EECE 5550
Mobile Robotics

Lecture 8: Bayes Filter Implementations

Derya Aksaray

Assistant Professor

Department of Electrical and Computer Engineering

Northeastern University

Recap: Filtering

Why do we use filtering?

- The process of finding the "best estimate" from noisy data by "filtering out" the noise

- **Online belief updates:** a principled way to incorporate noisy information from sensor measurements, which can change our prior belief, in an online fashion.

Some examples:

- Tracking targets such as aircraft, missiles using RADAR.
- Robot localization from range sensors.
- Determination of planet orbit parameters from limited earth observations.

Plan of the day

Last time: Probabilistic robotics and the Bayes Filter

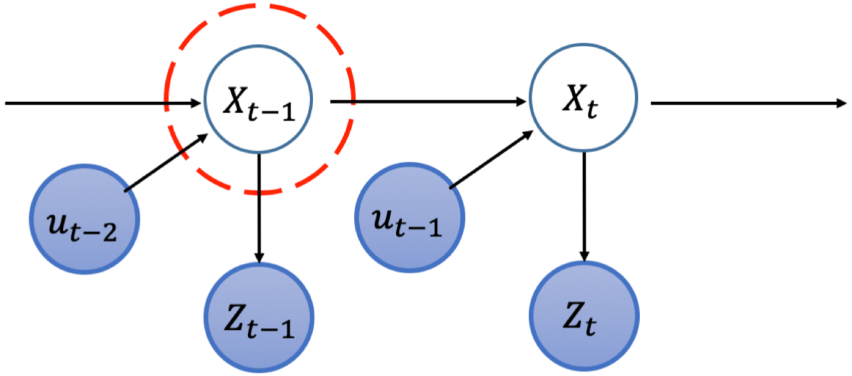
Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})\ dX_{t-1}$$

- **Update** belief after incorporating measurement Z_t at current state X_t:

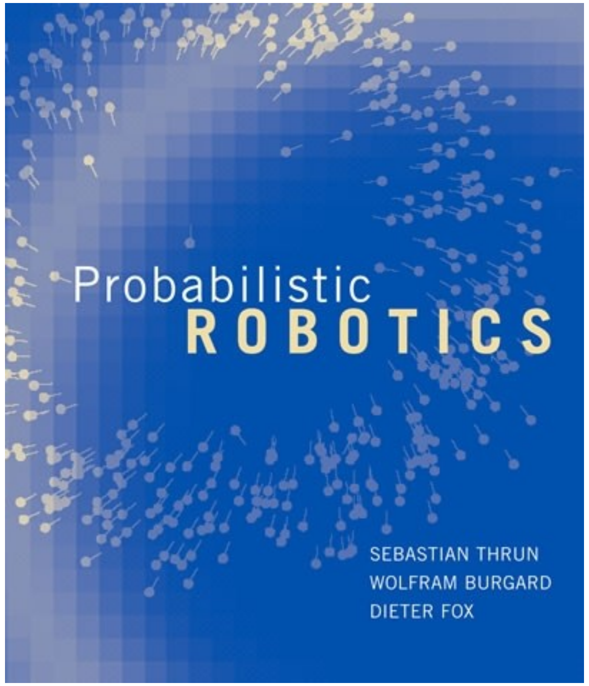
$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})\ dX_t}$$



Today: Three Implementations of the Bayes Filter

1. Linear-Gaussian case: The Kalman Filter (KF)
2. Nonlinear-Gaussian case: The extended Kalman Filter(EKF)
3. Nonlinear, non-Gaussian case: The Particle Filter (PF)

References



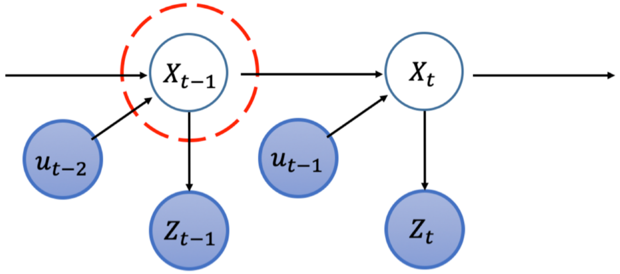
Chapters 3 & 4

Recap: Bayes' filter

Markov Assumptions:

$$p(X_t|X_{0:t-1}, u_{0:t-1}) = p(X_t|X_{t-1}, u_{t-1})$$

$$p(Z_t|X_t, u_{0:t-1}, Z_{0:t-1}) = p(Z_t|X_t)$$



Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})\ dX_{t-1}$$

- **Update** belief after incorporating measurement Z_t at current state X_t:

$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})\ dX_t} = \eta p(Z_t|X_t) \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})\ dX_{t-1}$$

Kalman Filter: An instance of Bayes' Filter

$$bel(x_t) = p(x_t|u_{0:t-1}, z_{1:t}) = \eta p(z_t|x_t) \int p(x_t|x_{t-1}, u_{t-1}) \cdot \underbrace{p(x_{t-1}|u_{0:t-2}, z_{1:t-1})}_{bel(x_{t-1})} \, dx_{t-1}$$

Linear observations with Gaussian noise (measurement model)

$$z_t = C_t x_t + \delta_t$$

With noise $\delta_t \sim N(0, Q_t)$

Linear dynamics with Gaussian noise (Process model)

$$x_{t+1} = A_t x_t + B_t u_t + \epsilon_t$$

With noise $\epsilon_t \sim N(0, R_t)$

+ Initial belief is Gaussian $bel(x_o) \sim N(\mu_0, \Sigma_0)$

The Kalman Filter (KF)

- Two assumptions inherited from Bayes' filter
- Linear dynamics and observation models
  - Recall that if X is Gaussian then Y=AX+b is also Gaussian. However, it is not true in general for Y=f(X).
- Initial belief is Gaussian.
  - The belief remains Gaussian after each propagation and update step. Hence, we only have to worry about how the mean and the covariance of the belief evolve recursively with each step.
- Noise variables δ_t are Gaussian and independent and identically distributed, $N(0, Q_t)$.
- Noise variables ϵ_t are Gaussian and independent and identically distributed, $N(0, R_t)$.
  - An iid noise: "**A sequence of random variables for which each element of the sequence has the same probability distribution as the others, and all values are mutually independent**."
  - Recursive updates of mean and covariance becomes simpler.

Kalman Filter: An instance of Bayes' Filter

$$bel(x_t) = p(x_t | u_{0:t-1}, z_{1:t}) = \eta p(z_t | x_t) \overbrace{\int p(x_t | x_{t-1}, u_{t-1}) \cdot \underbrace{p(x_{t-1} | u_{0:t-2}, z_{1:t-1})}_{bel(x_{t-1})} \, dx_{t-1}}^{\overline{bel}(x_t)}$$

Assumptions guarantee that if the belief before the prediction step is Gaussian

then the prior belief after the prediction step will be Gaussian

and the posterior belief (after the update step) will be Gaussian.

Kalman Filter: Propagation step

Given: Current belief $x_t \sim N(\mu_t, \Sigma_t)$ and control u_t

Find: Predicted belief $N(\mu_{t+1}, \Sigma_{t+1})$ for next state x_{t+1}

Process model:

$$x_{t+1} = \underbrace{A_t x_t + B_t u_t}_{\text{Affine transformation of } x_t} + \epsilon_t, \qquad \epsilon_t \sim N(0, R_t)$$

$$\Rightarrow A_t x_t + B_t u_t \sim N(A_t \mu_t + B_t u_t,\ A_t \Sigma_t A_t^T)$$

$$\Rightarrow x_{t+1} \sim N(A_t \mu_t + B_t u_t,\ A_t \Sigma_t A_t^T) + N(0, R_t)$$

$$\Rightarrow x_{t+1} \sim N(\underbrace{A_t \mu_t + B_t u_t}_{\hat{\mu}_{t+1}},\ \underbrace{A_t \Sigma_t A_t^T + R_t}_{\hat{\Sigma}_{t+1}})$$

Kalman Filter: Measurement update step

Given: Prior belief $x_t \sim N\left(\hat{\mu}_t, \hat{\Sigma}_t\right)$ for current state x_t, observation z_t

Find: Posterior belief $x_t | z_t \sim N(\mu_t, \Sigma_t)$

Measurement model:

$$z_t = C_t x_t + \delta_t, \qquad \delta_t \sim N(0, Q_t)$$

Bayes' Rule:

$$p(x_t | z_t) \propto p(z_t | x_t) p(x_t)$$

Recall multivariate Gaussian distribution for $x \sim N(\mu, \Sigma)$

$$p(x) = (2\pi)^{-\frac{n}{2}} \det(\Sigma)^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

Also, $\delta_t \sim N(0, Q_t)$

NB: If we condition on x_t, then the only random variable in the measurement model is δ_t.

Therefore:

$$p(x_t | z_t) \propto p(\delta_t) p(x_t)$$

Substituting for δ_t:

$$p(x_t | z_t) \propto \exp\left(-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right) \exp\left(-\frac{1}{2}(x_t - \hat{\mu}_t)^T \hat{\Sigma}_t^{-1} (x_t - \hat{\mu}_t)\right)$$

Kalman Filter: Measurement update step (cont'd)

$$p(x_t|z_t) \propto \exp\left(-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right)\exp\left(-\frac{1}{2}(x_t - \hat{\mu}_t)^T \hat{\Sigma}_t^{-1}(x_t - \hat{\mu}_t)\right)$$

Combine exponentials and collect like terms:

$$p(x_t|z_t) \propto \exp\left(-\frac{1}{2}\underbrace{\left[x_t^T\left(C_t^T Q_t^{-1} C_t + \hat{\Sigma}_t^{-1}\right)x_t - 2\left(C_t^T Q_t^{-1} z_t + \hat{\Sigma}_t^{-1}\hat{\mu}_t\right)^T x_t + \left(z_t^T Q_t^{-1} z_t + \hat{\mu}_t^T \hat{\Sigma}_t^{-1}\hat{\mu}_t\right)\right]}_{\triangleq L(x_t)}\right)$$

NB: $L(x)$ is quadratic
$\Rightarrow$ Posterior $p(x_t|z_t)$ is Gaussian!

- The mean and covariance of this distribution are the minimum and curvature of $L(x_t)$.
- First derivative: info related to posterior mean μ_t
- Second derivative: info related to posterior covariance $\Sigma_t = \left(C_t^T Q_t^{-1} C_t + \hat{\Sigma}_t^{-1}\right)^{-1}$

Kalman Filter: Measurement update step (cont'd)

$$L(x) = x^T\left(C_t^T Q_t^{-1} C_t + \hat{\Sigma}_t^{-1}\right)x - 2\left(C_t^T Q_t^{-1} z_t + \hat{\Sigma}_t^{-1}\hat{\mu}_t\right)^T x + \left(z_t^T Q_t^{-1} z_t + \hat{\mu}_t^T \hat{\Sigma}_t^{-1}\hat{\mu}_t\right)$$

$$\Rightarrow \frac{\partial L}{\partial x}(x) = 2\left(C_t^T Q_t^{-1} C_t + \hat{\Sigma}_t^{-1}\right)x - 2\left(C_t^T Q_t^{-1} z_t + \hat{\Sigma}_t^{-1}\hat{\mu}_t\right)$$

Set $\frac{\partial L}{\partial x} = 0$ to obtain an expression for posterior mean μ_t. After rearranging:

$$C_t^T Q_t^{-1}\underbrace{(z_t - C_t \mu_t)}_{\text{Measurement residual for posterior mean}} = \hat{\Sigma}_t^{-1}\underbrace{(\mu_t - \hat{\mu}_t)}_{\text{Difference of posterior and prior means}}$$

Let's rewrite the left-hand side above in terms of prior mean:

$$C_t^T Q_t^{-1}(z_t - C_t\hat{\mu}_t + C_t\hat{\mu}_t - C_t\mu_t) = \hat{\Sigma}_t^{-1}(\mu_t - \hat{\mu}_t)$$

$$\Leftrightarrow C_t^T Q_t^{-1}(z_t - C_t\hat{\mu}_t) = \left(C_t^T Q_t^{-1} C_t + \hat{\Sigma}_t^{-1}\right)(\mu_t - \hat{\mu}_t)$$

Kalman Filter: Measurement update step (cont'd)

$$\Leftrightarrow \; C_t^T Q_t^{-1} \underbrace{(z_t - C_t\hat{\mu}_t)}_{\text{Measurement residual at prior mean}} = \underbrace{(C_t^T Q_t^{-1} C_t + \hat{\Sigma}_t^{-1})}_{\Sigma_t^{-1}} \underbrace{(\mu_t - \hat{\mu}_t)}_{\text{Difference of posterior and prior means}}$$

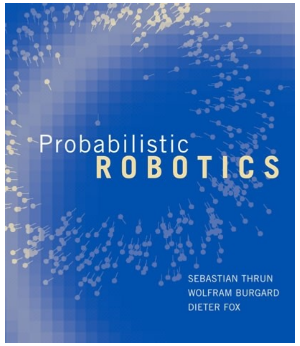
Rearranging the above:

$$\mu_t = \hat{\mu}_t + \underbrace{\Sigma_t C_t^T Q_t^{-1}}_{\text{“Kalman gain”}} \underbrace{(z_t - C_t\hat{\mu}_t)}_{\text{“innovation”}}$$

Kalman Filter: Measurement update step

Given: Prior belief $x_t \sim N\left(\hat{\mu}_t, \hat{\Sigma}_t\right)$ for current state x_t observation z_t

Find: Posterior belief $x_t | z_t \sim N(\mu_t, \Sigma_t)$



Section 3.2.4

Measurement model:

$$z_t = C_t x_t + \delta_t, \qquad \delta_t \sim N(0, Q_t)$$

1. Compute posterior covariance:

$$\Sigma_t = \left(C_t^T Q_t^{-1} C_t + \hat{\Sigma}_t^{-1}\right)^{-1}$$

2. Compute Kalman gain:

$$K_t = \Sigma_t C_t^T Q_t^{-1}$$

3. Compute posterior mean:

$$\mu_t = \hat{\mu}_t + K_t (z_t - C_t \hat{\mu}_t)$$

An alternative formulation

With a bit of algebra (cf. Sec. 3.2 of *Probabilistic Robotics*), one can an *alternative but equivalent* form of the Kalman filter measurement update

Version 1 (previous slide)

1. Compute posterior covariance:

$$\Sigma_t = \left(C_t^T Q_t^{-1} C_t + \hat{\Sigma}_t^{-1}\right)^{-1}$$

2. Compute Kalman gain:

$$K_t = \Sigma_t C_t^T Q_t^{-1}$$

3. Compute posterior mean:

$$\mu_t = \hat{\mu}_t + K_t(z_t - C_t \hat{\mu}_t)$$

Version 2 ("standard" form)

1. Compute Kalman gain:

$$K_t = \hat{\Sigma}_t C_t^T \left(C_t \hat{\Sigma}_t C_t^T + Q_t\right)^{-1}$$

2. Compute posterior mean:

$$\mu_t = \hat{\mu}_t + K_t(z_t - C_t \hat{\mu}_t)$$

3. Compute posterior covariance:

$$\Sigma_t = (I - K_t C_t)\hat{\Sigma}_t$$

Three inverses

One inverse

The Kalman Filter: Complete Algorithm

Predict: Given belief $x_t \sim N(\mu_t, \Sigma_t)$ for the current state x_t, control u_t, and process model:

$$x_{t+1} = A_t x_t + B_t u_t + \epsilon_t, \qquad \epsilon_t \sim N(0, R_t)$$

the belief for the *next* state x_{t+1} is $N(\mu_{t+1}, \Sigma_{t+1})$, where:

$$\mu_{t+1} = A_t \mu_t + B_t u_t, \qquad \Sigma_{t+1} = A_t \Sigma_t A_t^T + R_t$$

Update: Given prior belief $x_t \sim N\left(\hat{\mu}_t, \hat{\Sigma}_t\right)$ for the current state x_t, measurement z_t, and measurement model:

$$z_t = C_t x_t + \delta_t, \qquad \delta_t \sim N(0, Q_t)$$

the posterior belief for x_t given z_t is $N(\mu_t, \Sigma_t)$, where:

$$\mu_t = \hat{\mu}_t + K_t(z_t - C_t \hat{\mu}_t), \qquad \Sigma_t = (I - K_t C_t)\hat{\Sigma}_t$$

and

$$K_t = \hat{\Sigma}_t C_t^T \left(C_t \hat{\Sigma}_t C_t^T + Q_t\right)^{-1}$$

Kalman Filter with 1D state

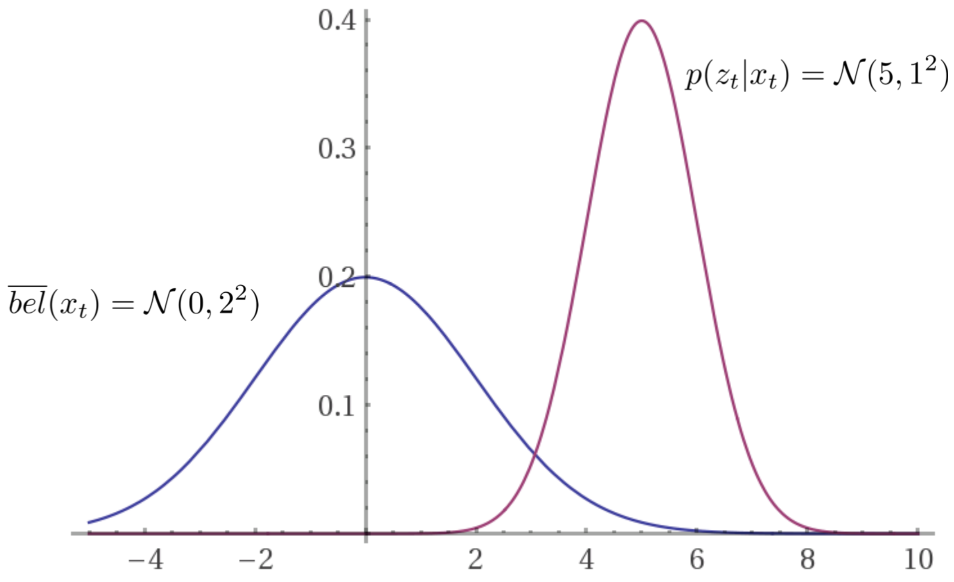
Suppose your belief after the prediction step is

$\overline{bel}(x_t) = \mathcal{N}(0, 2^2)$

Suppose your measurement model is $z_t = x_t + \delta_t$
with , $\delta_t \sim N(0,1^2)$

Suppose your first noisy measurement is $z_0 = 5$

Q: What is the mean and covariance of $bel(x_t)$?

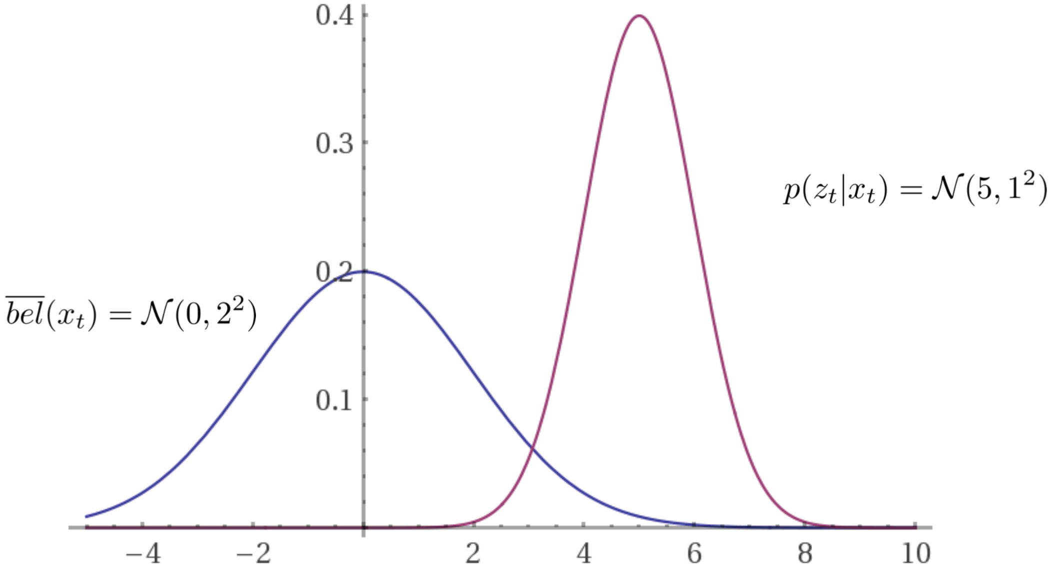


Kalman Filter with 1D state

$$z_0 = 5$$

$z_t = x_t + \delta_t$ with , $\delta_t \sim N(0,1^2)$

- $K_t = \hat{\Sigma}_t C_t^T \left(C_t \hat{\Sigma}_t C_t^T + Q_t \right)^{-1} = 2^2(4+1)^{-1} = \frac{4}{5}$
- $\mu_t = \hat{\mu}_t + K_t(z_t - C_t \hat{\mu}_t) = 0 + \frac{4}{5}(5-0) = 4$
- $\Sigma_t = (I - K_t C_t)\hat{\Sigma}_t = \left(1 - \frac{4}{5}\right) 4 = \frac{4}{5}$

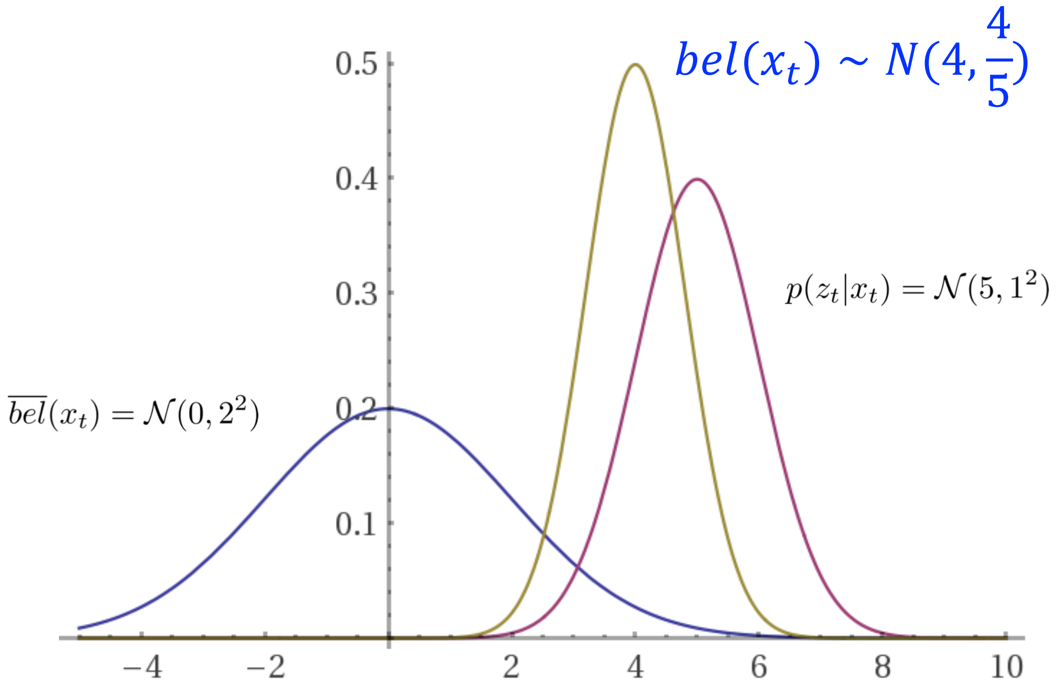


Kalman Filter with 1D state

$z_0 = 5$

$z_t = x_t + \delta_t$ with , $\delta_t \sim N(0,1^2)$

- $K_t = \hat{\Sigma}_t C_t^T \left(C_t \hat{\Sigma}_t C_t^T + Q_t\right)^{-1} = 2^2(4+1)^{-1} = \frac{4}{5}$
- $\mu_t = \hat{\mu}_t + K_t(z_t - C_t\hat{\mu}_t) = 0 + \frac{4}{5}(5-0) = 4$
- $\Sigma_t = (I - K_t C_t)\hat{\Sigma}_t = \left(1 - \frac{4}{5}\right) 4 = \frac{4}{5}$



New observations, no matter how noisy, always **reduce uncertainty** in the posterior.

The mean of the posterior, on the other hand,
only changes when there is a nonzero prediction residual.

Plan of the day

Last time: Probabilistic robotics and the Bayes Filter

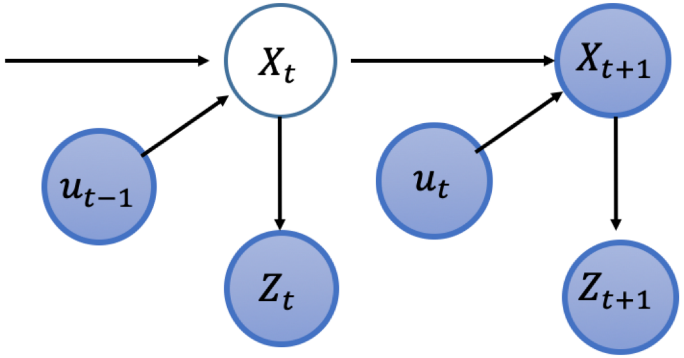
Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})\ dX_{t-1}$$

- **Update** belief after incorporating measurement Z_t at current state X_t:

$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})\ dX_t}$$



Today: Three Implementations of the Bayes Filter

1. Linear-Gaussian case: The Kalman Filter (KF)
2. Nonlinear-Gaussian case: The extended Kalman Filter(EKF)
3. Nonlinear, non-Gaussian case: The Particle Filter (PF)

The Extended Kalman Filter (EKF)

$$bel(x_t) = p(x_t|u_{0:t-1}, z_{1:t}) = \eta p(z_t|x_t) \overbrace{\int p(x_t|x_{t-1}, u_{t-1}) \cdot \underbrace{p(x_{t-1}|u_{0:t-2}, z_{1:t-1})}_{bel(X_{t-1})} \, dx_{t-1}}^{\overline{bel}(x_t)}$$

Nonlinear observations with Gaussian noise (measurement model)

$$z_t = h_t(x_t) + \delta_t$$

With noise $\delta_t \sim N(0, Q_t)$

Nonlinear dynamics with Gaussian noise (Process model)

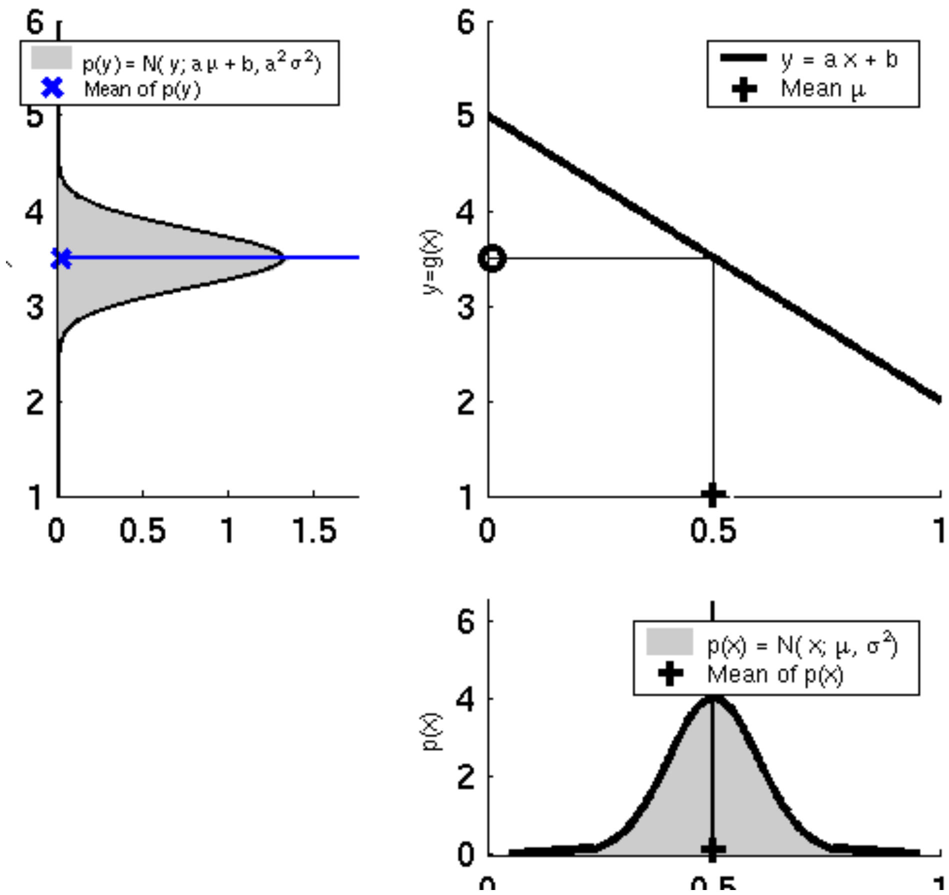
$$x_{t+1} = g_t(x_t, u_t) + \epsilon_t$$

With noise $\epsilon_t \sim N(0, R_t)$

+ Initial belief is Gaussian $bel(x_o) \sim N(\mu_0, \Sigma_0)$

Does the posterior $bel(x_t)$ remain Gaussian?
NO!

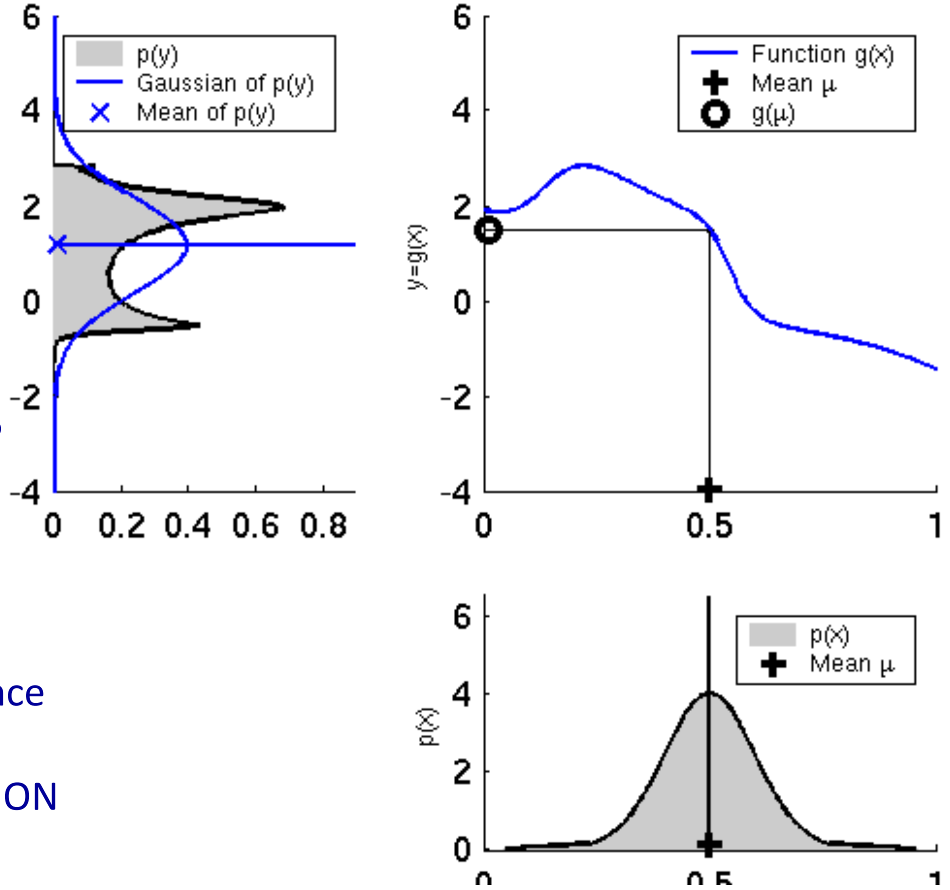
Linearity assumption



If $y = ax + b$
and $x \sim \mathcal{N}(\mu, \sigma^2)$

then $y \sim \mathcal{N}(a\mu + b, a^2\sigma^2)$

Nonlinear functions



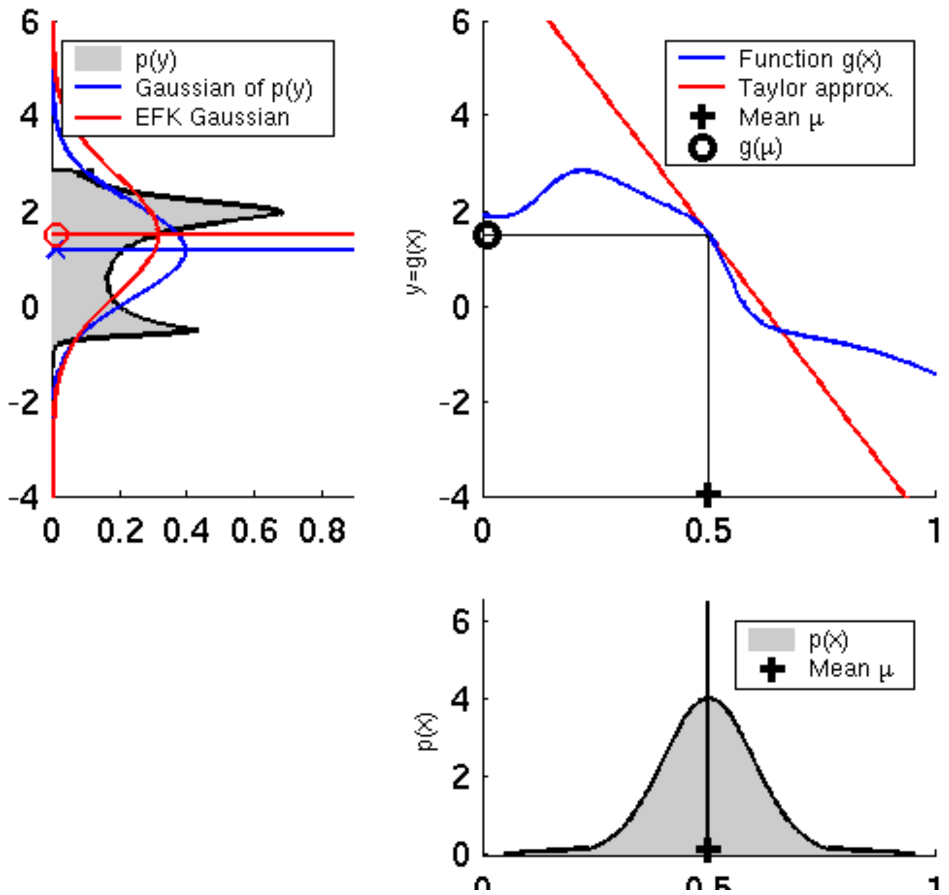
If $y = g(x)$
and $x \sim \mathcal{N}(\mu, \sigma^2)$
then y is not necessarily distributed as a Gaussian.

How to approximate p(y) using a single Gaussian, without having a formula for p(y)?

IDEA #1: MONTE CARLO SAMPLING
- Sample many x
- Pass them through y = g(x)
- Compute the empirical mean and covariance

IDEA #2: LINEARIZE THE NONLINEAR FUNCTION
Then y = Gx +c, so p(y) is a Gaussian

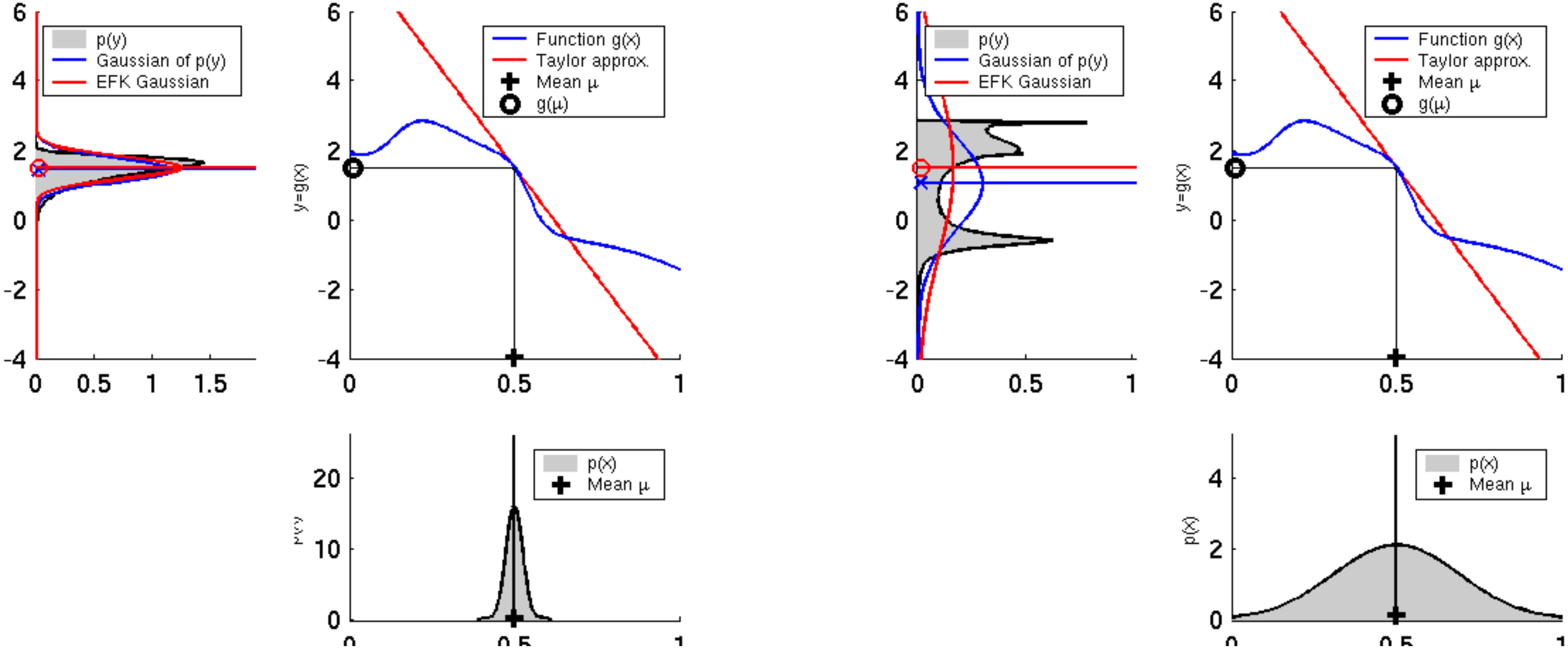
Linearization



More efficient than sampling and computing empirical Gaussian.

Linearization

The quality of the linearization approximation depends on the uncertainty of p(x) but also on the shape of the nonlinearity g(x).



Extended Kalman Filter: Propagation step

Given: Current belief $x_t \sim N(\mu_t, \Sigma_t)$ and control u_t
Find: Predicted belief $N(\mu_{t+1}, \Sigma_{t+1})$ for next state x_{t+1}

Approximate mean using *exact* state propagation model:

$$\mu_{t+1} = g_t(\mu_t, u_t)$$

Approximate covariance using *local linearization* of g_t about the current mean μ_t:

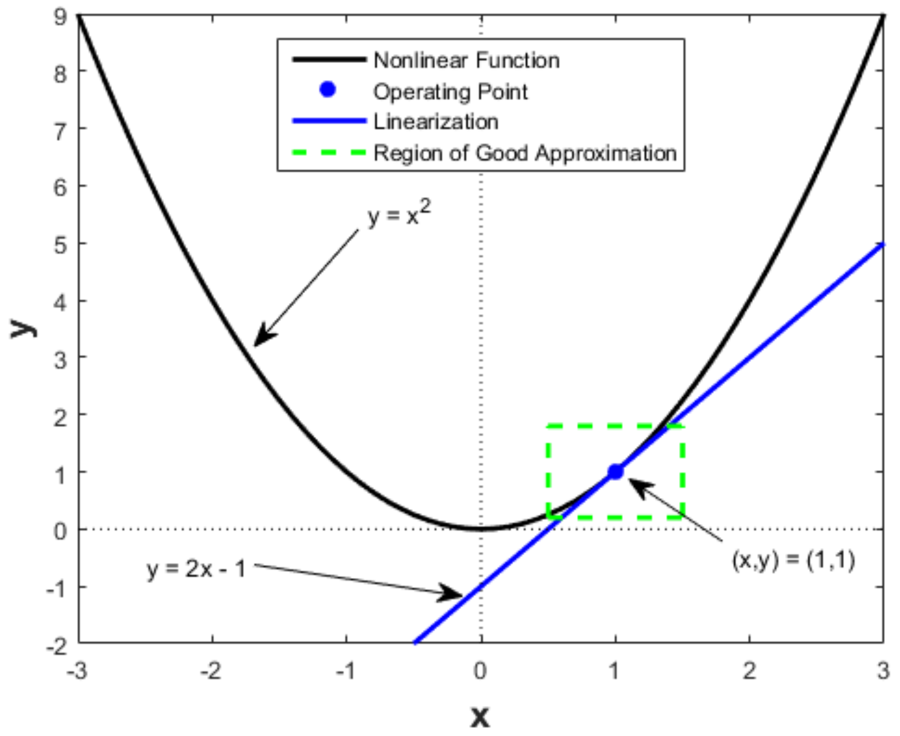
$$\Sigma_{t+1} = G_t \Sigma_t G_t^T + R_t$$

where

$$G_t \triangleq \frac{\partial g_t}{\partial x}(x, u),$$

Jacobian

$$g_t(x, u_t) = g_t(\mu_t, u_t) + G_t(x - \mu_t) + O(x - \mu_t)^2$$



Extended Kalman Filter: Measurement update step

Given: Prior belief $x_t \sim N(\hat{\mu}_t, \hat{\Sigma}_t)$ for current state x_t observation z_t

Find: Posterior belief $x_t | z_t \sim N(\mu_t, \Sigma_t)$

Linearize measurement model about current mean estimate μ_t:

$$h_t(x) \approx h_t(\mu_t) + H_t(x - \mu_t) + O(x - \mu_t)^2$$

where

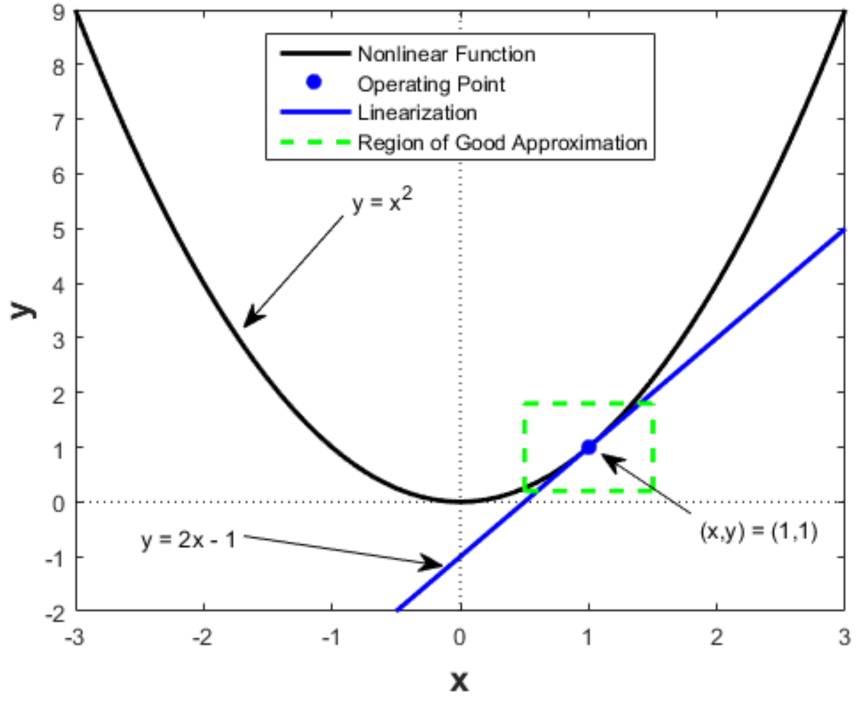
$$H_t \triangleq \frac{\partial h_t}{\partial x}(x).$$

1. Compute Kalman gain:

$$K_t = \hat{\Sigma}_t H_t^T \left(H_t \hat{\Sigma}_t H_t^T + Q_t\right)^{-1}$$

2. Approximate posterior mean and covariance:

$$\mu_t = \hat{\mu}_t + K_t(z_t - h_t(\hat{\mu}_t)), \qquad \Sigma_t = (I - K_t H_t)\hat{\Sigma}_t$$



The Extended Kalman Filter: Complete Algorithm

Predict: Given belief $x_t \sim N(\mu_t, \Sigma_t)$ for the current state x_t, control u_t, and process model:

$$x_{t+1} = g_t(x_t, u_t) + \epsilon_t, \qquad \epsilon_t \sim N(0, R_t)$$

the belief for the *next* state x_{t+1} is approximated by $N(\mu_{t+1}, \Sigma_{t+1})$, where:

$$\mu_{t+1} = g_t(\mu_t, u_t), \quad \Sigma_{t+1} = G_t \Sigma_t G_t^T + R_t, \quad G_t \triangleq \frac{\partial g_t}{\partial x}(x, u)$$

Update: Given prior belief $x_t \sim N\left(\hat{\mu}_t, \hat{\Sigma}_t\right)$ for the current state x_t, measurement z_t, and measurement model:

$$z_t = h_t(x_t) + \delta_t, \qquad \delta_t \sim N(0, Q_t)$$

the posterior belief for x_t given z_t is approximated by $N(\mu_t, \Sigma_t)$, where:

$$\mu_t = \hat{\mu}_t + K_t(z_t - h_t(\hat{\mu}_t)), \qquad \Sigma_t = (I - K_t H_t)\hat{\Sigma}_t, \qquad K_t = \hat{\Sigma}_t H_t^T \left(H_t \hat{\Sigma}_t H_t^T + Q_t\right)^{-1}$$

and

$$H_t \triangleq \frac{\partial h_t}{\partial x}(x)$$

KF

Predict: Given belief $x_t \sim N(\mu_t, \Sigma_t)$ for the current state x_t, control u_t, and process model:

$$x_{t+1} = A_t x_t + B_t u_t + \epsilon_t, \qquad \epsilon_t \sim N(0, R_t)$$

the belief for the *next* state x_{t+1} is $N(\mu_{t+1}, \Sigma_{t+1})$, where:

$$\mu_{t+1} = A_t \mu_t + B_t u_t, \quad \Sigma_{t+1} = A_t \Sigma_t A_t^T + R_t$$

Update: Given prior belief $x_t \sim N\left(\hat{\mu}_t, \hat{\Sigma}_t\right)$ for the current state x_t, measurement z_t, and mea

$$z_t = C_t x_t + \delta_t, \qquad \delta_t \sim N(0, Q_t)$$

the posterior belief for x_t given z_t is $N(\mu_t, \Sigma_t)$, where:

$$\mu_t = \hat{\mu}_t + K_t(z_t - C_t \hat{\mu}_t), \qquad \Sigma_t = (I - K_t C_t)\hat{\Sigma}_t$$

and

$$K_t = \hat{\Sigma}_t C_t^T \left(C_t \hat{\Sigma}_t C_t^T + Q_t\right)^{-1}$$

The Extended Kalman Filter: Fun Facts

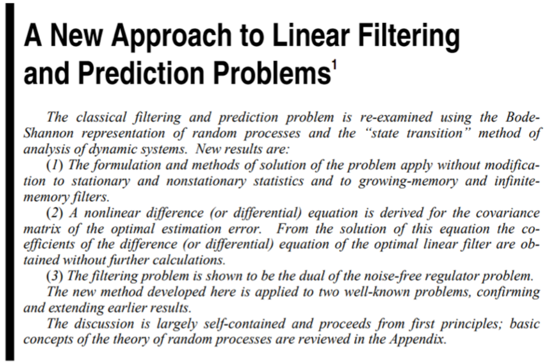
Probably *the* single most important state estimation algorithm of all time

- Kalman supposedly had difficulty publishing: the KF was thought “too good to be true”

- Applications in robotics, computer vision, aviation, astronautics, signal processing, economics, …

- **In particular**: Foundation of the primary guidance, navigation, and control systems (PGNCS) in Apollo Project spacecraft

R.E. Kalman receives the National Medal of Science



A New Approach to Linear Filtering and Prediction Problems[1]

The classical filtering and prediction problem is re-examined using the Bode-Shannon representation of random processes and the “state transition” method of analysis of dynamic systems. New results are:

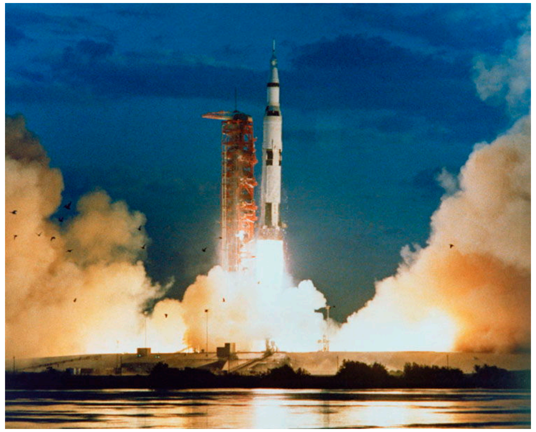
(1) The formulation and methods of solution of the problem apply without modification to stationary and nonstationary statistics and to growing-memory and infinite-memory filters.

(2) A nonlinear difference (or differential) equation is derived for the covariance matrix of the optimal estimation error. From the solution of this equation the coefficients of the difference (or differential) equation of the optimal linear filter are obtained without further calculations.

(3) The filtering problem is shown to be the dual of the noise-free regulator problem.

The new method developed here is applied to two well-known problems, confirming and extending earlier results.

The discussion is largely self-contained and proceeds from first principles; basic concepts of the theory of random processes are reviewed in the Appendix.



Kalman Filtering, summary

Algorithm Kalman_filter$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

$\bar{\mu}_t = A_t \, \mu_{t-1} + B_t \, u_t$

$\bar{\Sigma}_t = A_t \, \Sigma_{t-1} \, A_t^T + R_t$

$K_t = \bar{\Sigma}_t \, C_t^T (C_t \, \bar{\Sigma}_t \, C_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \, \bar{\mu}_t)$

$\Sigma_t = (I - K_t \, C_t) \, \bar{\Sigma}_t$

return μ_t, Σ_t

Algorithm Extended_Kalman_filter$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

$\bar{\mu}_t = g(u_t, \mu_{t-1})$

$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$

$K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$

$\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

return μ_t, Σ_t

Kalman's original paper

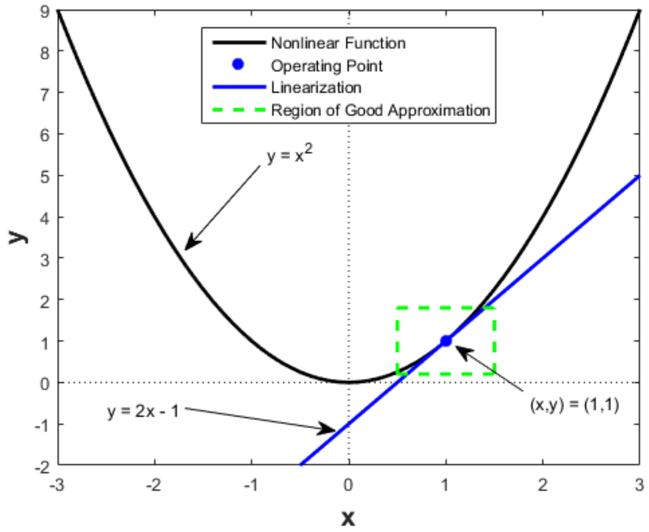
http://160.78.24.2/Public/Kalman/Kalman1960.pdf

NASA paper on Kalman filter for spacecrafts

https://ntrs.nasa.gov/api/citations/19860003843/downloads/19860003843.pdf

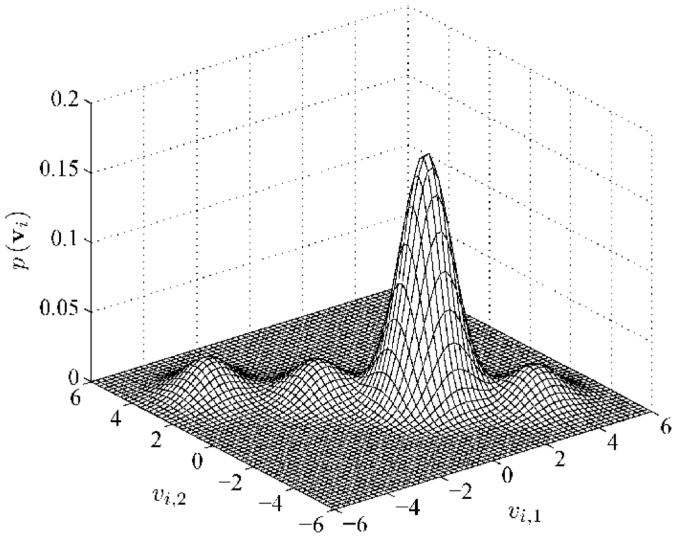
Kalman filter practicalities

The EKF is a *compact*, *parametric* approximation of the Bayes Filter for nonlinear systems with Gaussian beliefs



BUT: Crucially depends upon two key assumptions:

- Local linearization provides a good approximation of the nonlinear state and measurement models g_t and h_t
- A Gaussian distribution is a good model of the true uncertainty



$\Rightarrow$ Kalman filtering works best for systems with:

- Unimodal (true) distributions (priors + posteriors)
- Concentrated distributions (relative to scale of nonlinearities)

Plan of the day

Last time: Probabilistic robotics and the Bayes Filter

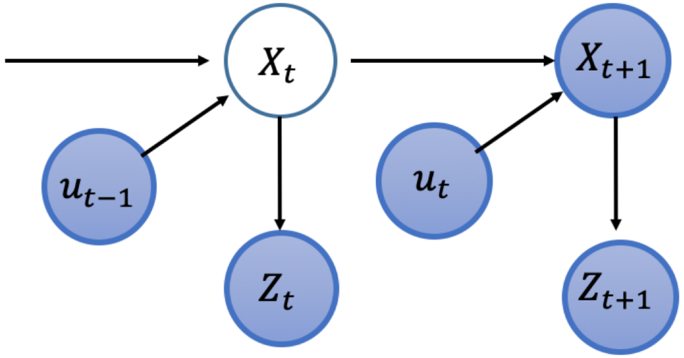
Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})\ dX_{t-1}$$

- **Update** belief after incorporating measurement Z_t at current state X_t:

$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})\ dX_t}$$

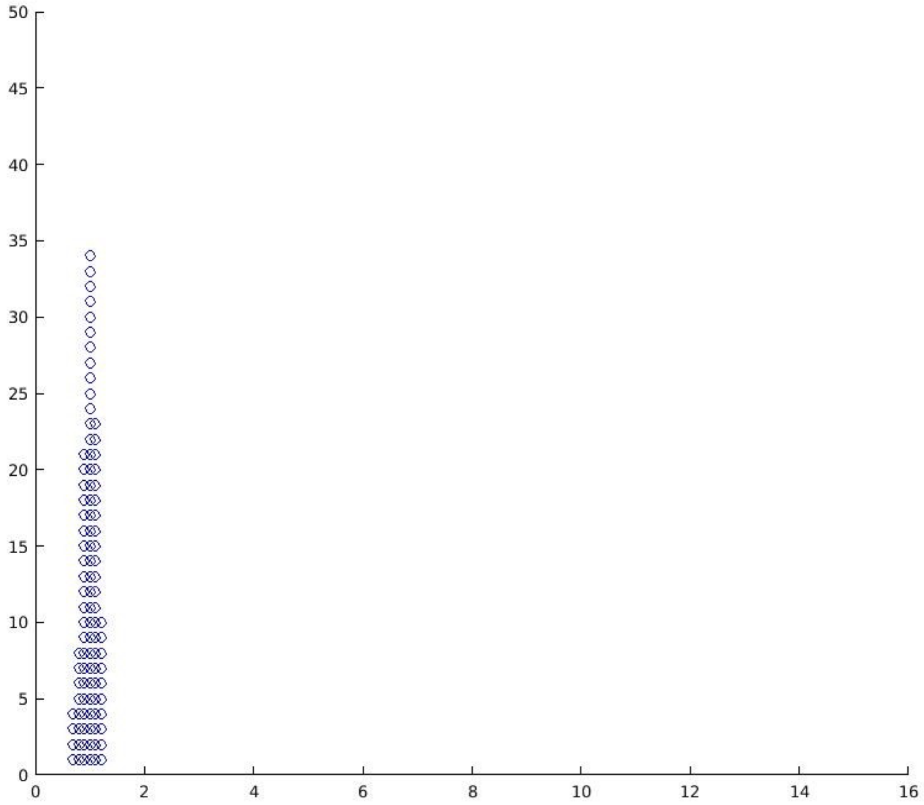


Today: Three Implementations of the Bayes Filter

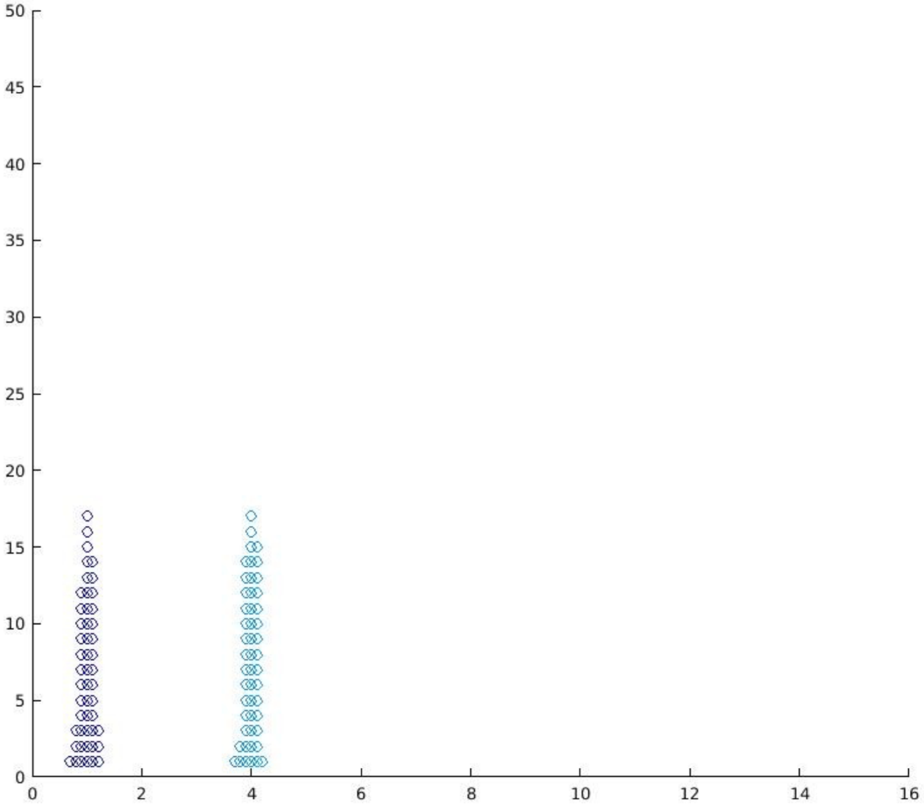
1. Linear-Gaussian case: The Kalman Filter (KF)
2. Nonlinear-Gaussian case: The extended Kalman Filter(EKF)
3. Nonlinear, non-Gaussian case: The Particle Filter (PF)

Where is the robot?

100 samples indicating robot position



x=1 with $\sigma^2 = 0.2$



?

The Particle Filter (PF)

The *particle filter* is an approximate Bayes Filter implementation for systems with nonlinear process and measurement models and general beliefs and uncertainty.

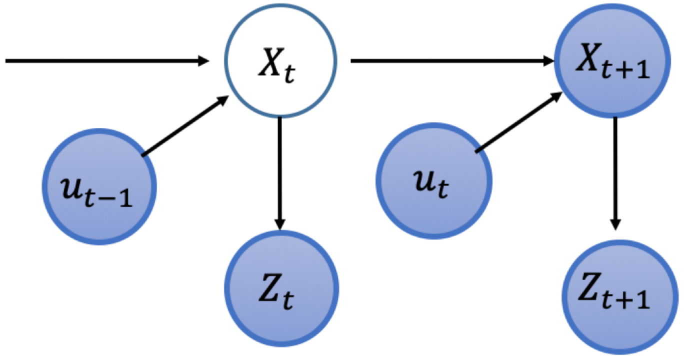
Given: State $x_t \sim p(x_t)$, control u_t

- **Process model:**

$$x_{t+1} \sim p(x_{t+1}|x_t, u_t)$$

- **Measurement model:**

$$z_t \sim p(z_t|x_t)$$



NB: Unlike the KF & EKF, the particle filter *does not* assume that the belief takes a specific form

Key question: How can we represent an *arbitrary* belief $p(x)$ in practice??

Sampling-based simulation and inference

Main idea: We can (*implicitly*) represent a probability distribution $p(x)$ using a *set of samples* $\{x\}_{i=1}^{n}$ drawn from it

In practice, we are often interested in some statistic:

$$T = E_{x \sim p}[f(x)]$$

of the distribution $p(x)$ (e.g. mean, covariance, etc.)

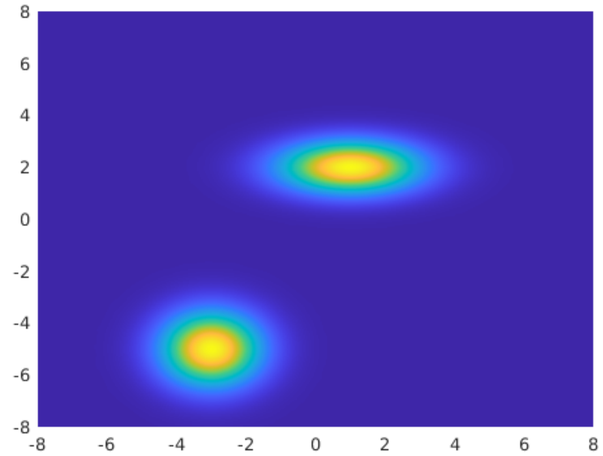
If we can *sample* from $p(x)$, then we can always approximate T using a *sample* or (*empirical*) *estimate*:

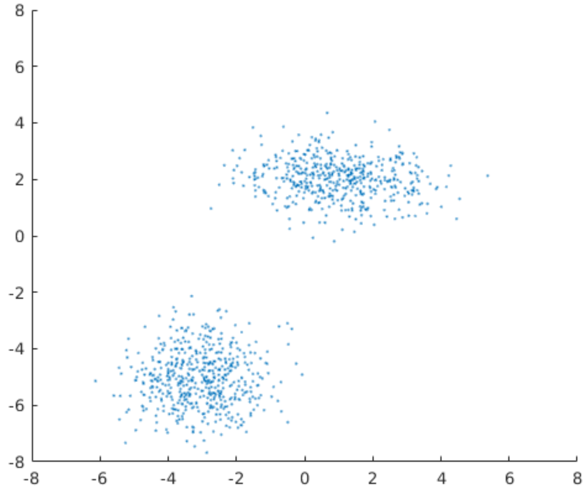
$$S_n \triangleq \frac{1}{n} \sum_{i=1}^{n} f(x_i)$$

Note that:

$$T = \lim_{n \to \infty} S_n$$

by the Law of Large Numbers



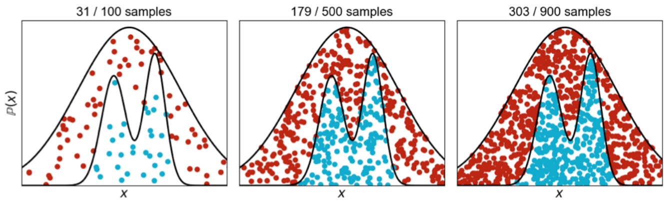


Sampling

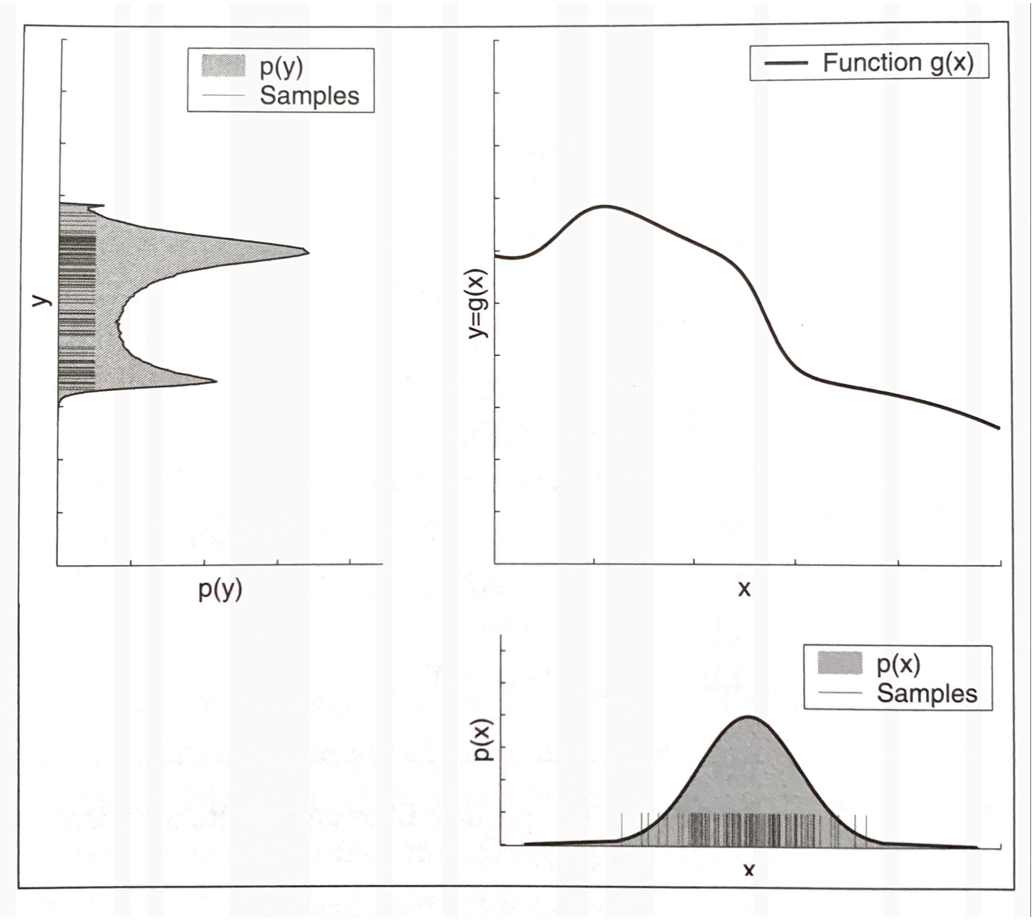
Sampling is harder for arbitrary distributions

How to obtain these samples?

- Rejection sampling (slow)



- Importance sampling



Closed form sampling is only possible for a few distributions (e.g., uniform, Gaussian).

Importance sampling

Problem: What if we *don't know how* to sample from our target distribution $p(x)$?

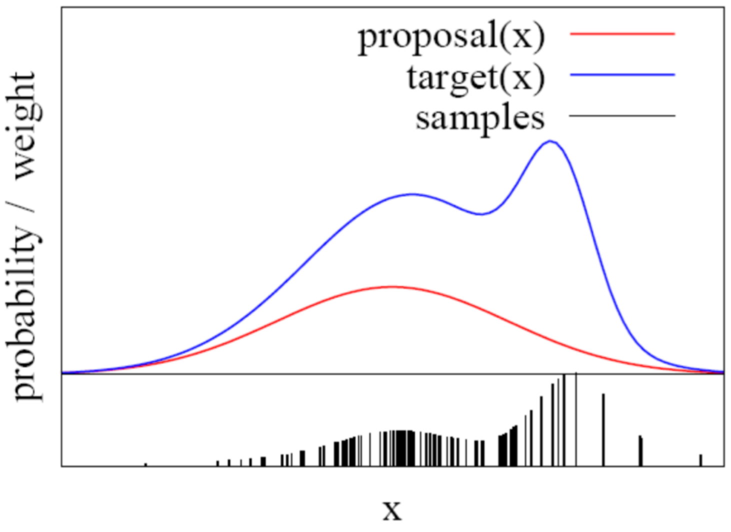
Importance sampling provides a means of simulating draws from $p(x)$ using draws from a tractable *proposal distribution* $q(x)$

Pre-condition: $p(x) > 0 \ \rightarrow q(x) > 0$

Algorithm:

1. Draw *n* samples $x_i \sim q$
2. Assign each sample x_i the *weight* $w_i \triangleq p(x_i)/q(x_i)$
3. Calculate weighted sample statistic:

$$S_n \triangleq \frac{1}{n}\sum_{i=1}^{n} w_i f(x_i)$$



One-slide derivation of importance sampling

Consider:

$$E_{x\sim q(x)}[w(x)f(x)] = E_{x\sim q(x)}\left[\frac{p(x)}{q(x)}f(x)\right]$$

$$= \int \left(\frac{p(x)}{\cancel{q(x)}}f(x)\right)\cancel{q(x)}dx$$

$$= \int p(x)f(x)dx$$

$$= E_{x\sim p(x)}[f(x)]$$

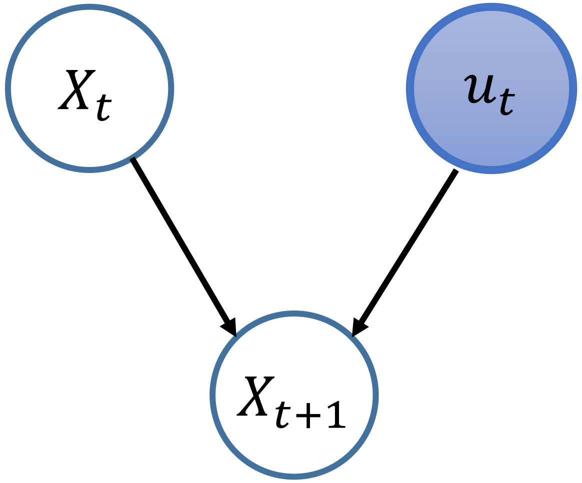
Particle Filter: Propagation step

Given: Sample set $\left\{x_t^{[i]}\right\}_{i=1}^{n}$ for *current* state x_t, control u_t, process model $p(x_{t+1}|x_t, u_t)$

Find: Sample set $\left\{x_{t+1}^{[i]}\right\}_{i=1}^{n}$ for *next* state x_{t+1}

For *i* = 1, … *n*, sample $x_{t+1}^{[i]}$ according to:

$$x_{t+1}^{[i]} \sim p(x_{t+1}|x_t^{[i]}, u_t)$$



Particle Filter: Measurement update step

Given: Sample set $\left\{\hat{x}_t^{[i]}\right\}_{i=1}^{n}$ for *prior* belief over x_t, measurement z_t, measurement model $p(z_t|\hat{x}_t)$

Find: Sample set $\left\{x_t^{[i]}\right\}_{i=1}^{n}$ for *posterior* belief over x_t given z_t

Bayes' Rule:

$$p(x_t|z_t) = \frac{p(z_t|x_t)p(x_t)}{p(z_t)}$$

Importance sampling: Calculate importance weights:

$$w_i = \frac{p(x_t|z_t)}{p(x_t)} = \frac{p(z_t|x_t)}{p(z_t)} \propto p(z_t|x_t)$$

For *i* = 1, …, n:

Draw a *posterior* sample $x_t^{[i]}$ from the set of *prior* samples $\left\{\hat{x}_t^{[i]}\right\}_{i=1}^{n}$ by selecting $\hat{x}_t^{[i]}$ with probability proportional to $w_i = p\left(z_t \middle| \hat{x}_t^{[i]}\right)$

Particle Filter: Complete algorithm

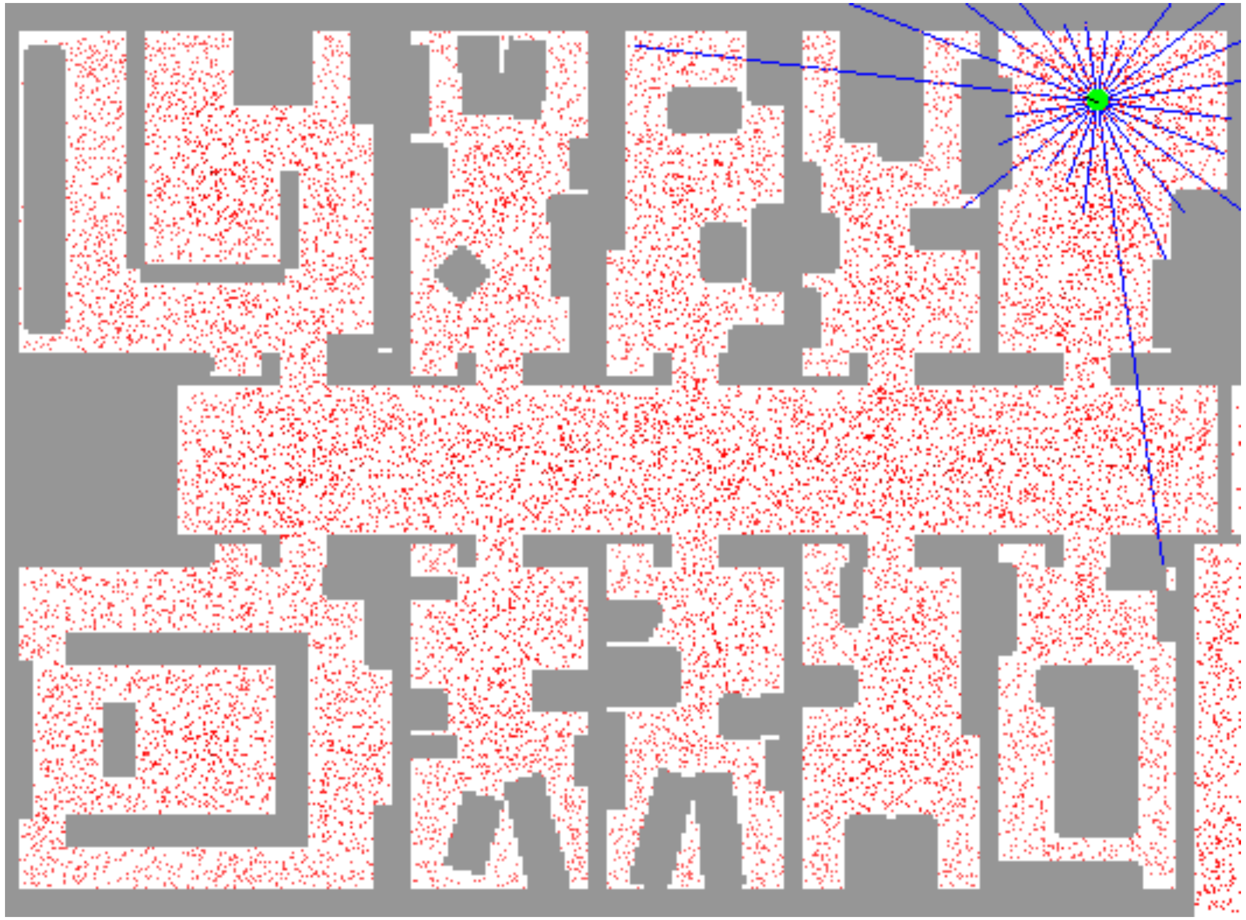
Predict: Given sample set $\left\{x_t^{[i]}\right\}_{i=1}^{n}$ for belief over *current* state x_t, control u_t, and process model $p(x_{t+1}|x_t, u_t)$, draw sample set $\left\{x_{t+1}^{[i]}\right\}_{i=1}^{n}$ for belief over *next* state x_{t+1} according to:

$$x_{t+1}^{[i]} \sim p(x_{t+1}|x_t^{[i]}, u_t)$$

Update: Given sample set $\left\{\hat{x}_t^{[i]}\right\}_{i=1}^{n}$ for *prior* belief over x_t, measurement z_t, and measurement model $p(z_t|x_t)$, draw sample set $\left\{x_t^{[i]}\right\}_{i=1}^{n}$ for *posterior* of x_t given z_t by:

1. **Calculate weights:** Compute particle importance weights: $w_i \triangleq p(z_t|\, \hat{x}_t^{[i]})$
2. **Resample:** For each *i* = 1, …, n, sample $x_t^{[i]}$ from $\left\{\hat{x}_t^{[i]}\right\}_{i=1}^{n}$ by drawing $\hat{x}_t^{[i]}$ with probability proportional to its weight w_i

Example: Particle filter localization



Particle filter practicalities

The PF is a *nonparametric* approximation of the Bayes Filter for general (nonlinear + non-Gaussian) state estimation problems

Super simple to implement – requires only:

- A sampler for the motion model. Typically this is very simple (simulation)
- The likelihood function for your sensors – usually this comes straight from the sensor geometry

BUT: Particle filters can be finicky in practice!

- How many samples do we need? Typically *exponential* in the dimension of the state!
  ⇒Can require lots of storage, compute
- **Particle depletion:** If particle set is *insufficiently diverse*, it may not be able to track all relevant modes of the true posterior!

⇒ Particle filtering works best for low-dimensional state estimation problems

Plan of the day

Last time: Probabilistic robotics and the Bayes Filter

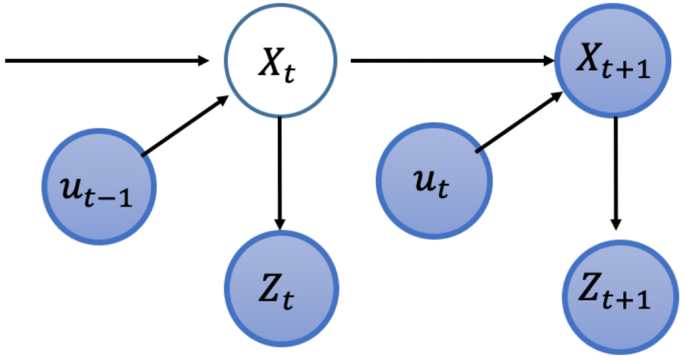
Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})\ dX_{t-1}$$

- **Update** belief after incorporating measurement Z_t at current state X_t:

$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})\ dX_t}$$



	Kalman Filter	Extended Kalman Filter	Particle Filter
Dynamics model	Linear	Nonlinear	Nonlinear
Sensor model	Linear	Nonlinear	Nonlinear
Noise	Gaussian (Unimodal)	Gaussian (Unimodal)	Multimodal