Tracking
with focus on the particle filter

Michael Rubinstein

IDC

Problem overview

- Input
  - (Noisy) Sensor measurements
- Goal
  - Estimate most probable measurement at time k using measurement up to time k’

    k’<k: **prediction**

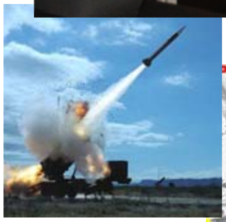
    k‘>k: **smoothing**

- Many problems require estimation of the state of systems that change over time using noisy measurements on the system

Applications

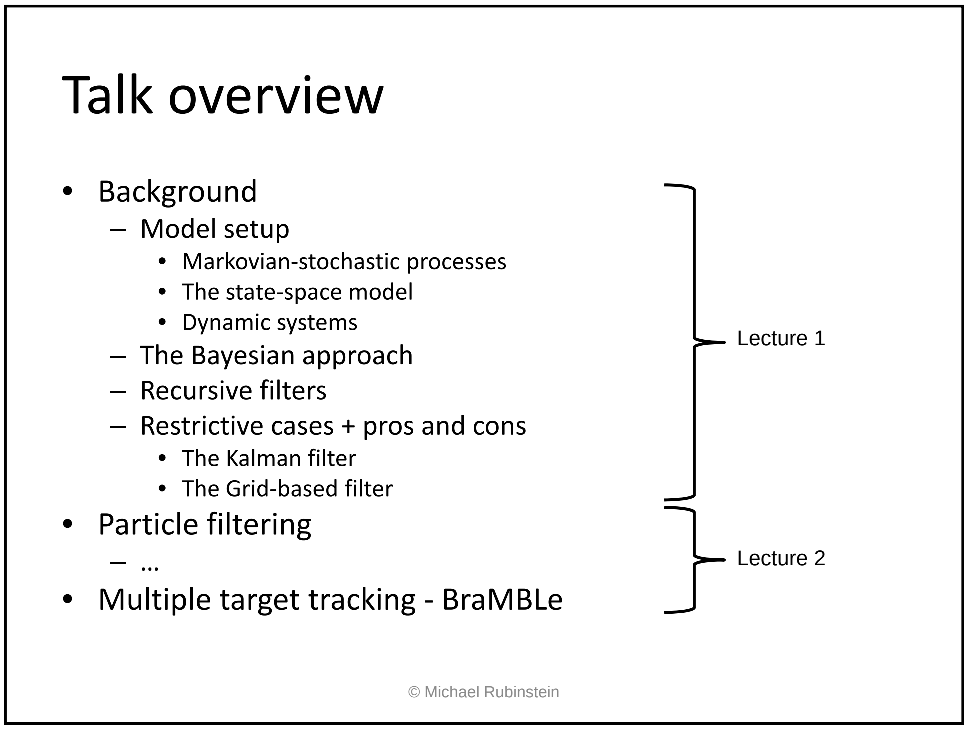
- Ballistics
- Robotics
  - Robot localization
- Tracking hands/cars/…
- Econometrics
  - Stock prediction
- Navigation

- Many more…



Challenges

- Measurements
  - Noise
  - Errors
- Detection specific
  - Full/partial occlusions
  - False positives/false negatives
  - Entering/leaving the scene
- Efficiency
- Multiple models and switching dynamics
- Multiple targets,
- …



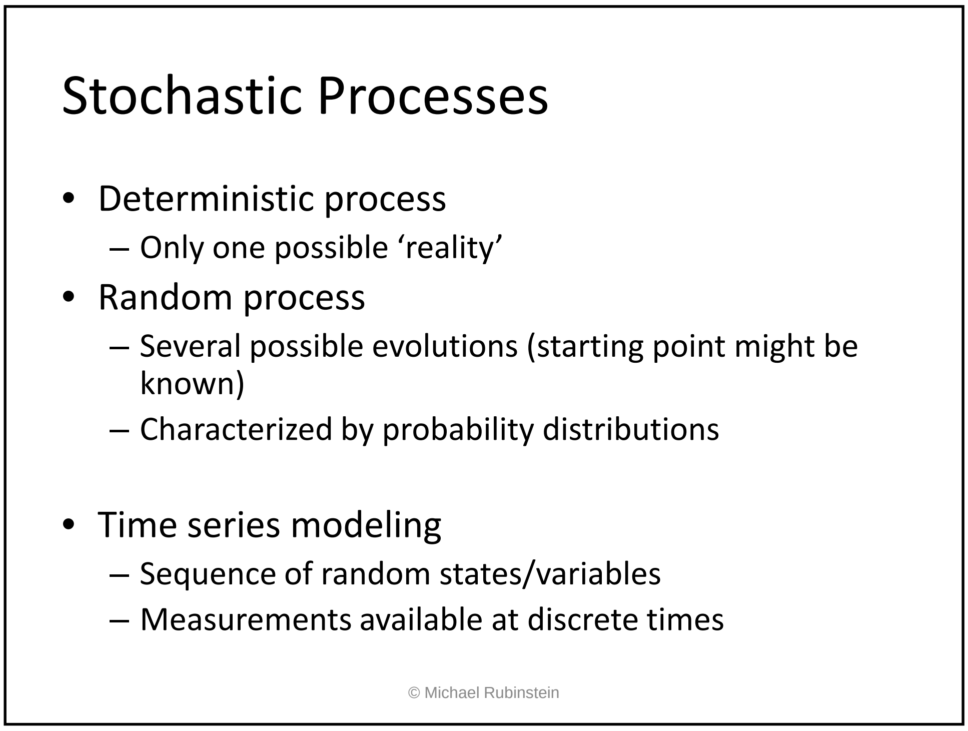
Talk overview

- Background
  - Model setup
    - Markovian-stochastic processes
    - The state-space model
    - Dynamic systems
  - The Bayesian approach
  - Recursive filters
  - Restrictive cases + pros and cons
    - The Kalman filter
    - The Grid-based filter

Lecture 1

- Particle filtering
  - …
- Multiple target tracking - BraMBLe

Lecture 2



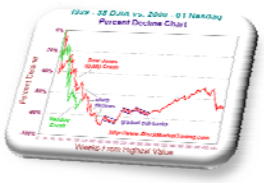
Stochastic Processes

- Deterministic process
  - Only one possible 'reality'
- Random process
  - Several possible evolutions (starting point might be known)
  - Characterized by probability distributions
- Time series modeling
  - Sequence of random states/variables
  - Measurements available at discrete times

State space

- **The state vector** contains all available information to describe the investigated system
  - usually multidimensional: $X(k) \in R^{N_x}$
- **The measurement vector** represents observations related to the state vector $Z(k) \in R^{N_z}$
  - Generally (but not necessarily) of lower dimension than the state vector

State space



- Tracking:

$N_x = 3$ $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ $N_x = 4$ $\begin{bmatrix} x \\ v_x \\ y \\ v_y \end{bmatrix}$

Econometrics:

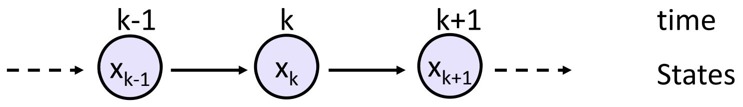
- Monetary flow
- Interest rates
- Inflation
- …

(First-order) Markov process

- The Markov property – the likelihood of a future state depends on present state only

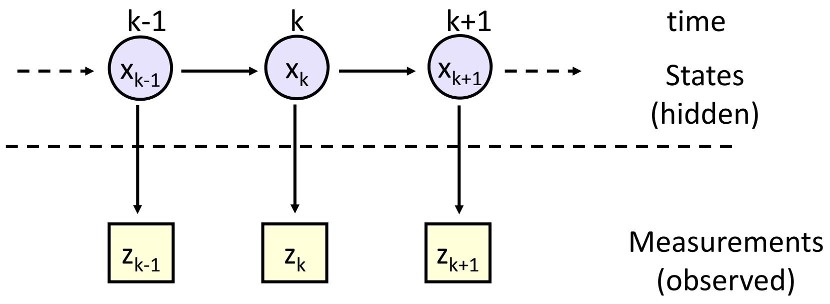
$$\Pr[X(k+h) = y \mid X(s) = x(s), \forall s \le k] = \Pr[X(k+h) = y \mid X(k) = x(k)], \forall h > 0$$

- Markov chain – A stochastic process with Markov property



Hidden Markov Model (HMM)

- the state is not directly visible, but output dependent on the state is visible



Dynamic System

$$\begin{tikzcd}
 & k-1 & k & k+1 & \\
\cdots \arrow[r, dashed] & x_{k-1} \arrow[r, "f_k"] \arrow[d] & x_k \arrow[r] \arrow[d, "h_k"] & x_{k+1} \arrow[r, dashed] \arrow[d] & \cdots \\
 & z_{k-1} & z_k & z_{k+1} &
\end{tikzcd}$$

Stochastic diffusion

State equation: $x_k = f_k(x_{k-1}, \boxed{v_k})$

x_k	state vector at time instant k
f_k	state transition function, $f_k : R^{N_x} \times R^{N_v} \to R^{N_x}$
v_k	i.i.d process noise

Observation equation: $z_k = h_k(x_k, \boxed{w_k})$

z_k	observations at time instant k
h_k	observation function, $h_k : R^{N_x} \times R^{N_w} \to R^{N_z}$
w_k	i.i.d measurement noise

A simple dynamic system

- $X = [x, y, v_x, v_y]$ (4-dimensional state space)
- Constant velocity motion:

$$f(X, v) = [x + \Delta t \cdot v_x, y + \Delta t \cdot v_y, v_x, v_y] + v$$

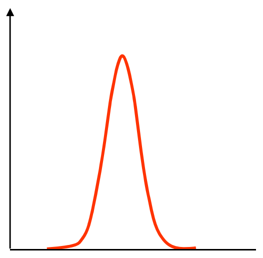
$$v \sim N(0, Q) \quad Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q^2 & 0 \\ 0 & 0 & 0 & q^2 \end{pmatrix}$$

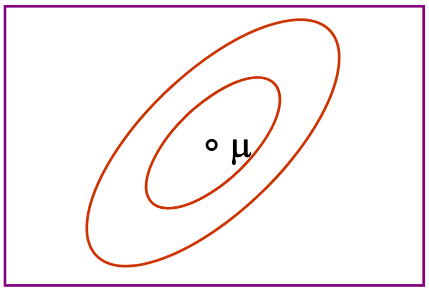
- Only position is observed:

$$z = h(X, w) = [x, y] + w$$

$$w \sim N(0, R) \quad R = \begin{pmatrix} r^2 & 0 \\ 0 & r^2 \end{pmatrix}$$

Gaussian distribution





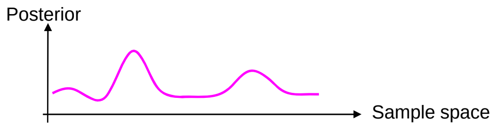
Yacov Hel-Or

$$p(x) \sim N(\mu,\Sigma) = \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\}$$

The Bayesian approach

Thomas Bayes

- Construct the posterior probability density function $p(x_k \mid z_{1:k})$ of the state based on all available information



- By knowing the posterior many kinds of estimates for x_k can be derived
  - mean (expectation), mode, median, …
  - Can also give estimation of the accuracy (e.g. covariance)

Recursive filters

- For many problems, estimate is required each time a new measurement arrives
- **Batch** processing
  - Requires *all* available data
- **Sequential** processing
  - New data is processed upon arrival
  - Need not store the complete dataset
  - Need not reprocess all data for each new measurement
  - Assume no out-of-sequence measurements (solutions for this exist as well…)

Recursive Bayes filters

- Given:
  - System models in probabilistic forms

$$x_k = f_k(x_{k-1}, v_k) \leftrightarrow p(x_k \mid x_{k-1})$$

Markovian process

$$z_k = h_k(x_k, w_k) \leftrightarrow p(z_k \mid x_k)$$

Measurements are conditionally independent given the state

(known statistics of v_k, w_k)

  - Initial state $p(x_0 \mid z_0) = p(x_0)$ also known as the **prior**
  - Measurements $z_1, \dots, z_k$

Recursive Bayes filters

- Prediction step (a-priori)

$$p(x_{k-1} \mid z_{1:k-1}) \rightarrow p(x_k \mid z_{1:k-1})$$

  - Uses the system model to predict forward
  - Deforms/translates/spreads state pdf due to random noise

- Update step (a-posteriori)

$$p(x_k \mid z_{1:k-1}) \rightarrow p(x_k \mid z_{1:k})$$

  - Update the prediction in light of new data
  - Tightens the state pdf

General prediction-update framework

- Assume $p(x_{k-1} \mid z_{1:k-1})$ is given at time k-1
- Prediction:

$$p(x_k \mid z_{1:k-1}) = \int \overbrace{p(x_k \mid x_{k-1})}^{\text{System model}} \overbrace{p(x_{k-1} \mid z_{1:k-1})}^{\text{Previous posterior}} dx_{k-1} \qquad (1)$$

- Using Chapman-Kolmogorov identity + Markov property

General prediction-update framework

- Update step $p(x_k \mid z_{1:k}) = p(x_k \mid z_k, z_{1:k-1})$

$$\boxed{p(A \mid B, C) = \frac{p(B \mid A, C)\, p(A \mid C)}{p(B \mid C)}} \qquad = \frac{p(z_k \mid x_k, z_{1:k-1})\, p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})}$$

$$\frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \qquad = \underbrace{\frac{\overset{\text{Measurement model}}{p(z_k \mid x_k)}\, \overset{\text{Current prior}}{p(x_k \mid z_{1:k-1})}}{p(z_k \mid z_{1:k-1})}}_{\text{Normalization constant}} \qquad (2)$$

Where $p(z_k \mid z_{1:k-1}) = \int p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})\, dx_k$

Generating estimates

- Knowledge of $p(x_k \mid z_{1:k})$ enables to compute optimal estimate with respect to any criterion. e.g.
  – Minimum mean-square error (MMSE)

$$\hat{x}_{k|k}^{MMSE} \equiv E\left[x_k \mid z_{1:k}\right] = \int x_k\, p(x_k \mid z_{1:k})\, dx_k$$

  – Maximum a-posteriori

$$\hat{x}_{k|k}^{MAP} \equiv \arg\max_{x_k} p(x_k \mid z_k)$$

General prediction-update framework

➔So (1) and (2) give optimal solution for the recursive estimation problem!

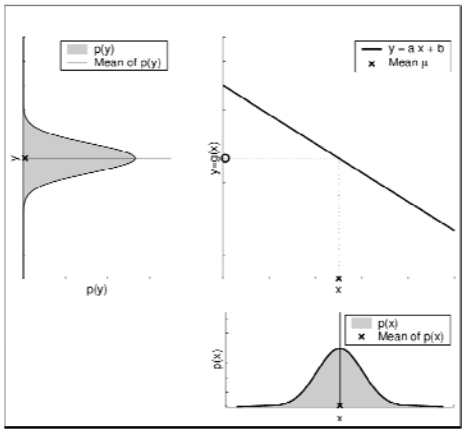
- Unfortunately no… only conceptual solution
  - integrals are intractable…
  - Can only implement the pdf to finite representation!
- However, optimal solution *does* exist for several restrictive cases

Restrictive case #1

- Posterior at each time step is Gaussian
  - Completely described by mean and covariance
- If $p(x_{k-1} \mid z_{1:k-1})$ is Gaussian it can be shown that $p(x_k \mid z_{1:k})$ is also Gaussian provided that:
  - v_k, w_k are Gaussian
  - f_k, h_k are linear

Restrictive case #1

- Why Linear?

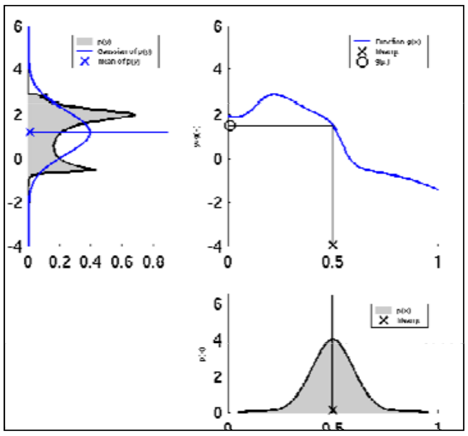


Yacov Hel-Or

$$y = Ax + B \Rightarrow p(y) \sim N\left(A\mu + B, A\Sigma A^T\right)$$

Restrictive case #1

- Why Linear?



Yacov Hel-Or

$$y = g(x) \not\Rightarrow p(y) \sim N(\ )$$

Restrictive case #1

- Linear system with additive noise

$$x_k = F_k x_{k-1} + v_k$$
$$z_k = H_k x_k + w_k$$
$$v_k \sim N(0, Q_k)$$
$$w_k \sim N(0, R_k)$$

- Simple example again

$$f(X, v) = [x + \Delta t \cdot v_x, y + \Delta t \cdot v_y, v_x, v_y] + v \qquad z = h(X, w) = [x, y] + w$$

$$\begin{pmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{F} \begin{pmatrix} x_k \\ y_k \\ v_{x,k-1} \\ v_{y,k-1} \end{pmatrix} + N(0, Q_k) \qquad \begin{pmatrix} x_{obs} \\ y_{obs} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}}_{H} \begin{pmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{pmatrix} + N(0, R_k)$$

The Kalman filter

Rudolf E. Kalman

$$p(x_{k-1} \mid z_{1:k-1}) = N(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$$
$$p(x_k \mid z_{1:k-1}) = N(x_k; \hat{x}_{k|k-1}, P_{k|k-1})$$
$$p(x_k \mid z_{1:k}) = N(x_k; \hat{x}_{k|k}, P_{k|k})$$

$$N(x; \mu, \Sigma) = |2\pi\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

- Substituting into (1) and (2) yields the predict and update equations

The Kalman filter

Predict:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

Update:

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$
$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1} \right)$$
$$P_{k|k} = \left[I - K_k H_k \right] P_{k|k-1}$$

Intuition via 1D example

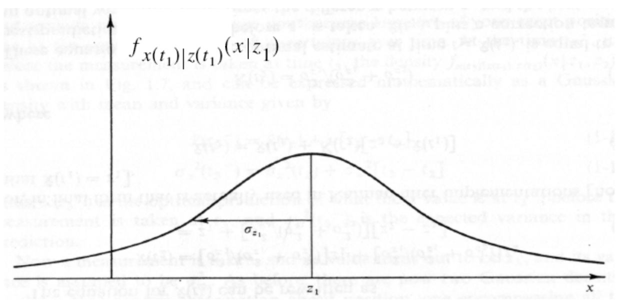
- Lost at sea
  - Night
  - No idea of location
  - For simplicity – let's assume 1D

* Example and plots by Maybeck, *"Stochastic models, estimation and control, volume 1"*

Example – cont'd

- Time t1: Star Sighting
  - Denote x(t1)=z1
- Uncertainty (inaccuracies, human error, etc)
  - Denote σ1 (normal)
- Can establish the conditional probability of x(t1) given measurement z1

Example – cont'd

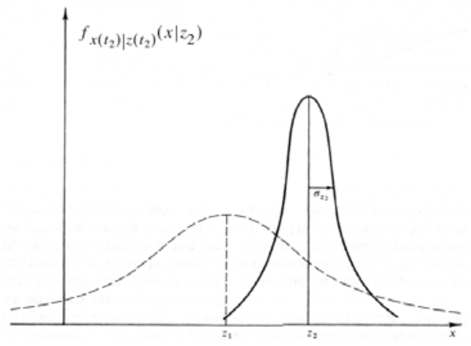


- Probability for any location, based on measurement
- For Gaussian density – 68.3% within ±σ1
- Best estimate of position: Mean/Mode/Median

$$\hat{x}(t_1) = z_1 \qquad \sigma_x^2(t_1) = \sigma_{z_1}^2$$

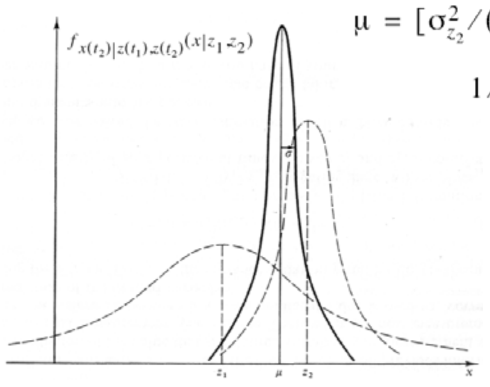
Example – cont'd

- Time t2≅t1: friend (more trained)
  - x(t2)=z2, σ(t2)=σ2
  - Since she has higher skill: σ2<σ1



Example – cont'd

- f(x(t2)|z1,z2) also Gaussian



$$\mu = [\sigma_{z_2}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)]z_2$$

$$1/\sigma^2 = (1/\sigma_{z_1}^2) + (1/\sigma_{z_2}^2)$$

Example – cont’d

$$\mu = [\sigma_{z_2}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)]z_2$$

$$1/\sigma^2 = (1/\sigma_{z_1}^2) + (1/\sigma_{z_2}^2)$$

- σ less than both σ1 and σ2
- σ1= σ2: average
- σ1> σ2: more weight to z2
- Rewrite:

$$\begin{aligned}\hat{x}(t_2) &= [\sigma_{z_2}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)]z_2 \\ &= z_1 + [\sigma_{z_1}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)][z_2 - z_1]\end{aligned}$$

Example – cont’d

- The Kalman update rule:

$$\boxed{\begin{aligned}\underbrace{\hat{x}(t_2)}_{\substack{\text{Best estimate}\\\text{Given z2}\\\text{(a poseteriori)}}} &= \underbrace{\hat{x}(t_1)}_{\substack{\textbf{Best Prediction prior to z2}\\\textbf{(\textit{a priori})}}} + \underbrace{K(t_2)}_{\substack{\textbf{Optimal Weighting}\\\textbf{(\textit{Kalman Gain})}}}\underbrace{[z_2 - \hat{x}(t_1)]}_{\textbf{Residual}} \\ K(t_2) &= \sigma_{z_1}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)\end{aligned}}$$

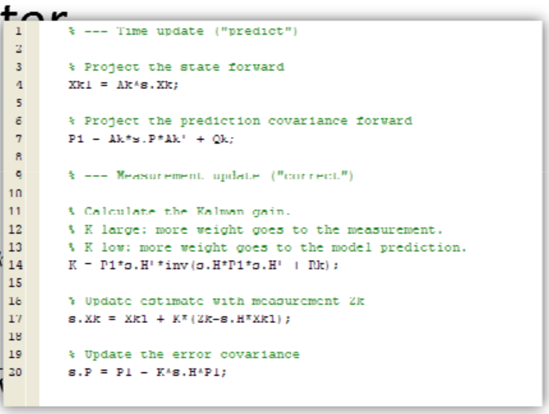
The Kalman fil

Predict:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k$$

Update:

$$S_k = H_k P_{k|k-1} H_k^T + R$$
$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1} \right)$$
$$P_{k|k} = \left[I - K_k H_k \right] P_{k|k-1}$$



```
% --- Time update ("predict")

% Project the state forward
Xk1 = Ak*s.Xk;

% Project the prediction covariance forward
P1 = Ak*s.P*Ak' + Qk;

% --- Measurement update ("correct")

% Calculate the Kalman gain.
% K large: more weight goes to the measurement.
% K low: more weight goes to the model prediction.
K = P1*s.H'*inv(s.H*P1*s.H' + Rk);

% Update estimate with measurement Zk
s.Xk = Xk1 + K*(Zk-s.H*Xk1);

% Update the error covariance
s.P = P1 - K*s.H*P1;
```

$$K(t_2) = \sigma_{z_1}^2 / (\sigma_{z_1}^2 + \sigma_{z_2}^2)$$
$$\hat{x}(t_2) = \hat{x}(t_1) + K(t_2)[z_2 - \hat{x}(t_1)]$$

Kalman gain

$$\boxed{\begin{aligned} S_k &= H_k P_{k|k-1} H_k^T + R_k \\ K_k &= P_{k|k-1} H_k^T S_k^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1} \right) \\ P_{k|k} &= \left[I - K_k H_k \right] P_{k|k-1} \end{aligned}}$$

- Small measurement error:

$$\lim_{\mathbf{R}_k \to 0} K_k = H_k^{-1} \Rightarrow \lim_{\mathbf{R}_k \to 0} \hat{x}_{k|k} = H_k^{-1} z_k$$

- Small prediction error:

$$\lim_{\mathbf{P}_k \to 0} K_k = 0 \Rightarrow \lim_{\mathbf{P}_k \to 0} \hat{x}_{k|k} = \hat{x}_{k|k-1}$$

The Kalman filter

- Pros
  - Optimal closed-form solution to the tracking problem (under the assumptions)
    - No algorithm can do better in a linear-Gaussian environment!
  - All ‘logical’ estimations collapse to a unique solution
  - Simple to implement
  - Fast to execute
- Cons
  - If either the system or measurement model is non-linear $\rightarrow$ the posterior will be non-Gaussian

Restrictive case #2

- The state space (domain) is discrete and finite
- Assume the state space at time k-1 consists of states $x_{k-1}^i, i = 1..N_s$
- Let $\Pr(x_{k-1} = x_{k-1}^i \mid z_{1:k-1}) = w_{k-1|k-1}^i$ be the conditional probability of the state at time k-1, given measurements up to k-1

The Grid-based filter

- The posterior pdf at k-1 can be expressed as sum of delta functions

$$p(x_{k-1} \mid z_{1:k-1}) = \sum_{i=1}^{N_s} w_{k-1|k-1}^{i} \delta(x_{k-1} - x_{k-1}^{i})$$

- Again, substitution into (1) and (2) yields the predict and update equations

The Grid-based filter

- Prediction

$$p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1} \qquad (1)$$

$$p(x_k \mid z_{1:k-1}) = \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} p(x_k^{i} \mid x_{k-1}^{j}) w_{k-1|k-1}^{j} \delta(x_{k-1} - x_{k-1}^{i})$$

$$= \sum_{i=1}^{N_s} w_{k|k-1}^{i} \delta(x_{k-1} - x_{k-1}^{i})$$

$$w_{k|k-1}^{i} = \sum_{j=1}^{N_s} w_{k-1|k-1}^{j} p(x_k^{i} \mid x_{k-1}^{j})$$

- New prior is also weighted sum of delta functions
- New prior weights are reweighting of old posterior weights using state transition probabilities

The Grid-based filter

- Update

$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \qquad (2)$$

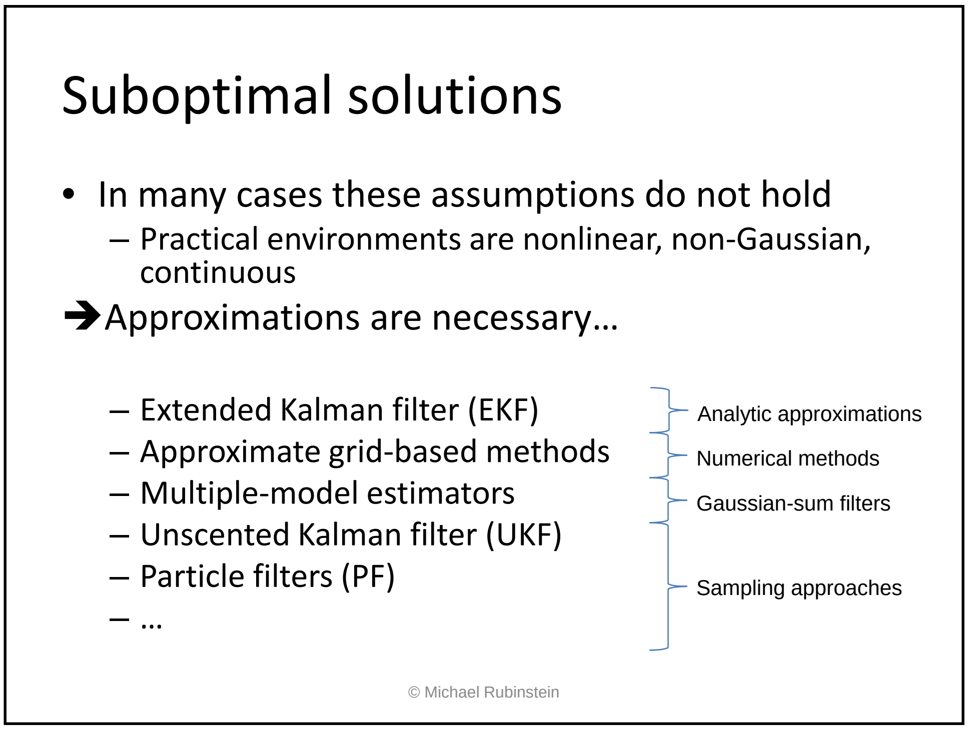
$$p(x_k \mid z_{1:k}) = \sum_{i=1}^{N_s} w_{k|k}^{i} \delta(x_{k-1} - x_{k-1}^{i})$$

$$w_{k|k}^{i} = \frac{w_{k|k-1}^{i} p(z_k \mid x_k^{i})}{\sum_{j=1}^{N_s} w_{k|k-1}^{j} p(z_k \mid x_k^{j})}$$

- Posterior weights are reweighting of prior weights using likelihoods (+ normalization)

The Grid-based filter

- Pros:
  - $p(x_k \mid x_{k-1}), p(z_k \mid x_k)$ assumed known, but no constraint on their (discrete) shapes
  - Easy extension to varying number of states
  - Optimal solution for the discrete-finite environment!
- Cons:
  - Curse of dimensionality
    - Inefficient if the state space is large
  - Statically considers *all* possible hypotheses



Suboptimal solutions

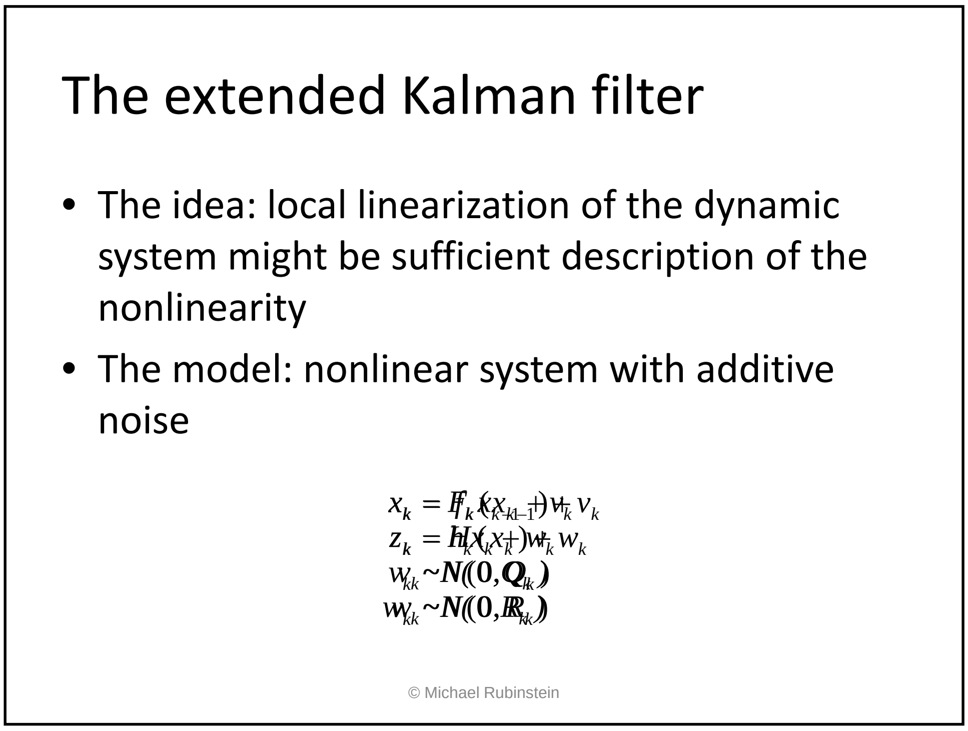
- In many cases these assumptions do not hold
  - Practical environments are nonlinear, non-Gaussian, continuous

➔Approximations are necessary…

  - Extended Kalman filter (EKF) — Analytic approximations
  - Approximate grid-based methods — Numerical methods
  - Multiple-model estimators — Gaussian-sum filters
  - Unscented Kalman filter (UKF)
  - Particle filters (PF)
  - …

  (UKF, PF, …) — Sampling approaches

© Michael Rubinstein



The extended Kalman filter

- The idea: local linearization of the dynamic system might be sufficient description of the nonlinearity
- The model: nonlinear system with additive noise

$$\begin{aligned} x_k &= f_k(x_{k-1}) + v_k \\ z_k &= h_k(x_k) + w_k \\ v_k &\sim N(0, Q_k) \\ w_k &\sim N(0, R_k) \end{aligned}$$

© Michael Rubinstein

The extended Kalman filter

- *f, h* are approximated using a first-order Taylor series expansion (eval at state estimations)

Predict:

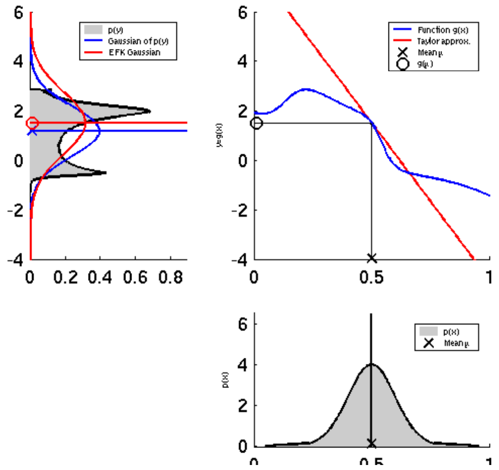
$$\hat{x}_{k|k-1} = f_k(\hat{x}_{k-1|k-1})$$
$$P_{k|k-1} = \hat{F}_k P_{k-1|k-1} \hat{F}_k^T + Q_k$$

Update:

$$S_k = \hat{H}_k P_{k|k-1} \hat{H}_k^T + R_k$$
$$K_k = P_{k|k-1} \hat{H}_k^T S_k^{-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - h_k(\hat{x}_{k|k-1})\right)$$
$$P_{k|k} = \left[I - K_k H_k\right] P_{k|k-1}$$

$$\boxed{\begin{aligned} \hat{F}_k[i,j] &= \left.\frac{\partial f_k[i]}{\partial x_k[j]}\right|_{x_k = \hat{x}_{k-1|k-1}} \\ \hat{H}_k[i,j] &= \left.\frac{\partial h_k[i]}{\partial x_k[j]}\right|_{x_k = \hat{x}_{k|k-1}} \end{aligned}}$$

The extended Kalman filter



The extended Kalman filter

- Pros
  - Good approximation when models are near-linear
  - Efficient to calculate

  (de facto method for navigation systems and GPS)
- Cons
  - Only approximation (optimality not proven)
  - Still a single Gaussian approximations
    - Nonlinearity → non-Gaussianity (e.g. bimodal)
  - If we have multimodal hypothesis, and choose incorrectly – can be difficult to recover
  - Inapplicable when *f,h* discontinuous

Particle filtering

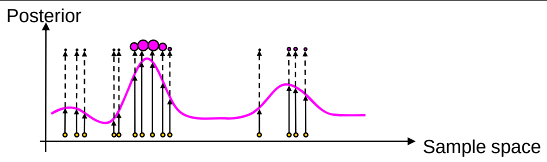
- Family of techniques
  - Condensation algorithms (MacCormick&Blake, ‘99)
  - Bootstrap filtering (Gordon et al., ‘93)
  - Particle filtering (Carpenter et al., ‘99)
  - Interacting particle approximations (Moral ‘98)
  - Survival of the fittest (Kanazawa et al., ‘95)
  - Sequential Monte Carlo methods (SMC,SMCM)
  - SIS, SIR, ASIR, RPF, ….

- Statistics introduced in 1950s. Incorporated in vision in Last decade

Particle filtering

- Many variations, one general concept:

Represent the posterior pdf by a set of randomly chosen weighted samples (particles)



- Randomly Chosen = Monte Carlo (MC)
- As the number of samples become very large – the characterization becomes an equivalent representation of the true pdf

Particle filtering

- Compared to previous methods
  - Can represent any arbitrary distribution
    - multimodal support
  - Keep track of many hypotheses as there are particles
  - **Approximate representation of complex model rather than exact representation of simplified model**
- The basic building-block: *Importance Sampling*