Introduction to Mobile Robotics

Bayes Filter – Kalman Filter

Daniel Büscher

Bayes Filter Reminder

$$Bel(x_t) = \eta \; p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1}) \, Bel(x_{t-1}) \, dx_{t-1}$$

- Prediction

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

- Correction

$$Bel(x_t) = \eta \, p(z_t \mid x_t) \, \overline{Bel}(x_t)$$

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**!

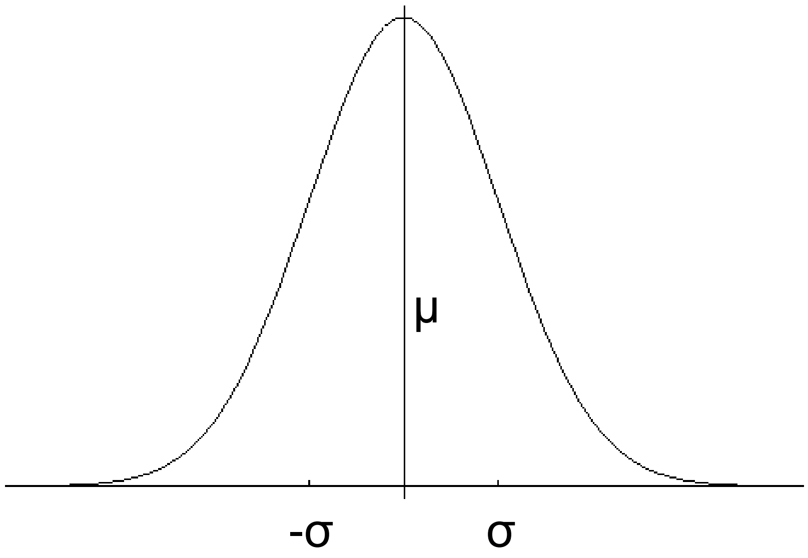
1. Gaussians

Gaussians

Univariate

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

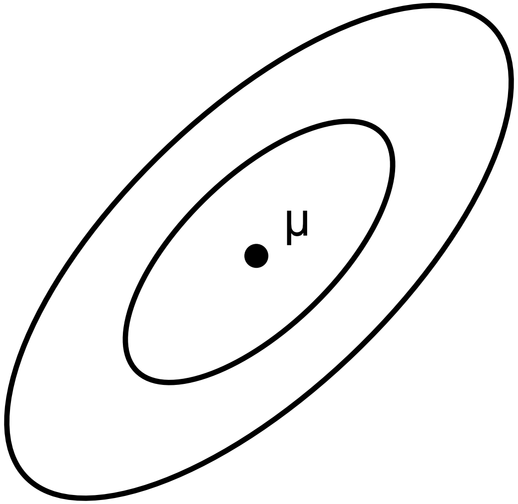
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$



Multivariate

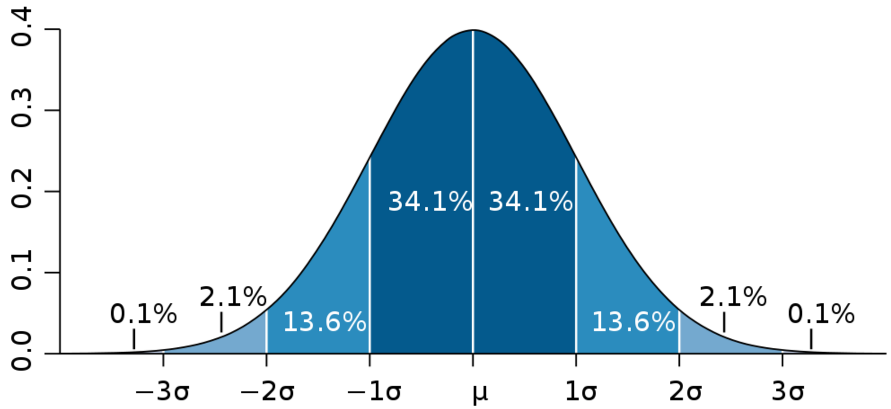
$$p(\mathbf{x}) \sim N(\mu, \Sigma):$$

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

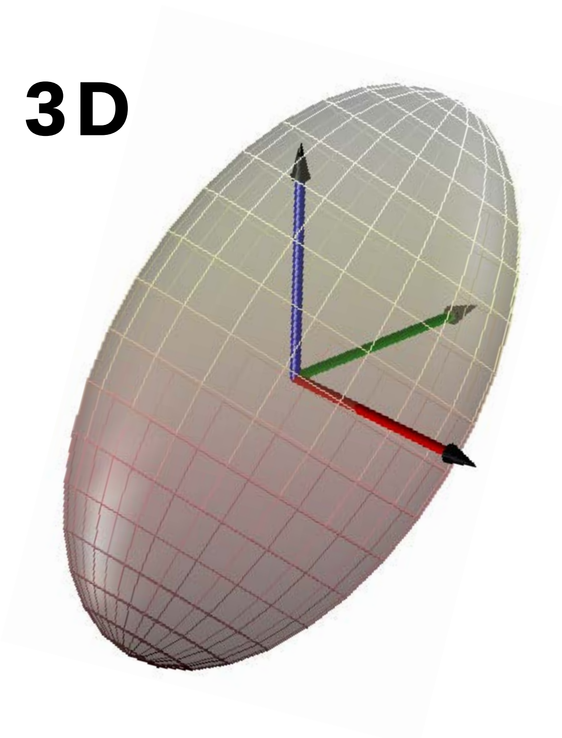


Gaussians

1D



3D



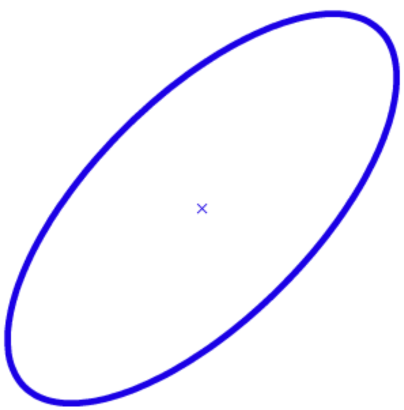
2D

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

$\lambda_1 = 0.007$

$\lambda_2 = 0.033$

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



Properties of Gaussians

- Univariate case

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

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

Properties of Gaussians

- Multivariate case

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

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

(where division "–" denotes matrix inversion)

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

2. The Kalman Filter

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

$$z_t = C_t x_t + \delta_t$$

Discrete Kalman Filter

$n \times n$: state evolution from t-1 to t without controls or noise

$n \times l$: state evolution under control u_t

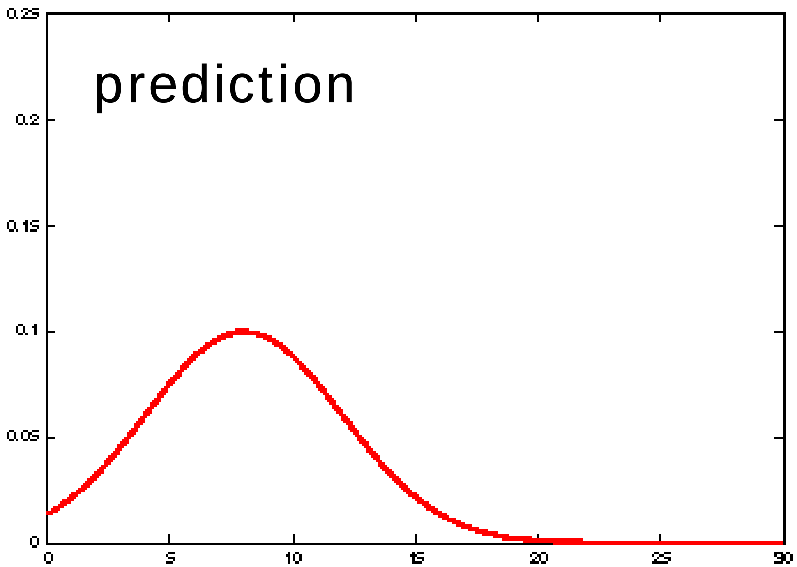
$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

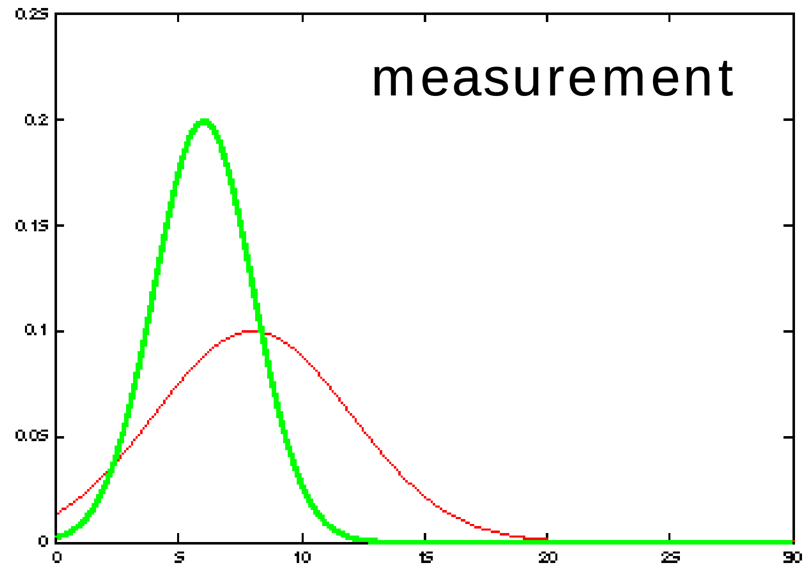
Random variables representing the system / measurement noise, independent and normally distributed with covariance Q_t and R_t respectively.

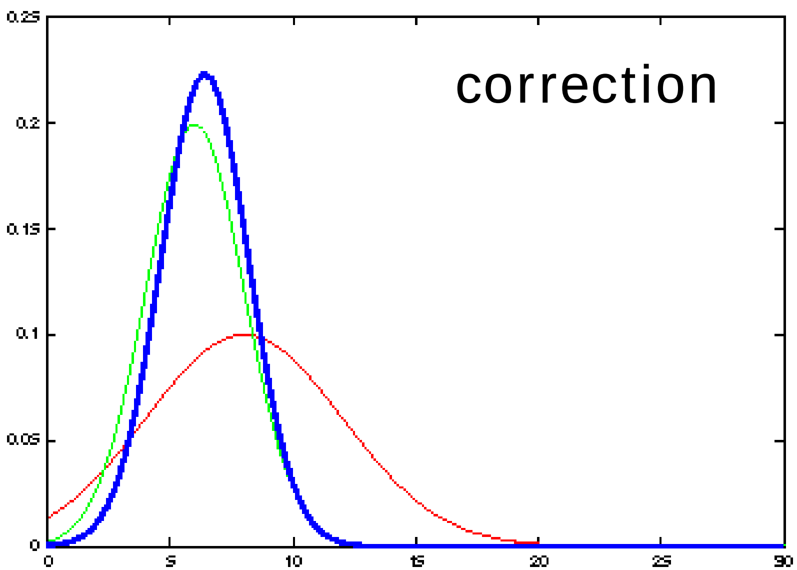
$$z_t = C_t x_t + \delta_t$$

$k \times n$: mapping state x_t to observation z_t

Kalman Filter Updates in 1D

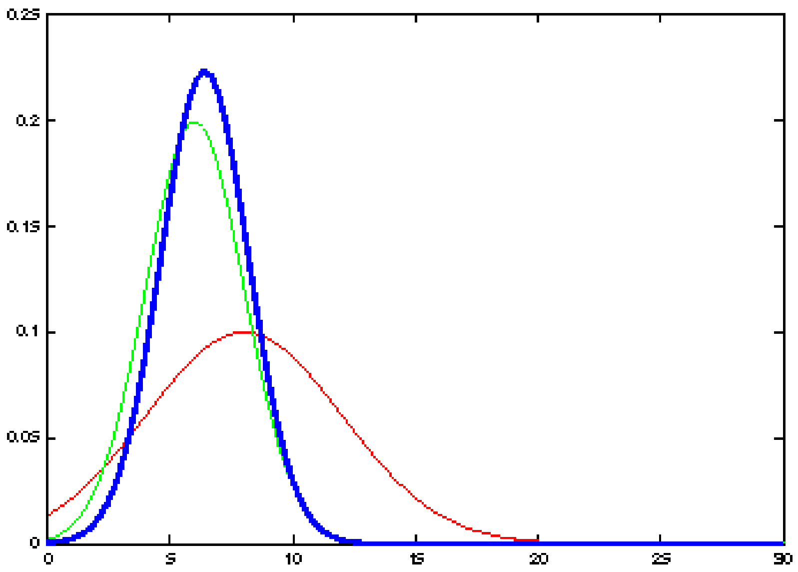






It's a weighted mean!

Kalman Filter Updates in 1D



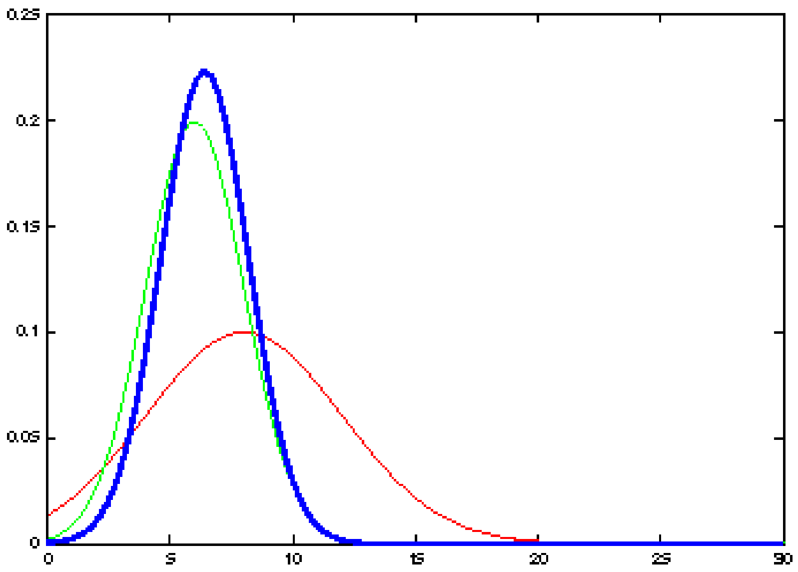
How to get the blue curve?

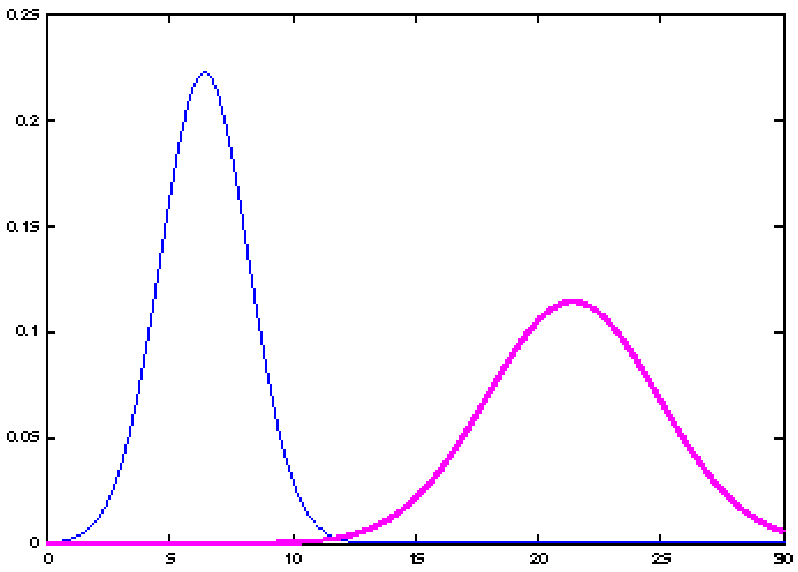
Kalman correction step

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

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

Kalman Filter Updates in 1D





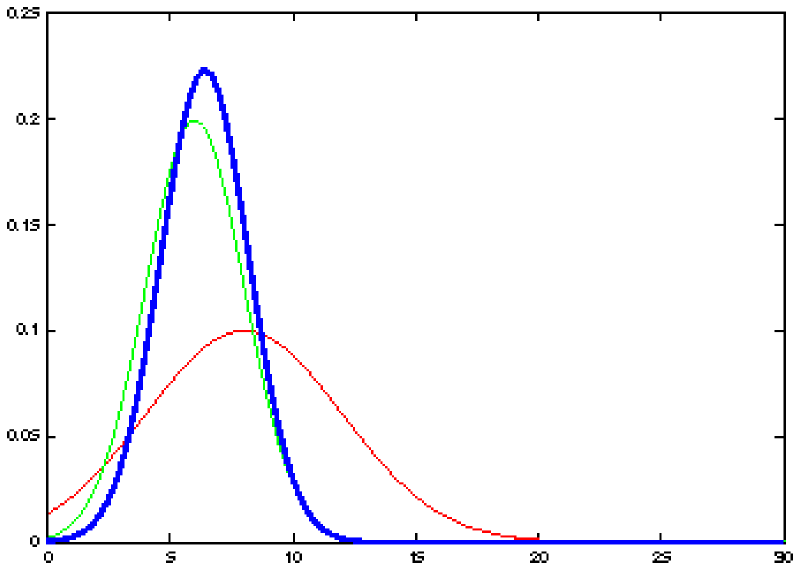
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

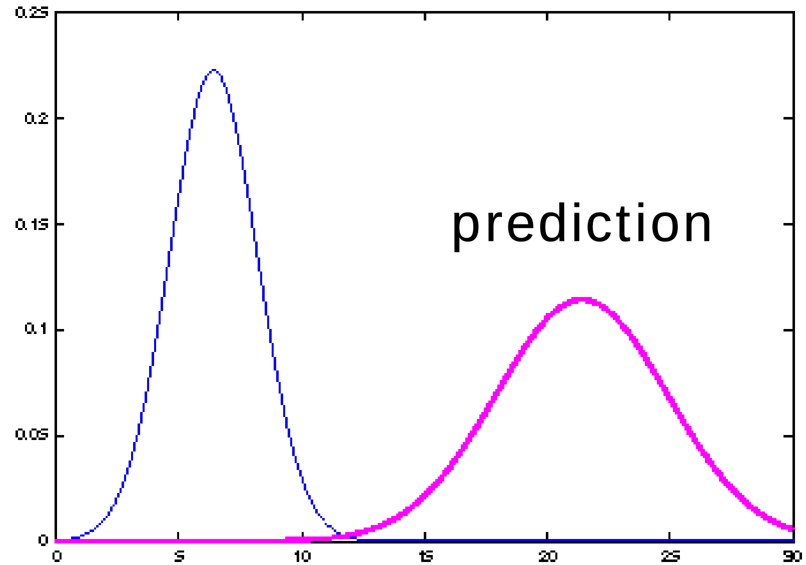
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t \end{cases}$$

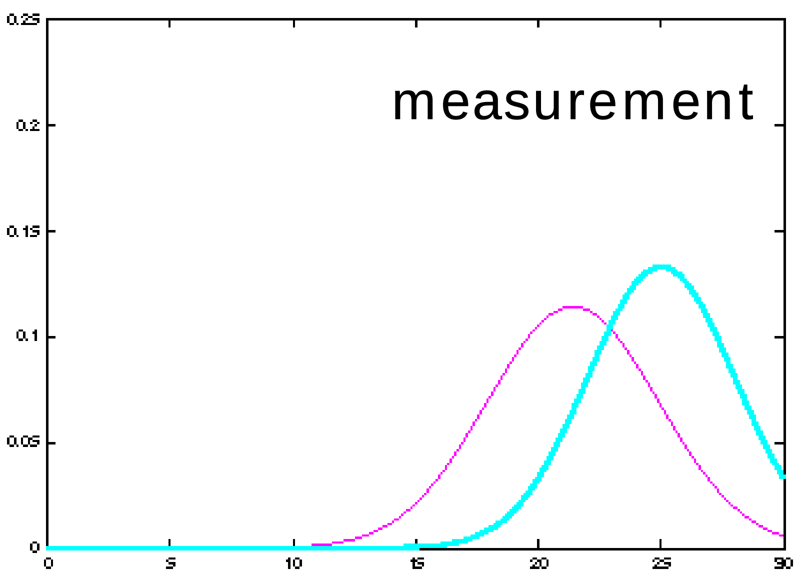
How to get the magenta curve?

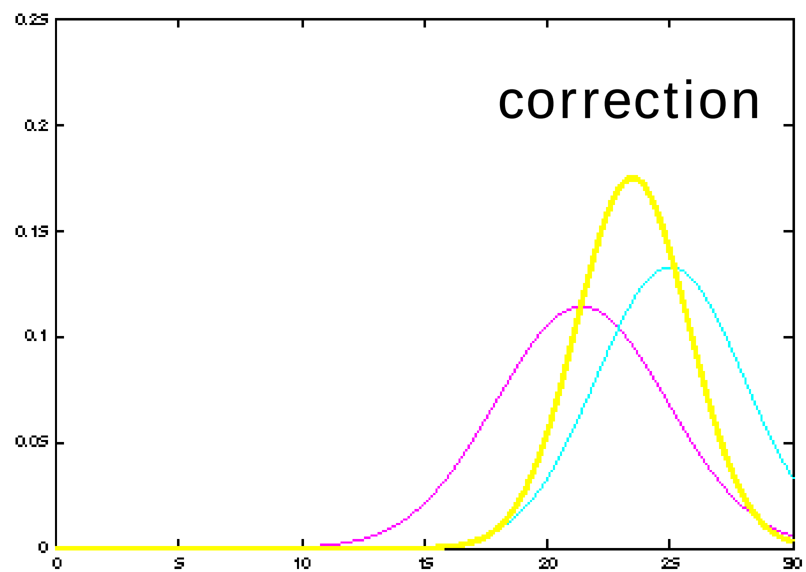
Kalman prediction step

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:

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

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

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

$$\Downarrow \qquad \Downarrow$$

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

Linear Gaussian Systems: Dynamics

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

$$\Downarrow \qquad\qquad \Downarrow$$

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

$$\Downarrow$$

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

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

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

Linear Gaussian Systems: Observations

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

$$z_t = C_t x_t + \delta_t$$

$$p(z_t \mid x_t) = N\left(z_t; C_t x_t, R_t\right)$$

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

$$\Downarrow \qquad \Downarrow$$

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

Linear Gaussian Systems: Observations

$$bel(x_t) = \quad \eta \quad p(z_t \mid x_t) \qquad\qquad \overline{bel}(x_t)$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

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

$$\Downarrow$$

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

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

3. The KF Algorithm

Kalman Filter Algorithm

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

   Prediction:

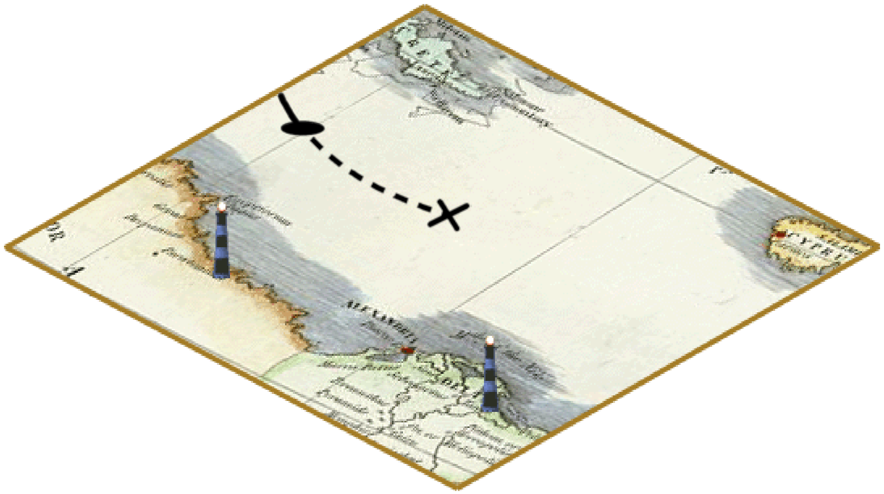
2. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
3. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$

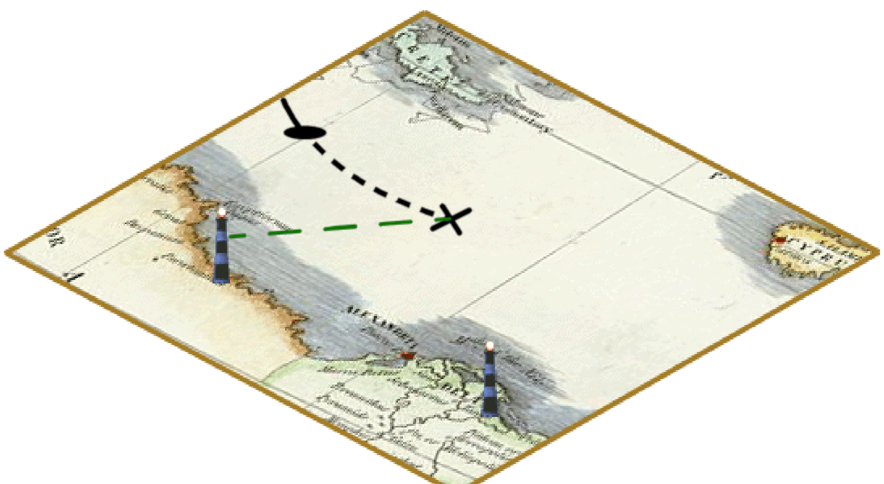
   Correction:

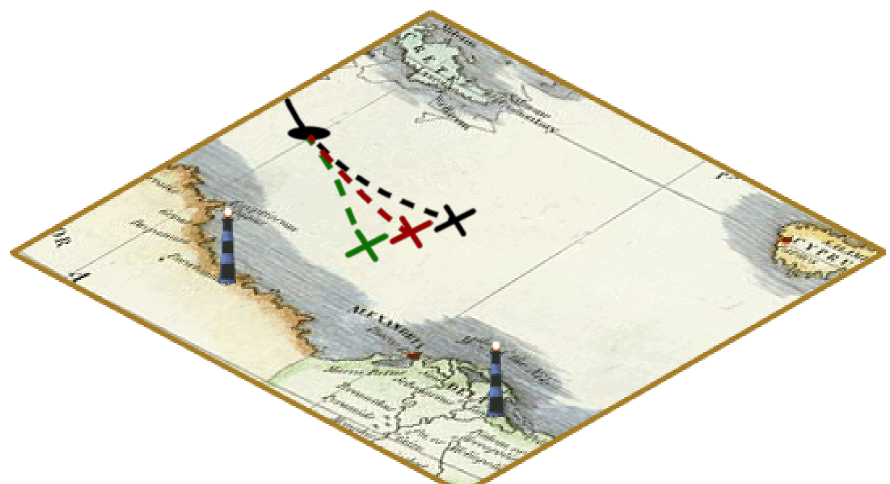
4. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$
5. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
6. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

7. Return μ_t, Σ_t

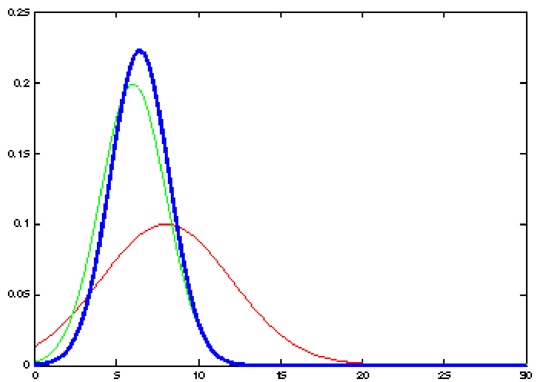
Kalman Filter Algorithm





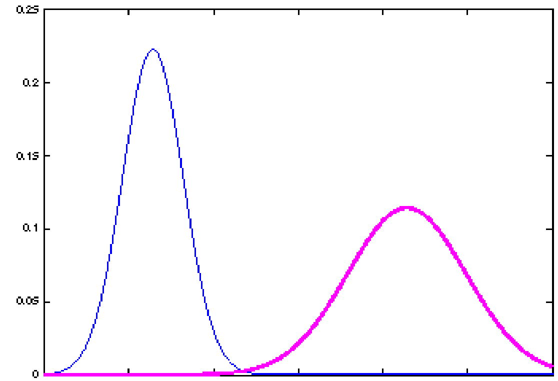


The Prediction-Correction-Cycle

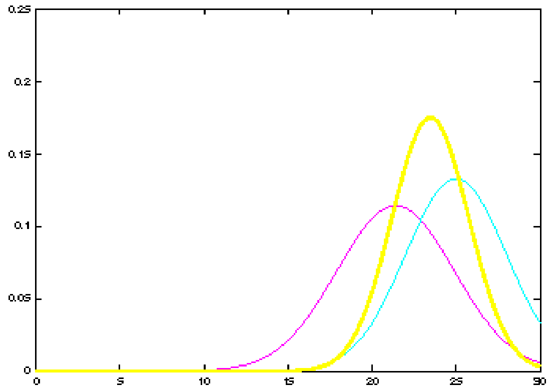


Prediction

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

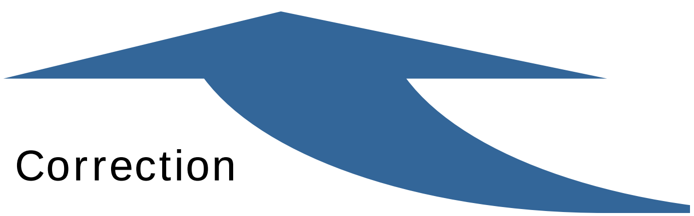


The Prediction-Correction-Cycle

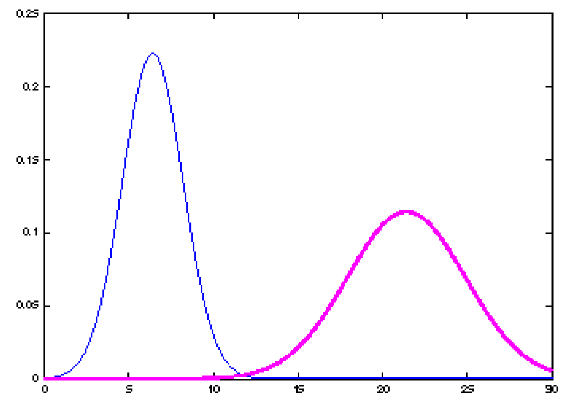


$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$$

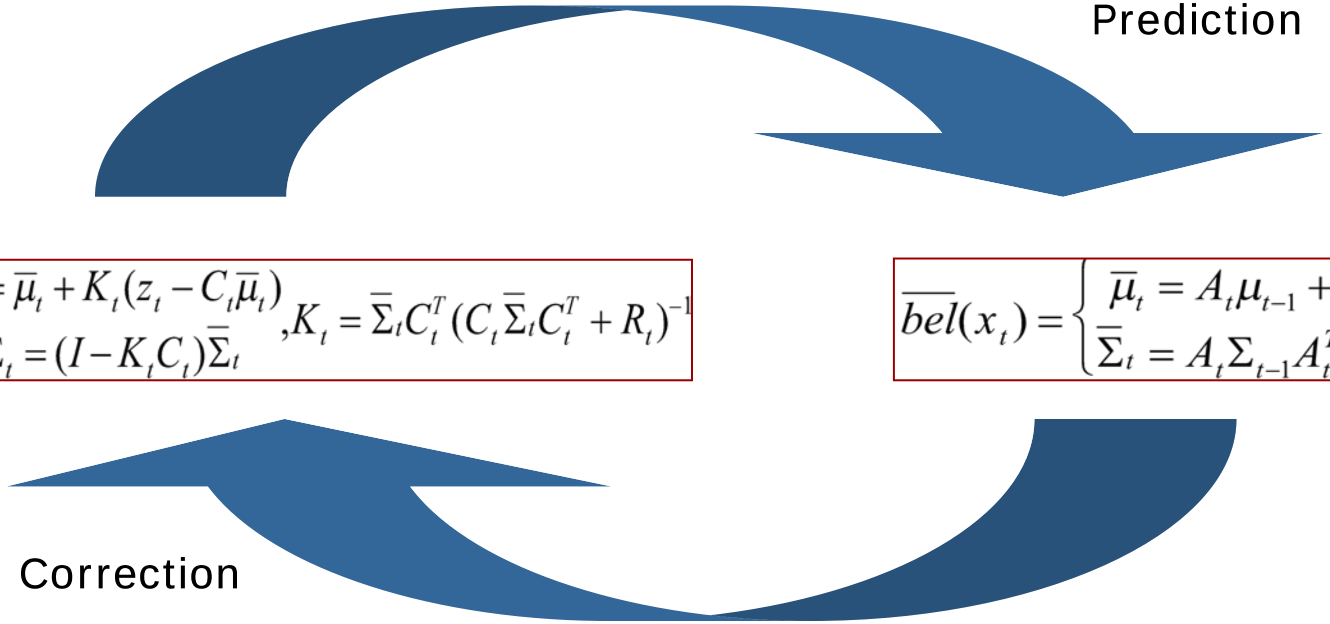
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t \end{cases}$$



Correction



The Prediction-Correction-Cycle



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

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

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