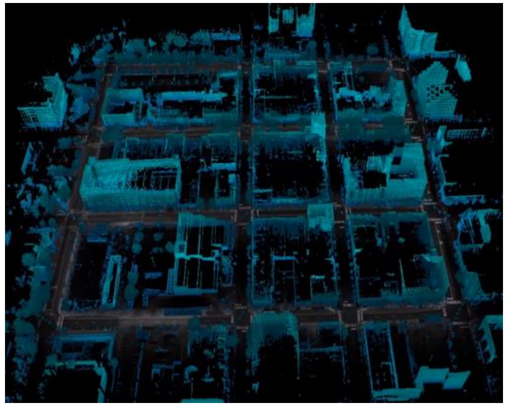
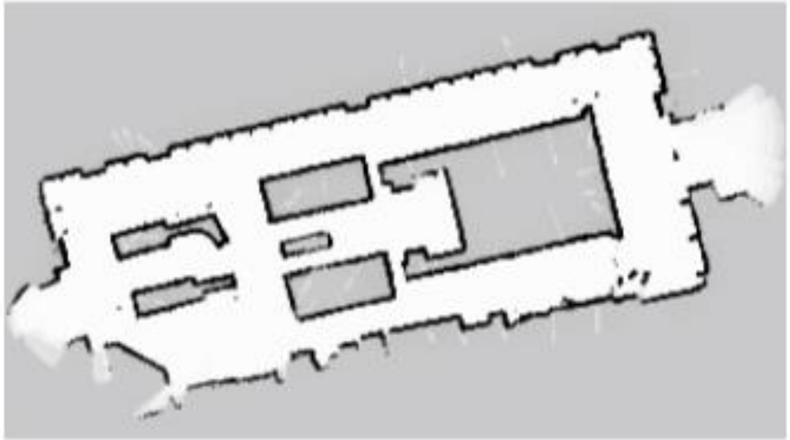
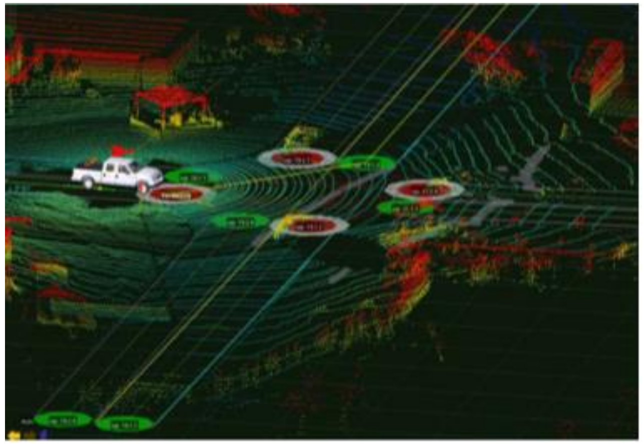
BYU Electrical & Computer Engineering





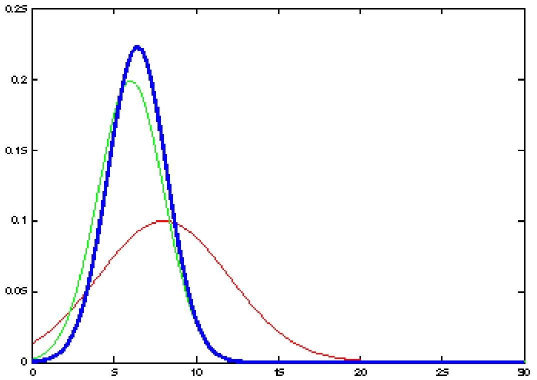




EXTENDED KALMAN FILTER

ECEN 633: Robotic Localization and Mapping

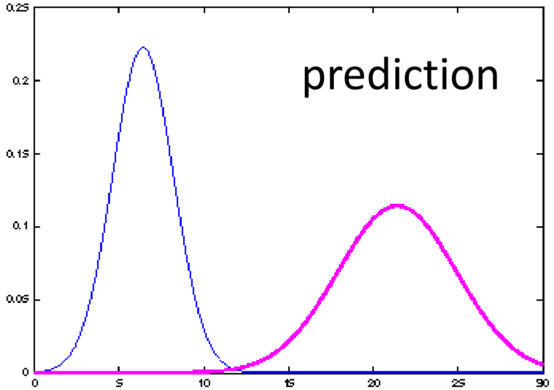
Some slides courtesy of Ryan Eustice.

Kalman Prediction-Correction Cycle

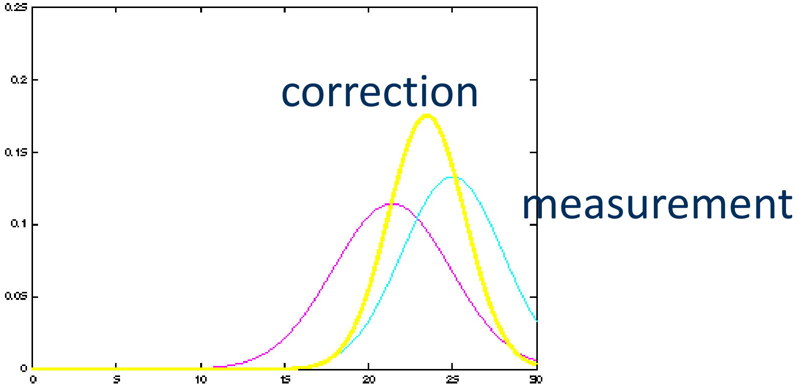


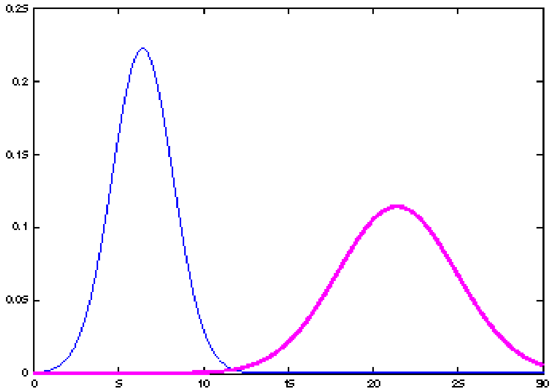
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{\varepsilon_t}^2 \end{cases}$$

$$\overline{bel}(\mathbf{x}_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t \mathbf{u}_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$



Kalman Prediction-Correction Cycle





$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + k_t(z_t - c_t\bar{\mu}_t) \\ \sigma_t^2 = (1 - k_t c_t)\bar{\sigma}_t^2 \end{cases}, k_t = \frac{c_t \bar{\sigma}_t^2}{c_t^2 \bar{\sigma}_t^2 + \sigma_{\delta_t}^2}$$

$$bel(\mathbf{x}_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(\mathbf{z}_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

Kalman Prediction-Correction Cycle

Prediction

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + k_t(z_t - c_t\bar{\mu}_t) \\ \sigma_t^2 = (1 - k_t c_t)\bar{\sigma}_t^2 \end{cases}, k_t = \frac{c_t\bar{\sigma}_t^2}{c_t^2\bar{\sigma}_t^2 + \sigma_{\delta_t}^2}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t\mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2\sigma_t^2 + \sigma_{\varepsilon_t}^2 \end{cases}$$

$$bel(\mathbf{x}_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(\mathbf{z}_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

$$\overline{bel}(\mathbf{x}_t) = \begin{cases} \bar{\mu}_t = A_t\mu_{t-1} + B_t\mathbf{u}_t \\ \bar{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + R_t \end{cases}$$

Correction

Kalman Filter Summary

- **Highly efficient:** Polynomial in measurement dimensionality k and state dimensionality n:
  $$O(k^{2.376} + n^2)$$
- **Kalman Gain**
  - Weights measurement update by:
    - State Uncertainty
    - Measurement Uncertainty
- **Optimal for linear Gaussian systems**!
  - No other estimator can do better
- Most robotics systems are **nonlinear**!

Nonlinear Dynamic Systems

- Most realistic robotic problems involve nonlinear functions

$$\boxed{\mathbf{x}_t = g(\mathbf{u}_t, \mathbf{x}_{t-1}) + \varepsilon_t}$$

$$\boxed{\mathbf{z}_t = h(\mathbf{x}_t) + \delta_t}$$

Projecting Covariances (Nonlinear Case)

- Again, suppose: $x \sim \mu_x, \Sigma_x$

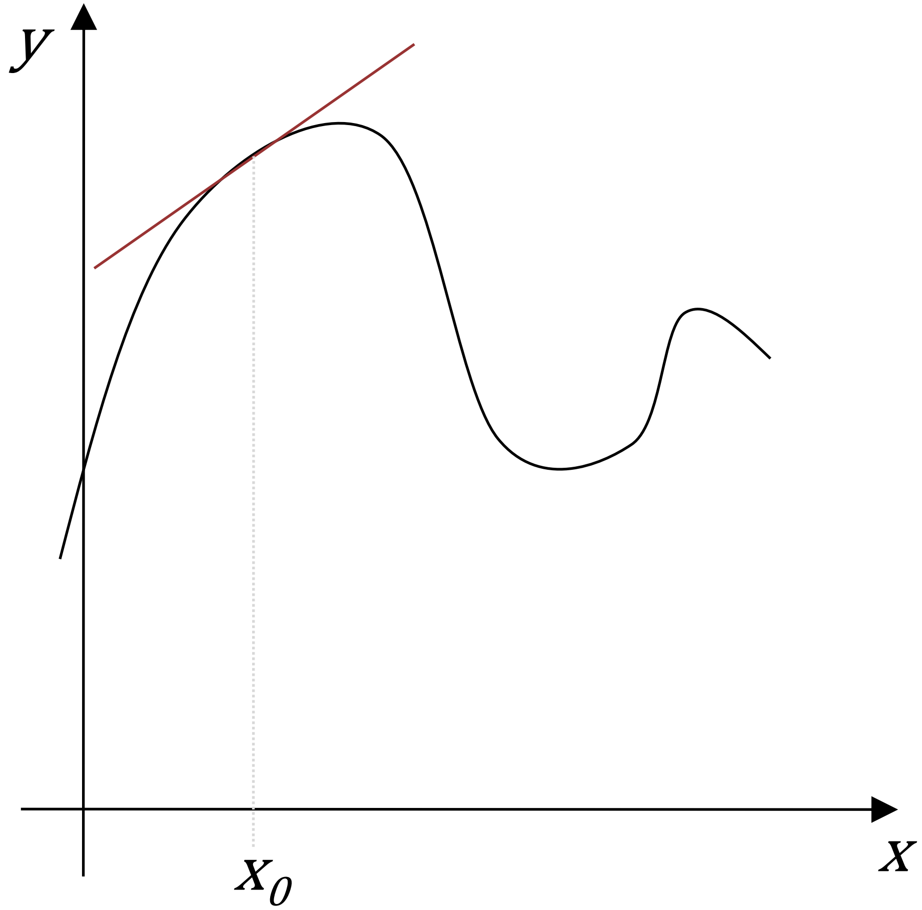
$$y = \cancel{Ax} + b \qquad\qquad y = f(x)$$

- Approach: approximate *f(x)* with Taylor expansion
  - What point should we approximate *f(x)* around?

Projecting Covariances (Nonlinear Case)

- First-order Taylor expansion
  - Let's review 1D case

$$y \approx \left.\frac{df}{dx}\right|_{x_0} (x - x_0) + f(x_0)$$



Projecting Covariances (Nonlinear Case)

- Generalized case:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, \dots) \\ f_2(x_1, x_2, \dots) \\ \dots \end{bmatrix}$$

$$\mathbf{y} \approx \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots \\ \dots & \dots & \dots \end{bmatrix}}_{\text{“Jacobian”}} \begin{bmatrix} x_1 - x_{1_0} \\ x_2 - x_{2_0} \\ \dots \end{bmatrix} + \begin{bmatrix} f_1(x_{1_0}, x_{2_0}) \\ f_2(x_{1_0}, x_{2_0}) \\ \dots \end{bmatrix}$$

$$\mathbf{y} \approx J|_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0) + \mathbf{f}(\mathbf{x}_0)$$

Projecting Covariances (Nonlinear Case)

$$\mathbf{y} = \mathbf{f}(\mathbf{x})$$

$$\mathbf{y} \approx J|_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0) + \mathbf{f}(\mathbf{x}_0)$$

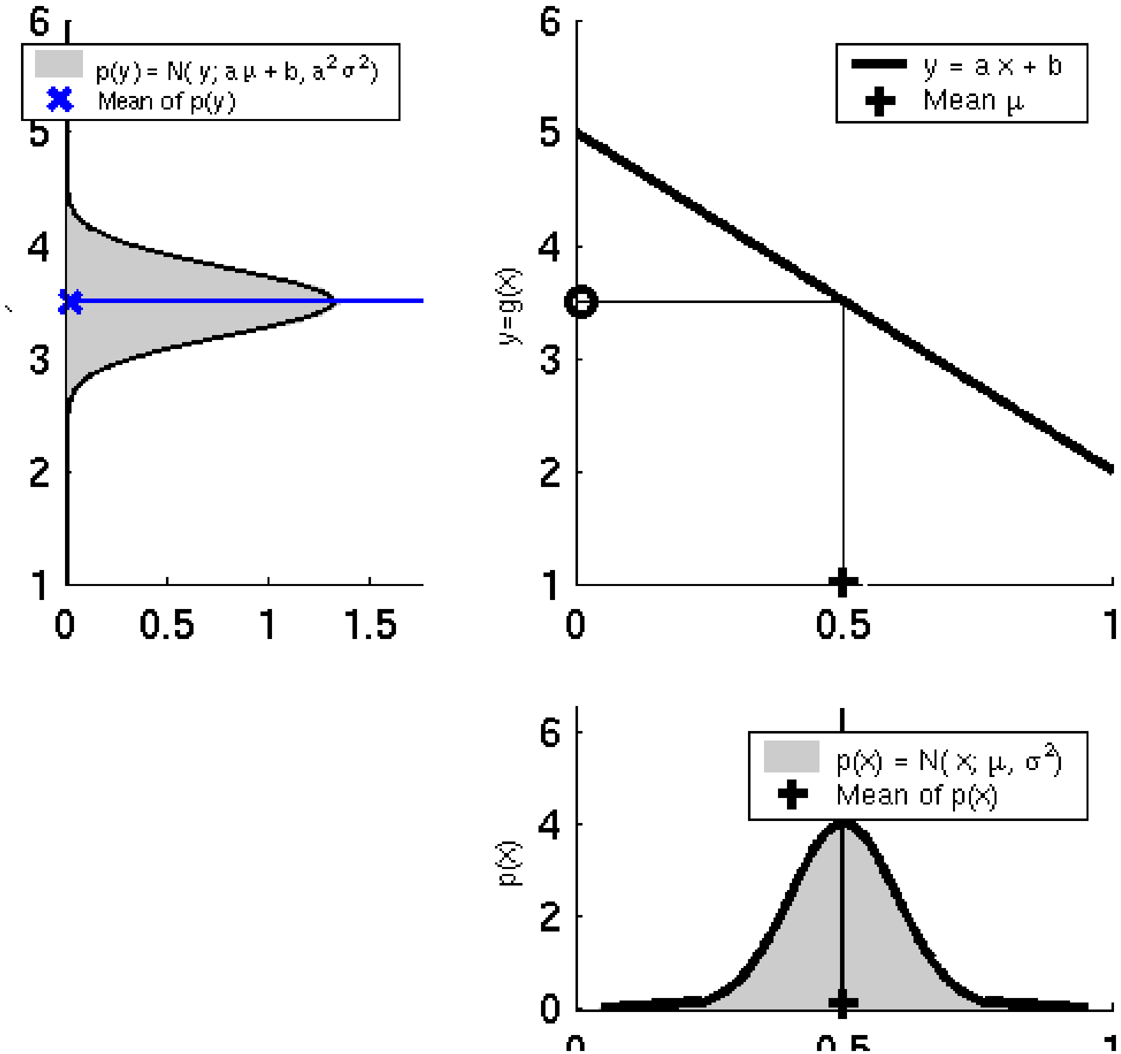
$$\mathbf{y} \approx \underbrace{J|_{\mathbf{x}_0}}_{\mathrm{A}}\mathbf{x} \underbrace{- J|_{\mathbf{x}_0}\mathbf{x}_0 + \mathbf{f}(\mathbf{x}_0)}_{\mathrm{b}}$$

$$\boxed{\begin{aligned} y &= Ax + b \\ \Sigma_y &= A\Sigma_x A^T \end{aligned}}$$

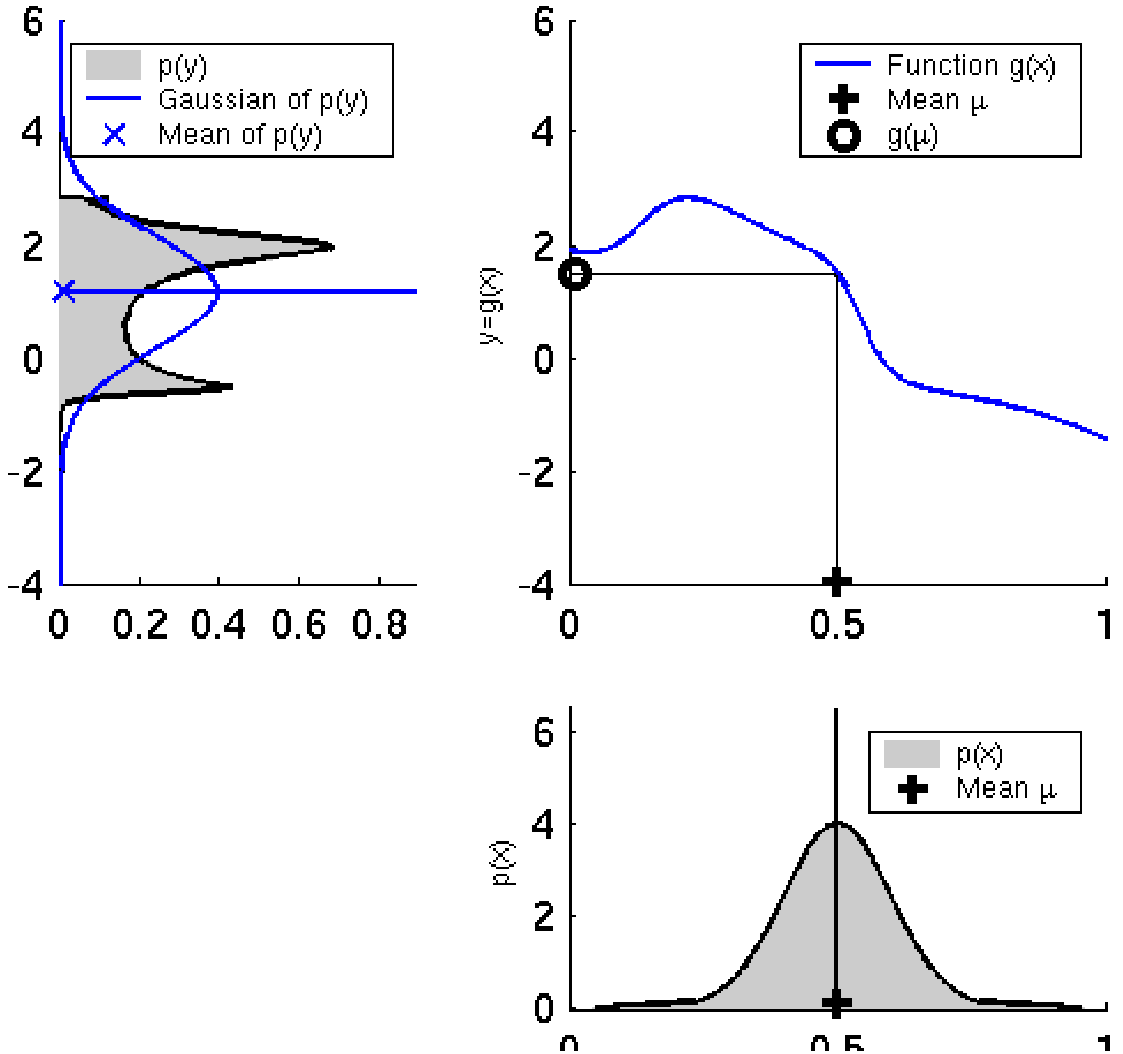
Non-linear case is reduced to linear case via first-order Taylor approximation. Expansion point $\mathbf{x}_0$ is typically taken as the mean.

What do we lose by dropping higher order terms?

Linearity Assumption Revisited



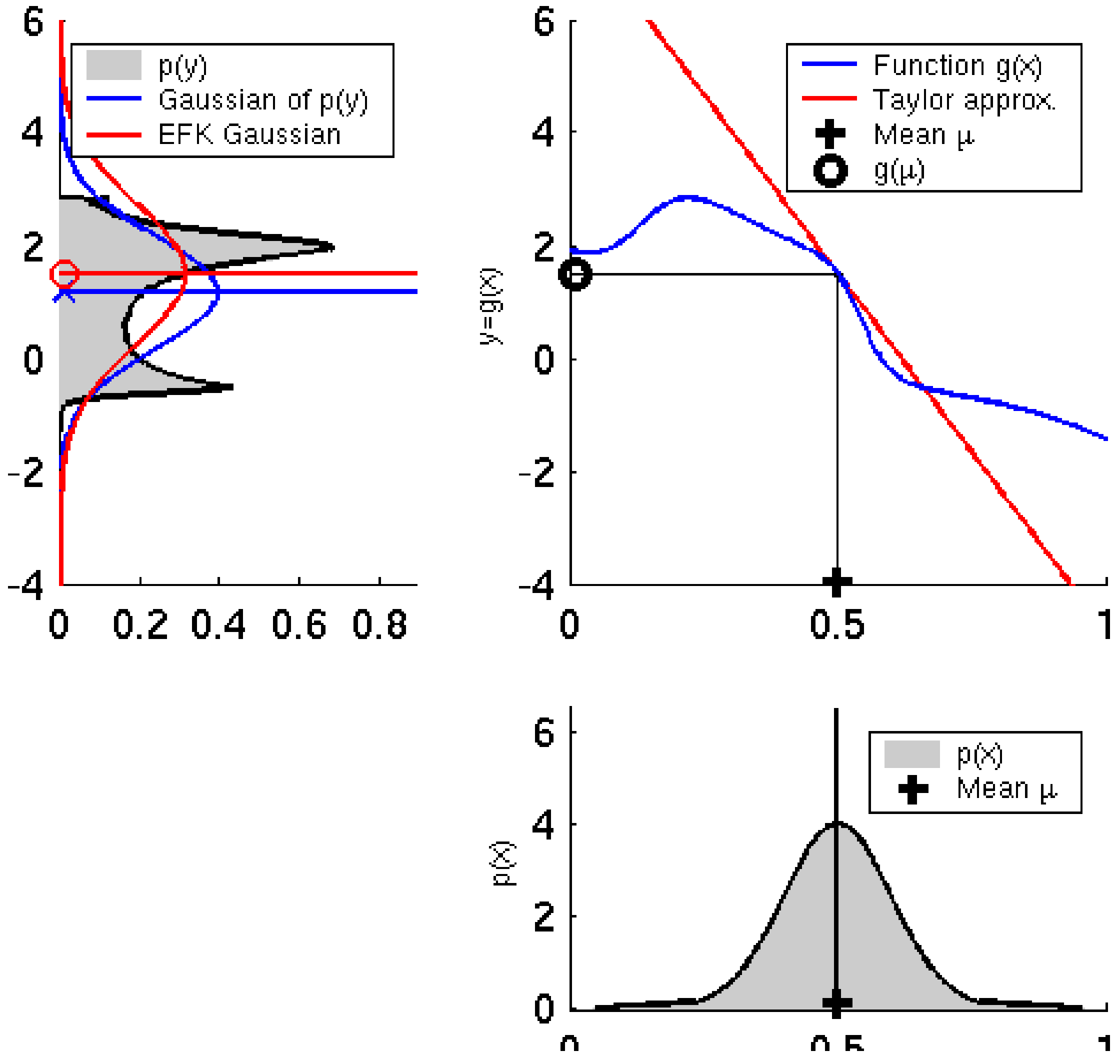
Nonlinear Function



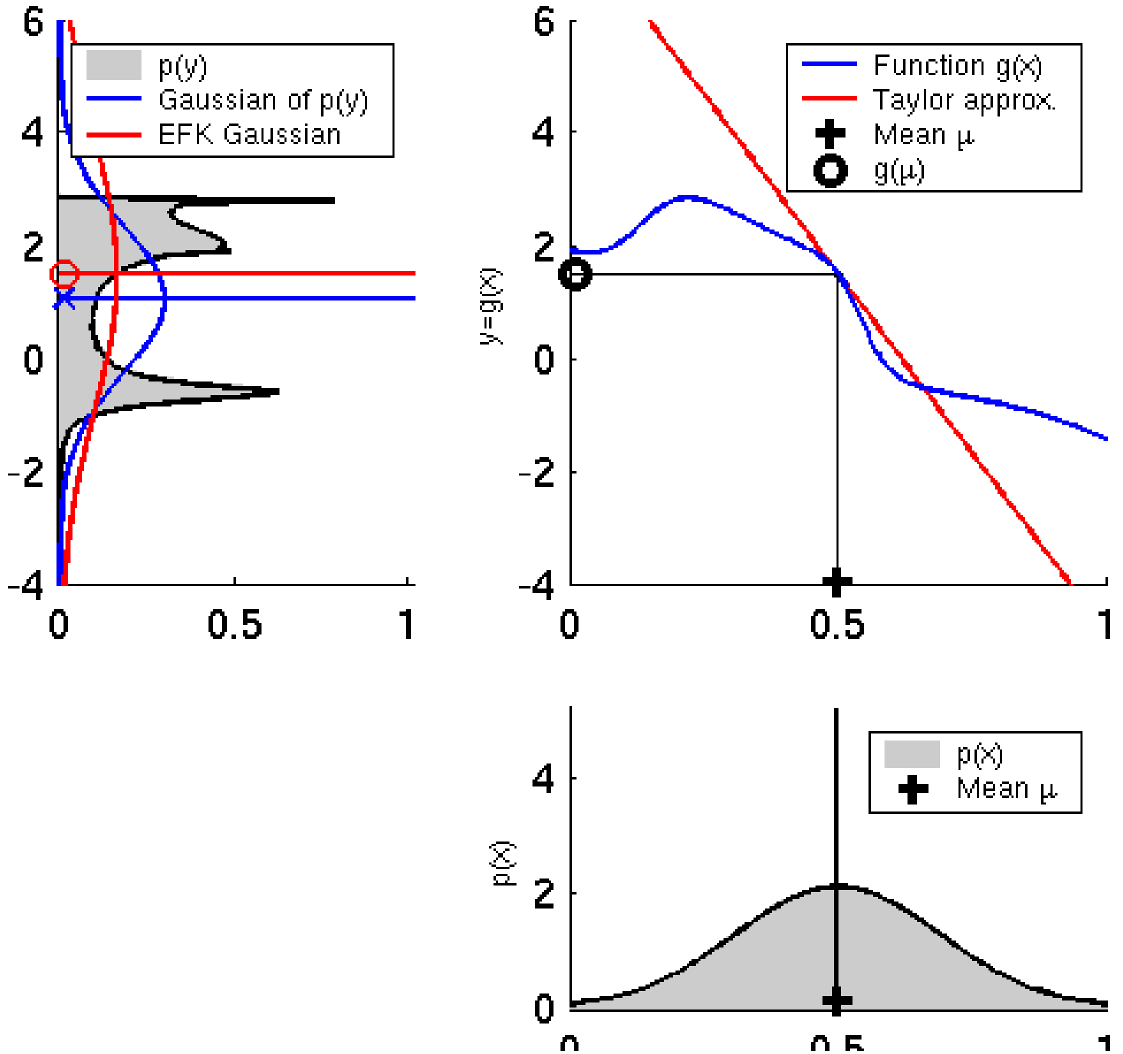
Nonlinear Gaussian Filters

- Approach 1: Extended Kalman Filter
  - **Approximate the model!**
  - Linearize our nonlinear plant and/or observation model(s) about the current mean and use the linear KF equations.

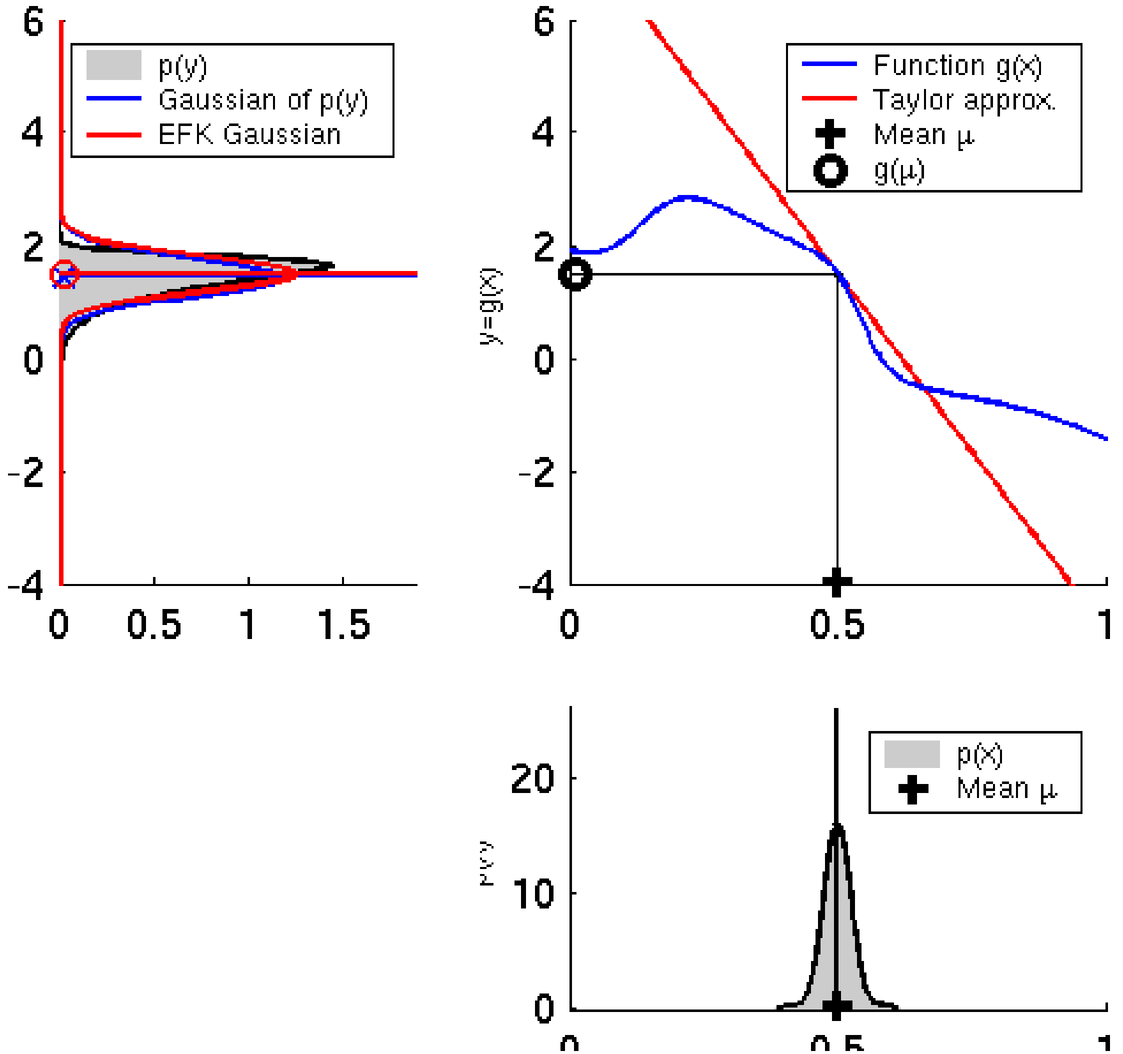
EKF Linearization via First Order Taylor Series



EKF Linearization: Large Variance



EKF Linearization: Narrow Variance



EKF Linearization: First Order Taylor Series Expansion

- Prediction:

$$g(\mathbf{u}_t, \mathbf{x}_{t-1}) \approx g(\mathbf{u}_t, \mu_{t-1}) + \frac{\partial g(\mathbf{u}_t, \mu_{t-1})}{\partial \mathbf{x}_{t-1}}(\mathbf{x}_{t-1} - \mu_{t-1})$$

$$g(\mathbf{u}_t, \mathbf{x}_{t-1}) \approx g(\mathbf{u}_t, \mu_{t-1}) + G_t(\mathbf{x}_{t-1} - \mu_{t-1})$$

- Correction:

$$h(\mathbf{x}_t) \approx h(\overline{\mu}_t) + \frac{\partial h(\overline{\mu}_t)}{\partial \mathbf{x}_t}(\mathbf{x}_t - \overline{\mu}_t)$$

$$h(\mathbf{x}_t) \approx h(\overline{\mu}_t) + H_t(\mathbf{x}_t - \overline{\mu}_t)$$

EKF Algorithm*

1. **Extended_Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, \boldsymbol{u}_t, \boldsymbol{z}_t$):

2. Prediction:
3. $\bar{\mu}_t = g(\mathbf{u}_t, \mu_{t-1})$
4. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

5. Correction:
6. $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t(\mathbf{z}_t - h(\bar{\mu}_t))$
8. $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$
9. Return μ_t, Σ_t

Linear KF

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t \mathbf{u}_t$$

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

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

$$\mu_t = \bar{\mu}_t + K_t(\mathbf{z}_t - C_t \bar{\mu}_t)$$

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

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial \mathbf{x}_t} \qquad G_t = \frac{\partial g(\mathbf{u}_t, \mu_{t-1})}{\partial \mathbf{x}_{t-1}}$$

* The form shown assumes additive process and observation model noise

EKF Summary

- **Highly efficient**: Polynomial in measurement dimensionality k and state dimensionality n:

  $$O(k^{2.376} + n^2)$$

- **Not optimal**!
- Can diverge if nonlinearities are large!
- Can work surprisingly well even when all assumptions are violated!

KF, EKF and UKF

- Kalman filter requires linear models
- EKF linearizes via Taylor expansion

Is there a better way to linearize?

- Sometimes yes, w/ tradeoffs
  - Unscented Transform (one popular option) -> Unscented Kalman Filter