

Introduction to Mobile Robotics

Bayes Filter – Kalman Filter

Wolfram Burgard, Diego Tipaldi



Bayes Filter Reminder

$$bel(x_t) = \eta p(z_t | x_t) \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$$

- Prediction

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$$

- Correction

$$bel(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t)$$

Kalman Filter

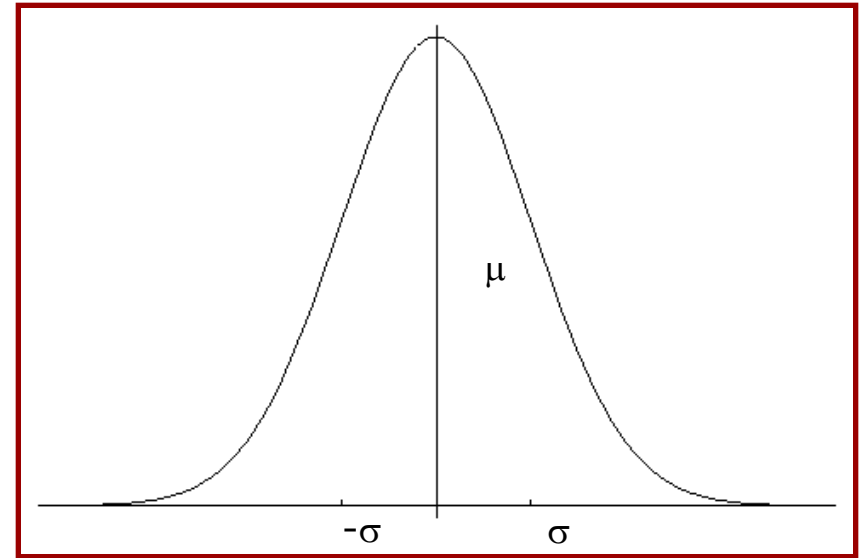
- Bayes filter with **Gaussians**
- Developed in the late 1950's
- Most relevant Bayes filter variant in practice
- Applications range from economics, weather forecasting, satellite navigation to robotics and many more.
- The Kalman filter “algorithm” is a couple of **matrix multiplications!**

Gaussians

$p(x) \sim N(\mu, \sigma^2):$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}}$$

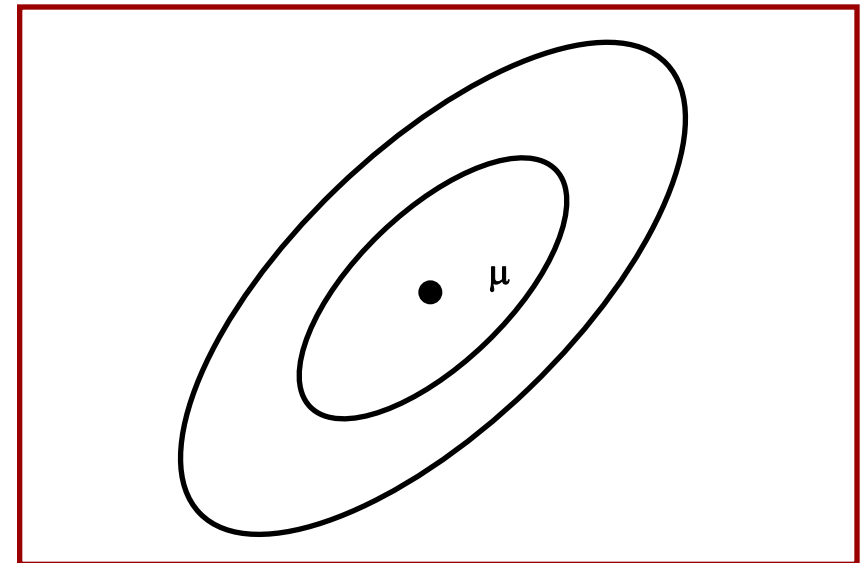
Univariate



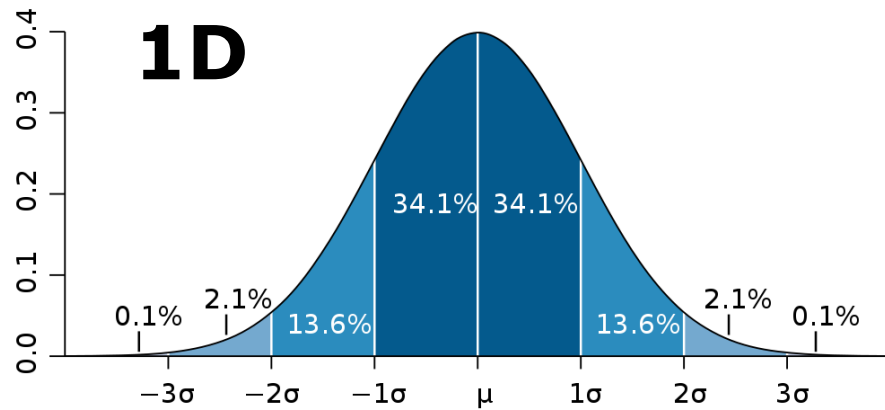
$p(\mathbf{x}) \sim N(\mu, \Sigma):$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2} (\mathbf{x}-\mu)' \Sigma^{-1} (\mathbf{x}-\mu)}$$

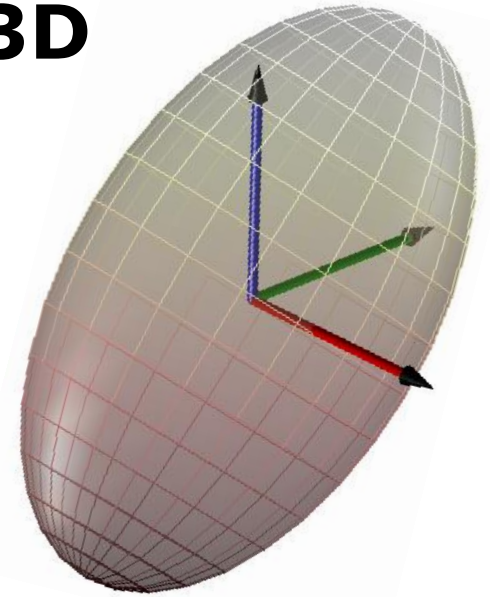
Multivariate



Gaussians



3D



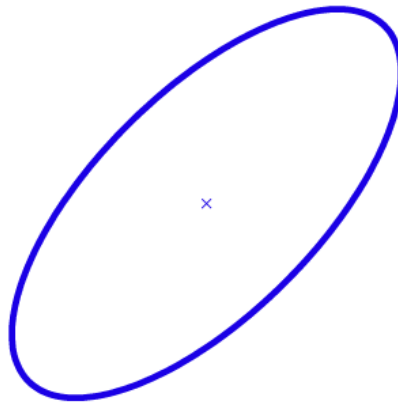
2D

$$C = \begin{bmatrix} 0.020 & 0.013 \\ 0.013 & 0.020 \end{bmatrix}$$

$$\lambda_1 = 0.007$$

$$\lambda_2 = 0.033$$

$$\rho = \sigma_{XY} / \sigma_X \sigma_Y = 0.673$$



Properties of Gaussians

- Univariate case

$$\left. \begin{array}{l} X \sim N(\mu, \sigma^2) \\ Y = aX + b \end{array} \right\} \Rightarrow Y \sim N(a\mu + b, a^2\sigma^2)$$

$$\left. \begin{array}{l} X_1 \sim N(\mu_1, \sigma_1^2) \\ X_2 \sim N(\mu_2, \sigma_2^2) \end{array} \right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \mu_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \mu_2, \frac{1}{\sigma_1^{-2} + \sigma_2^{-2}}\right)$$

Properties of Gaussians

- Multivariate case

$$\left. \begin{array}{l} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{array} \right\} \Rightarrow Y \sim N(A\mu + B, A\Sigma A^T)$$

$$\left. \begin{array}{l} X_1 \sim N(\mu_1, \Sigma_1) \\ X_2 \sim N(\mu_2, \Sigma_2) \end{array} \right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2} \mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2} \mu_2, \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}}\right)$$

(where division "-" denotes matrix inversion)

- We **stay Gaussian** as long as we start with Gaussians and perform only **linear transformations**

Discrete Kalman Filter

Estimates the state x of a discrete-time controlled process that is governed by the linear stochastic difference equation

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

with a measurement

$$z_t = C_t x_t + \delta_t$$

Components of a Kalman Filter

 A_t

Matrix ($n \times n$) that describes how the state evolves from $t-1$ to t without controls or noise.

 B_t

Matrix ($n \times l$) that describes how the control u_t changes the state from $t-1$ to t .

 C_t

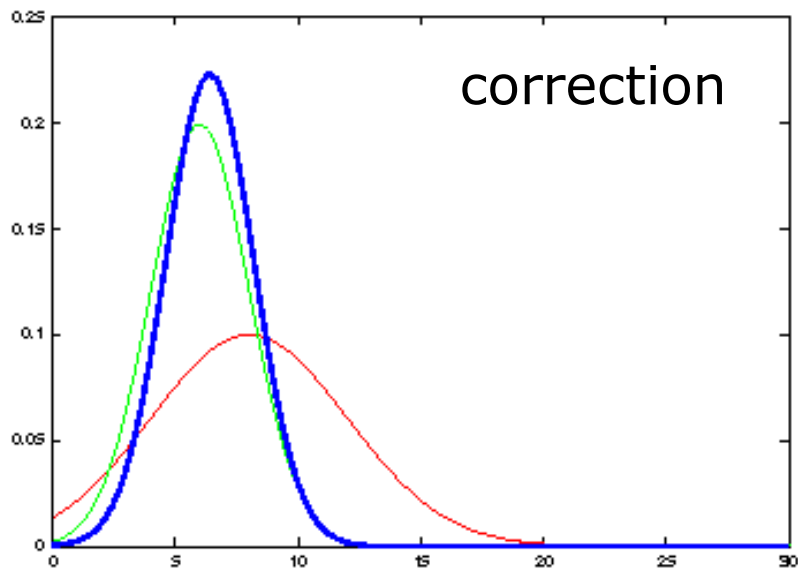
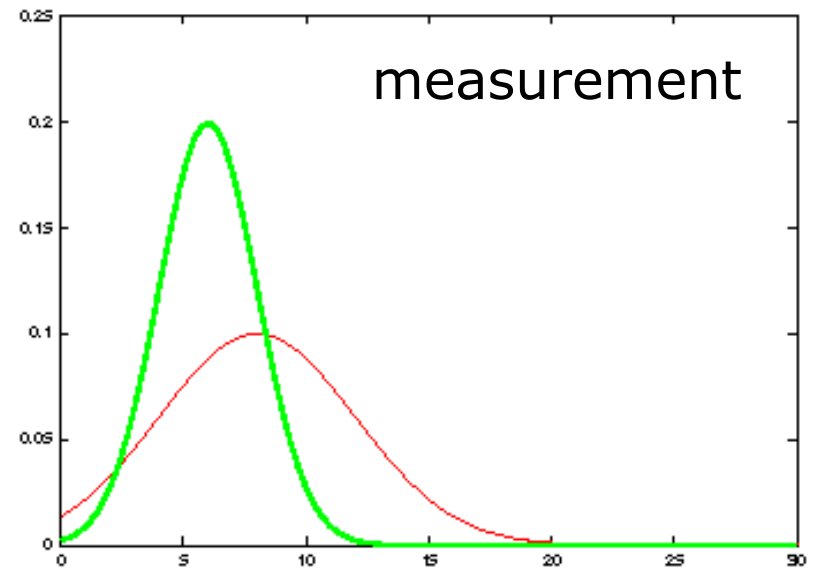
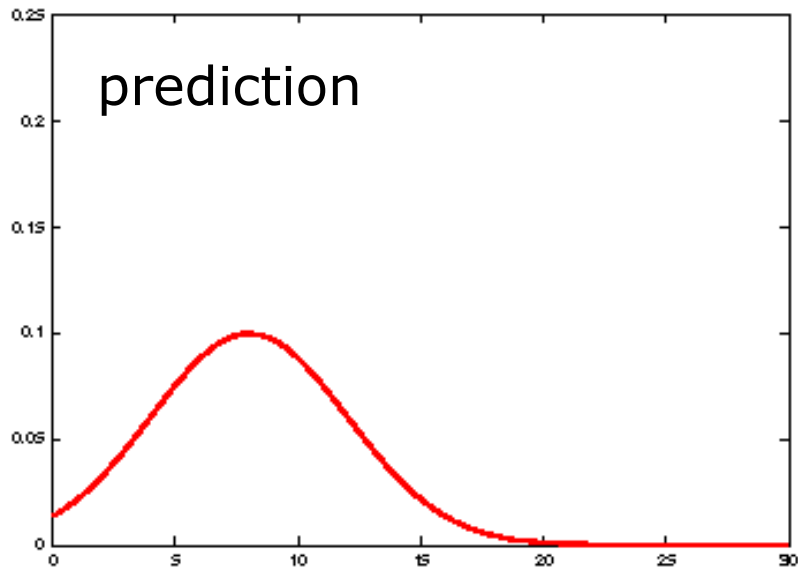
Matrix ($k \times n$) that describes how to map the state x_t to an observation z_t .

 ε_t

Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance Q_t and R_t respectively.

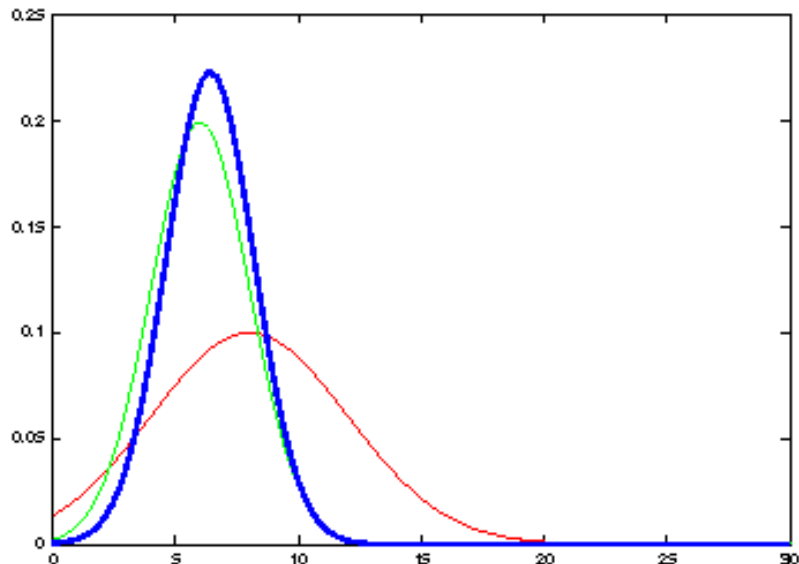
 δ_t

Kalman Filter Updates in 1D



It's a weighted mean!

Kalman Filter Updates in 1D

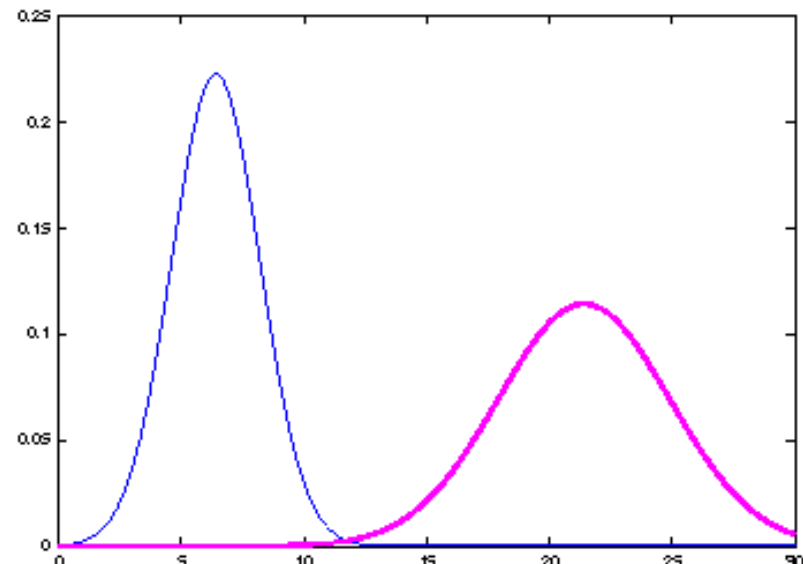
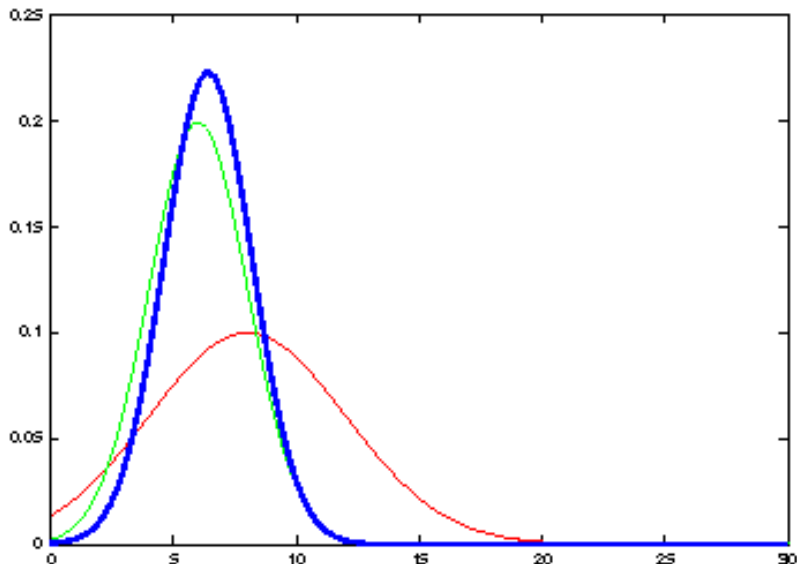


How to get the blue one?
Kalman correction step

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases} \text{ with } K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_tC_t)\bar{\Sigma}_t \end{cases} \text{ with } K_t = \bar{\Sigma}_tC_t^T(C_t\bar{\Sigma}_tC_t^T + R_t)^{-1}$$

Kalman Filter Updates in 1D



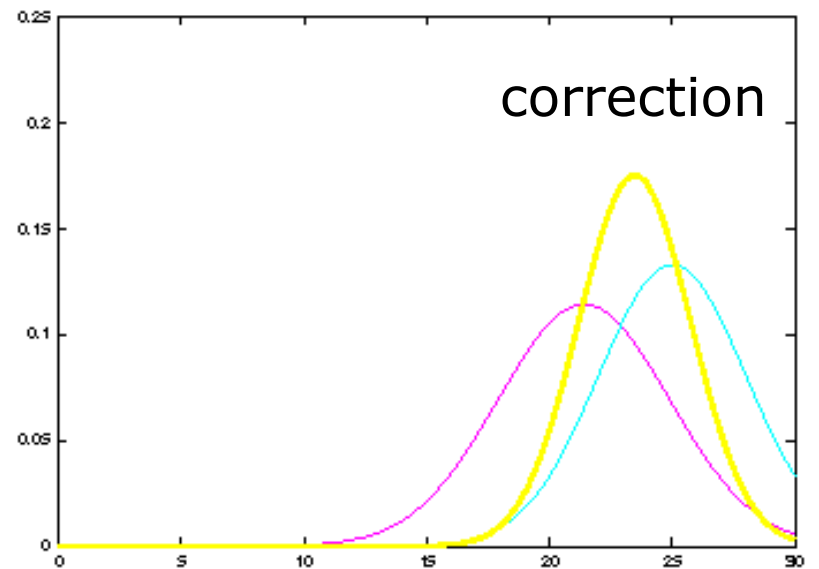
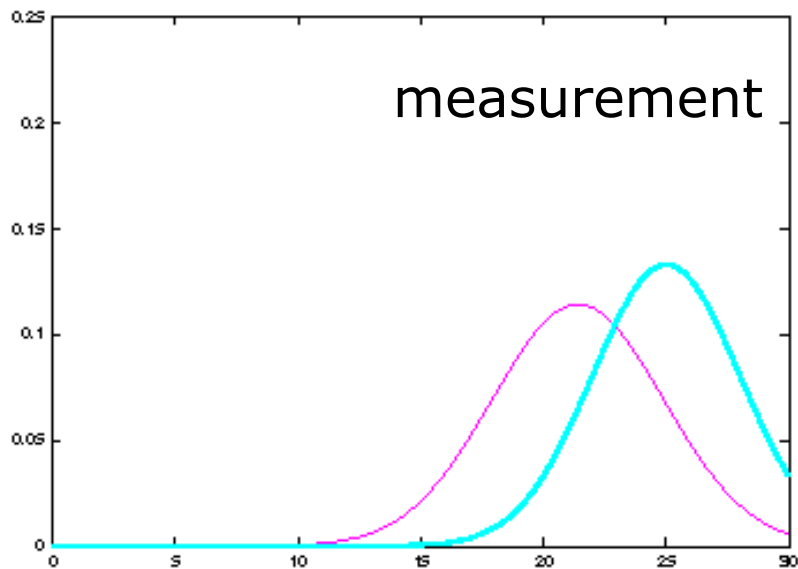
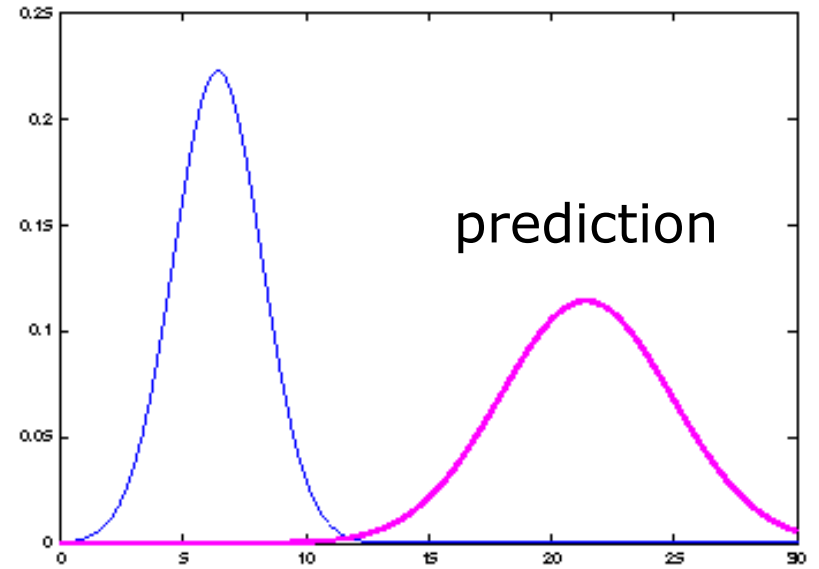
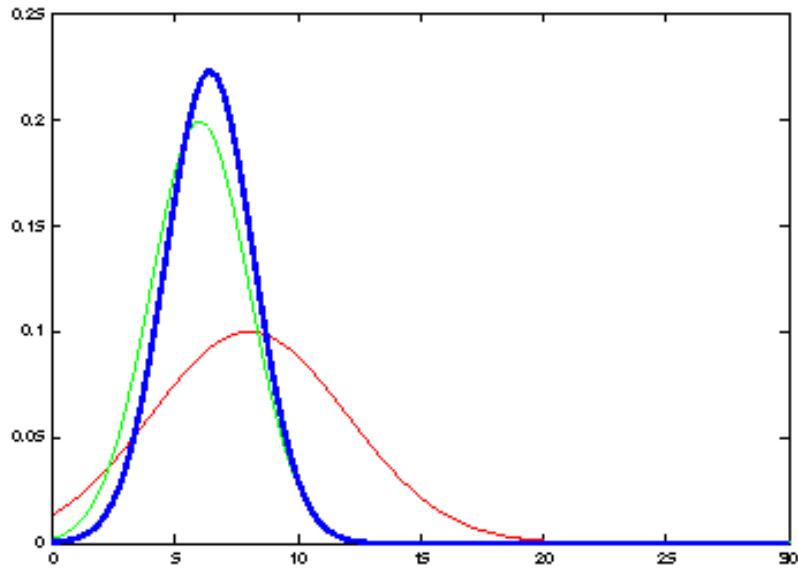
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_{t-1}^2 + \sigma_{act,t}^2 \end{cases}$$

How to get the magenta one?

State prediction step

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t \end{cases}$$

Kalman Filter Updates



Linear Gaussian Systems: Initialization

Initial belief is normally distributed:

$$bel(x_0) = N(x_0; \mu_0, \Sigma_0)$$

Linear Gaussian Systems: Dynamics

Dynamics are linear functions of the state and the control plus additive noise:

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

$$p(x_t | u_t, x_{t-1}) = N(x_t; A_t x_{t-1} + B_t u_t, Q_t)$$

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$$

⇓ ⇓

$$\sim N(x_t; A_t x_{t-1} + B_t u_t, Q_t) \sim N(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})$$

Linear Gaussian Systems: Dynamics

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$$

⇓ ⇓

$$\sim N(x_t; A_t x_{t-1} + B_t u_t, Q_t) \sim N(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})$$

⇓

$$\overline{bel}(x_t) = \eta \int \exp \left\{ -\frac{1}{2} (x_t - A_t x_{t-1} - B_t u_t)^T Q_t^{-1} (x_t - A_t x_{t-1} - B_t u_t) \right\}$$

$$\exp \left\{ -\frac{1}{2} (x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1}) \right\} dx_{t-1}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t \end{cases}$$

Linear Gaussian Systems: Observations

Observations is a linear function of the state plus additive noise:

$$z_t = C_t x_t + \delta_t$$

$$p(z_t | x_t) = N(z_t; C_t x_t, R_t)$$

$$bel(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t)$$

$\Downarrow \Downarrow$

$$\sim N(z_t; C_t x_t, R_t) \sim N(x_t; \overline{\mu}_t, \overline{\Sigma}_t)$$

Linear Gaussian Systems: Observations

$$bel(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t)$$

⇓ ⇓

$$\sim N(z_t; C_t x_t, R_t) \sim N(x_t; \bar{\mu}_t, \bar{\Sigma}_t)$$

⇓

$$bel(x_t) = \eta \exp \left\{ -\frac{1}{2} (z_t - C_t x_t)^T R_t^{-1} (z_t - C_t x_t) \right\} \exp \left\{ -\frac{1}{2} (x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1} (x_t - \bar{\mu}_t) \right\}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \end{cases} \text{ with } K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$$

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**(μ_{t-1} , Σ_{t-1} , u_t , z_t):

2. Prediction:

3.
$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$$

4.
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$$

5. Correction:

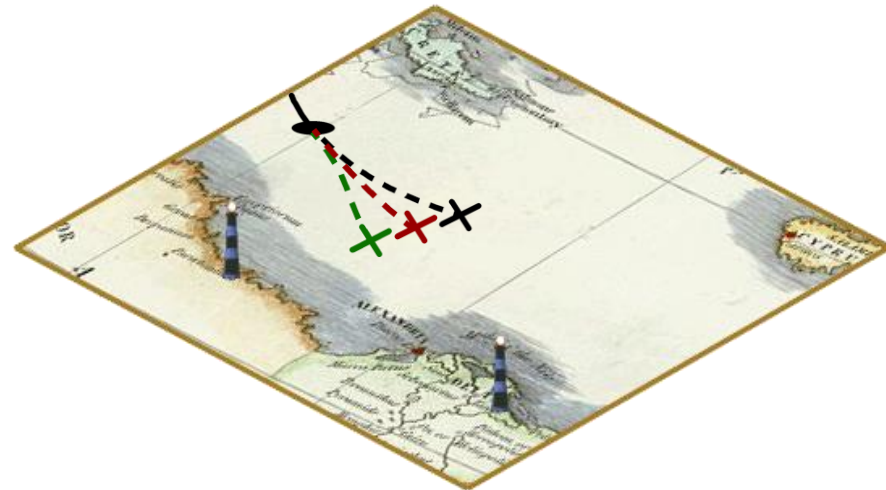
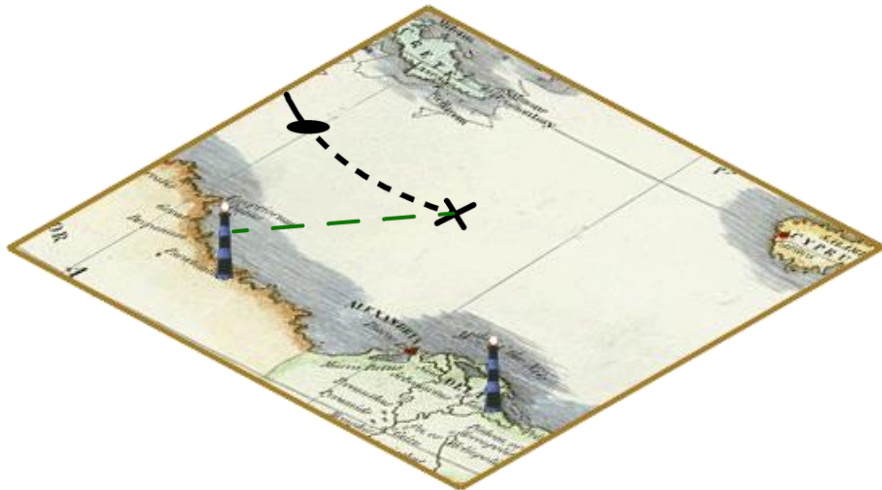
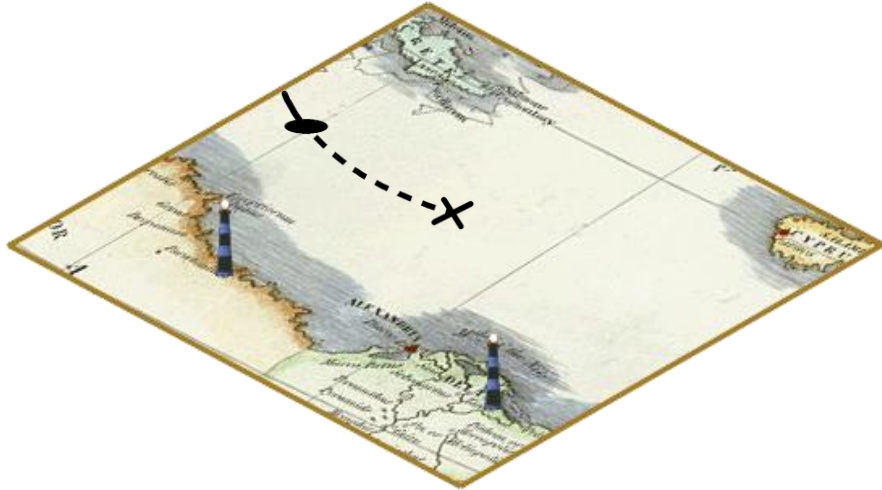
6.
$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$$

7.
$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$

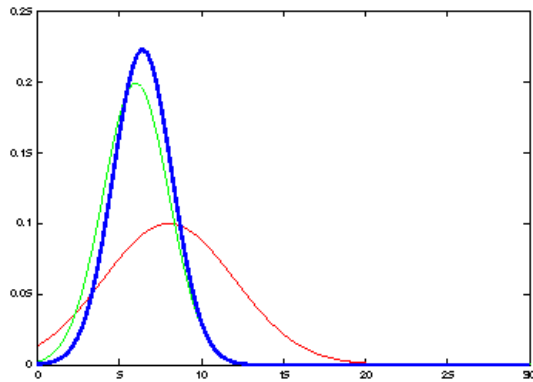
8.
$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

9. Return μ_t , Σ_t

Kalman Filter Algorithm

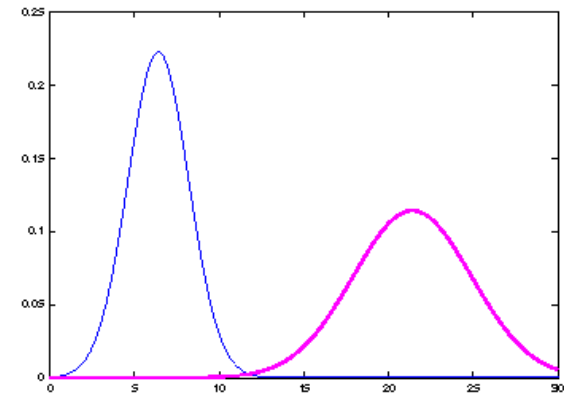


The Prediction-Correction-Cycle

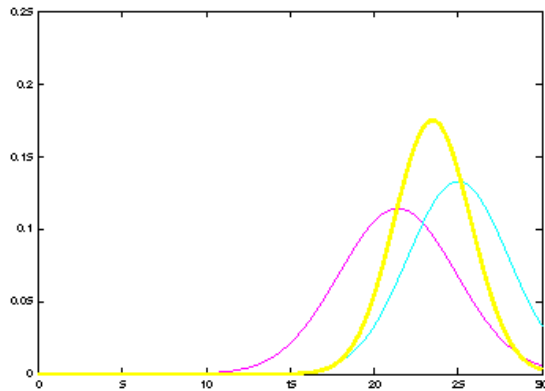


$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t \end{cases}$$

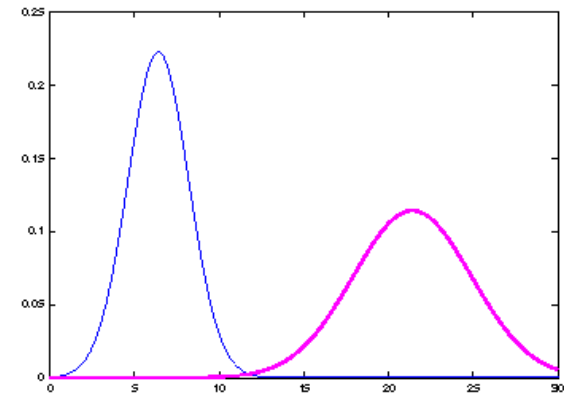


The Prediction-Correction-Cycle



$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_tC_t)\bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_tC_t^T(C_t\bar{\Sigma}_tC_t^T + R_t)^{-1}$$



Correction

The Prediction-Correction-Cycle



$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

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

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t \end{cases}$$



Kalman Filter Summary

- Only two parameters describe belief about the state of the system
- **Highly efficient:** Polynomial in the measurement dimensionality k and state dimensionality n :

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

- **Optimal for linear Gaussian systems!**
- However: Most robotics systems are **nonlinear!**
- Can only model unimodal beliefs