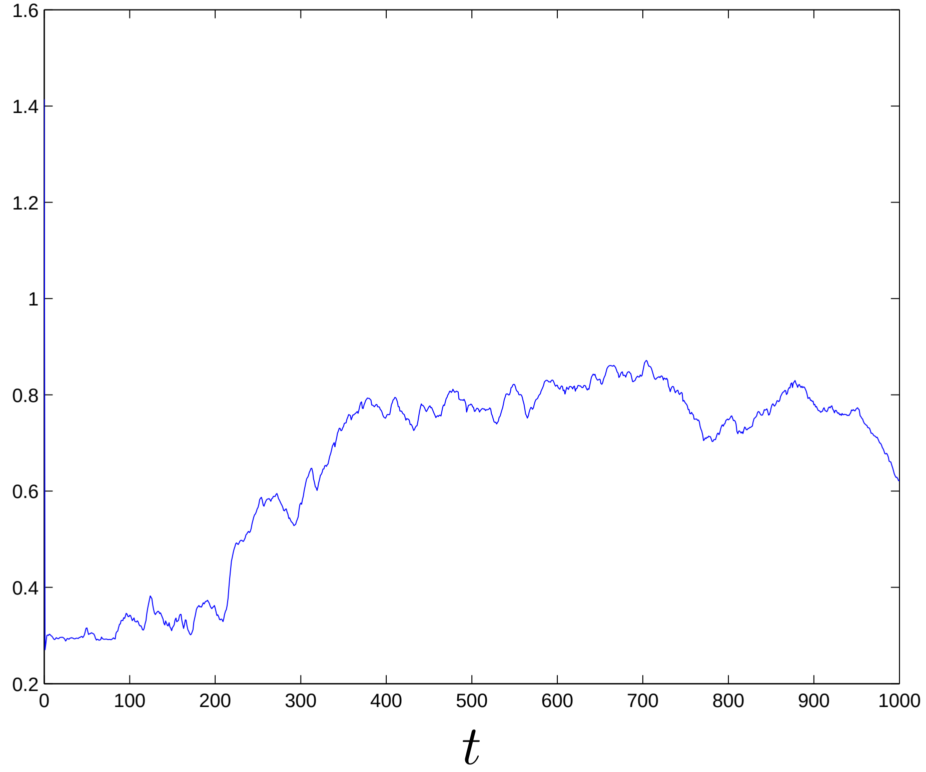
Current position estimation error

$\|\hat{p}_{t|t} - p_t\|$ versus t



Current position estimation predicted error

$(\Sigma(t|t)_{11} + \Sigma(t|t)_{22})^{1/2}$ versus t



Monte Carlo based methods

- instead of linearizing f, g via derivatives, use Monte Carlo methods
- requires ability to generate samples of w_t, v_t
- we'll *assume* that error covariances are normal (which is generally not the case):

$$\hat{x}_{t|t-1} - x_t \sim \mathcal{N}(0, \Sigma_{t|t-1}), \qquad \hat{x}_{t|t} - x_t \sim \mathcal{N}(0, \Sigma_{t|t})$$

Monte Carlo time update

- generate samples of $x_t^{(k)}$, $k = 1, \ldots, K$, from $\mathcal{N}(\hat{x}_{t|t}, \Sigma_{t|t})$
- generate samples $w_t^{(k)}$, $k = 1, \ldots, K$, of w_t
- these give samples $x_{t+1}^{(k)} = f(x_t^{(k)}, w_t^{(k)})$ of $x_{t+1|t}$
- use sample mean as $\hat{x}_{t+1|t}$: $\quad \hat{x}_{t+1|t} = \frac{1}{K}\sum_{k=1}^{K} x_{t+1}^{(k)}$
- use sample covariance as $\Sigma_{t+1|t}$:

$$\Sigma_{t+1|t} = \frac{1}{K}\sum_{k=1}^{K}(x_{t+1}^{(k)} - \hat{x}_{t+1|t})(x_{t+1}^{(k)} - \hat{x}_{t+1|t})^T$$

Monte Carlo measurement update

- generate samples of $x_t^{(k)}$, $k = 1, \dots, K$, from $\mathcal{N}(\hat{x}_{t|t-1}, \Sigma_{t|t-1})$
- generate samples $v_t^{(k)}$, $k = 1, \dots, K$, of v_t
- these give samples $y_t^{(k)} = g(x_t^{(k)}, v_t^{(k)})$ of y_t
- use sample mean as estimate of $\mathbf{E}\, y_t$: $\bar{y}_t = \frac{1}{K}\sum_{k=1}^{K} y_t^{(k)}$
- use sample mean to estimate covariances

$$\begin{aligned}
\Sigma_{xy} &= \frac{1}{K}\sum_{k=1}^{K}(x_t^{(k)} - \hat{x}_{t|t-1})(y_t^{(k)} - \bar{y}_t)^T \\
\Sigma_y &= \frac{1}{K}\sum_{k=1}^{K}(y_t^{(k)} - \bar{y}_t)(y_t^{(k)} - \bar{y}_t)^T
\end{aligned}$$

- do measurement update using these approximations:

$$\begin{aligned}\hat{x}_{t|t} &= \hat{x}_{t|t-1} + \Sigma_{xy}\Sigma_y^{-1}(y_t - \bar{y}_t) \\ \Sigma_{t|t} &= \Sigma_{t|t-1} - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{xy}^T\end{aligned}$$