Recursive Bayes filter
Particular case: Kalman filter

Dr Alexandru Stancu
Dr Mario Martinez

Kalman filter

- **Gaussian PDF**
- **Linear or linearised models for the system and the observation**

Consider a linear model for a system:

$$x_k = Ax_{k-1} + Bu_k + q_k$$

$$z_k = Cx_k + r_k$$

Kalman filter

Gaussian PDFs:

$x_k \sim \mathcal{N}(\mu_k, \Sigma_k)$ → current state

$x_{k-1} | \mathrm{U}_{k-1}, Z_{k-1} \sim \mathcal{N}(\mu_{k-1}, \Sigma_{k-1})$ → previous state

$x_k | \mathrm{x}_{k-1}, u_k \sim \mathcal{N}(A\mu_{k-1} + Bu_k, \mathrm{Q}_{k-1})$ → motion model

$z_k | \mathrm{x}_\mathrm{k} \sim \mathcal{N}(Cx_k, \mathrm{R}_k)$ → observation model

Kalman filter

Similar to Bayes filter, Kalman filter follows **prediction step** and **correction step** to estimate the *belief* by incorporating the control signal u_k and the measurement z_k .

Kalman filter – prediction step

Prediction step:

$$\begin{bmatrix} x_k | \mathrm{U}_k, Z_{k-1} \\ x_{k-1} | \mathrm{U}_{k-1}, Z_{k-1} \end{bmatrix}$$

We use Marginalisation and Conditioning properties for Gaussian distributions:

General considerations for Marginalisation and Conditioning:

1. Given $X = \begin{bmatrix} U \\ V \end{bmatrix}$ and $X \sim \mathcal{N}(\mu_X, \Sigma_X)$

Σ_X is a multi-variable Gaussian distribution

then, the marginals are also Gaussian:

$U \sim \mathcal{N}(\mu_U, \Sigma_U)$ $\quad$ $V \sim \mathcal{N}(\mu_V, \Sigma_V)$

Kalman filter – prediction step

2. The conditionals are also Gaussians:

$$U|V \sim \mathcal{N}\left(\mu_{\mathrm{U}|V}, \Sigma_{\mathrm{U}|V}\right)$$

The Marginalisation and the Conditioning are the most important properties for Kalman filter.

For Gaussian case, Marginalisation is very easy:

$$X = \begin{bmatrix} U \\ V \end{bmatrix}, \qquad X \sim \mathcal{N}(\mu_X, \Sigma_X)$$

$$\mu_X = \begin{bmatrix} \mu_U \\ \mu_V \end{bmatrix}, \qquad \Sigma_X = \begin{bmatrix} \Sigma_U & \Sigma_{UV} \\ \Sigma_{VU} & \Sigma_V \end{bmatrix}$$

Kalman filter – prediction step

The marginal distribution is:

$$p(U) = \int p(U,V)dV \sim \mathcal{N}(\mu, \Sigma)$$

where: $\mu = \mu_U$

$\Sigma = \Sigma_U$

Therefore, If I have a high dimensional Gaussian distribution and I want to compute the marginal for a small number of elements, then I just need to cut out a part of the mean vector and cut out a part of the covariance matrix.

Kalman filter – prediction step

Conditioning is not easy. The proof is out of the scope of this unit. So, the conditional property will be provided without proof.

Given $p(X) = p\begin{bmatrix} U \\ V \end{bmatrix} \sim \mathcal{N}(\mu_X, \Sigma_X)$

where: $\mu_X = \begin{bmatrix} \mu_U \\ \mu_V \end{bmatrix}$ and $\Sigma_X = \begin{bmatrix} \Sigma_U & \Sigma_{UV} \\ \Sigma_{VU} & \Sigma_U \end{bmatrix}$

Then the conditional distribution is as follows:

$$p(U|V) = \frac{p(U,V)}{p(V)} \sim \mathcal{N}\left(\mu_{\mathrm{U}|V}, \Sigma_{\mathrm{U}|V}\right)$$

$$\mu_{\mathrm{U}|V} = \mu_U + \Sigma_{UV}\Sigma_V^{-1}(V - \mu_V)$$

$$\Sigma_{\mathrm{U}|V} = \Sigma_U - \Sigma_{UV}\Sigma_V^{-1}\Sigma_{UV}^T$$

Kalman filter – prediction step

Observations:

1. Σ_V^{-1} means that if I have a high dimensional Gaussian distribution and I want to estimate just a small part out of that, given I know the rest, is very costly operation because I need to invert a large part of the matrix.

Example: I want to estimate the pose of the robot given the landmarks position. Therefore, I need to compute the inverse, which is quite expensive.

Kalman filter – prediction step

Observations:

2. If we don't know anything (almost) about *V*, so we have big uncertainty $\Sigma_v \gg 0$, hence:

$$\Sigma_V^{-1} \cong 0 \qquad \mu_{U|V} = \mu_U \qquad \Sigma_{U|V} = \Sigma_U$$

Note that $p(U|V)$ is the probability of *U* given *V*, and if we don't know anything about *V*, then $p(U|V)$ is almost the same as $p(U)$.

If Σ_V is very small then Σ_V^{-1} is very large; hence it has large influence in $\mu_{U|V}$ and $\Sigma_{U|V}$.

Kalman filter – prediction step

Analogy with the linear system:

V in our case is $x_{k-1}|U_{k-1}, Z_{k-1}$

$U|V$ in our case is $x_k|x_{k-1}, U_k$ $\Rightarrow$ motion model

We want to find:

U which in our case is $x_k|U_k, Z_{k-1}$

$$\begin{bmatrix} U \\ V \end{bmatrix} \rightarrow \begin{bmatrix} x_k|U_k, Z_{k-1} \\ x_{k-1}|U_{k-1}, Z_{k-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} A\mu_{k-1} + Bu_k \\ \mu_{k-1} \end{bmatrix}, \begin{bmatrix} A\Sigma_{k-1}A^T + Q_k & A\Sigma_{k-1} \\ (A\Sigma_{k-1})^T & \Sigma_{k-1} \end{bmatrix} \right)$$

Kalman filter – prediction step

Gaussian distribution for the motion model:

$$p(x_k|x_{k-1},u_k)$$
$$= (\det(2\pi Q_k))^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_k - Ax_{k-1} - Bu_k)^T Q_k^{-1}(x_k - Ax_{k-1} - Bu_k)\right)$$

Gaussian distribution for observation model:

$$p(z_k|x_k) = \det(2\pi R_k)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_k - Cx_k)^T R_k^{-1}(z_k - Cx_k)\right)$$

Q_k is the noise of the motion and R_k is the measurement noise.

Kalman filter – prediction step

$$x_k | U_k, Z_{k-1} \sim \mathcal{N}(A\mu_{k-1} + Bu_k, A\Sigma_{k-1}A^T + Q_k)$$

$$\hat{x}_k = x_k | U_k, Z_{k-1} \sim \mathcal{N}\left(\hat{\mu}_k, \hat{\Sigma}_k\right)$$

$$\hat{\mu}_k = A\mu_{k-1} + B\mu_k$$

$$\hat{\Sigma}_k = A\Sigma_{k-1}A^T + Q_k$$

$$\overline{bel}(x_k) = p(\hat{x}_k)$$

Kalman filter – correction step

Correction step:

We address conditional distribution again. Let's consider:

$$U \equiv \hat{x}_k$$

$$V \equiv z_k$$

$$U|V \equiv \hat{x}_k|z_k$$

Chain rule:

$$p(\hat{x}_k|z_k) = \frac{p(\hat{x}_k, z_k)}{p(z_k)} \sim \mathcal{N}\left(\mu_{\hat{x}_k|z_k}, \Sigma_{\hat{x}_k|z_k}\right)$$

$$\begin{bmatrix} \hat{x}_k \\ z_k \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{\mu}_k \\ C\hat{\mu}_k \end{bmatrix}, \begin{bmatrix} \hat{\Sigma}_k & \hat{\Sigma}_k C^T \\ C\hat{\Sigma}_k & C\hat{\Sigma}_k C^T + R_k \end{bmatrix}\right)$$

Kalman filter – correction step

Therefore, by applying equation 2.42 from lecture notes (Conditioning):

$$\mu_{U|V} = \mu_{\hat{x}_k|z_k} = \hat{\mu}_k + \hat{\Sigma}_k C^T \left(C\hat{\Sigma}_k C^T + R_k\right)^{-1} [z_k - C\hat{\mu}_k]$$

where:

$\hat{\Sigma}_k C^T \left(C\hat{\Sigma}_k C^T + R_k\right)^{-1} = K_k$ is the Kalman Gain

z_k is the real measurement

$C\hat{\mu}_k$ is the predicted measurement (given by the observation model)

$$\Sigma_{\hat{x}_k|\mathrm{z_k}} = \hat{\Sigma}_k - \hat{\Sigma}_k C^T \left(C\hat{\Sigma}_k C^T + R_k\right)^{-1} \left(C\hat{\Sigma}_k\right)$$

Kalman filter recursive algorithm

Kalman filter algorithm (**inputs:** $\mu_{k-1}, \Sigma_{k-1}, u_k, z_k$, **outputs:** μ_k, Σ_k):

Prediction step:

$$\hat{\mu}_k = A\mu_{k-1} + Bu_k$$

$$\hat{\Sigma}_{\mathrm{k}} = A\Sigma_{k-1}A^T + \mathrm{Q}_k$$

Correction step:

$$K_k = \hat{\Sigma}_k C^T \left(C\hat{\Sigma}_k C^T + R_k\right)^{-1}$$

$$\mu_k = \hat{\mu}_k + K_k[z_k - C\hat{\mu}_k]$$

$$\Sigma_k = \hat{\Sigma}_k - K_k C\hat{\Sigma}_k$$

Kalman filter

Particular cases:

1. In the case of a perfect exteroceptive sensor, i.e., $R_k \cong 0$:

$$K_k = \hat{\Sigma}_k C^T \left(C \hat{\Sigma}_k C^T\right)^{-1} = \hat{\Sigma}_k C^T (C^T)^{-1} \hat{\Sigma}_k^{-1} C^{-1} = C^{-1}$$

$$\mu_k = \hat{\mu}_k + C^{-1}(z_k - C\hat{\mu}_k)$$

$$\mu_k = C^{-1} z_k \quad \Longrightarrow$$

we trust only the exteroceptive sensor,
we don't trust the prediction model at all.

Kalman filter

Particular cases:

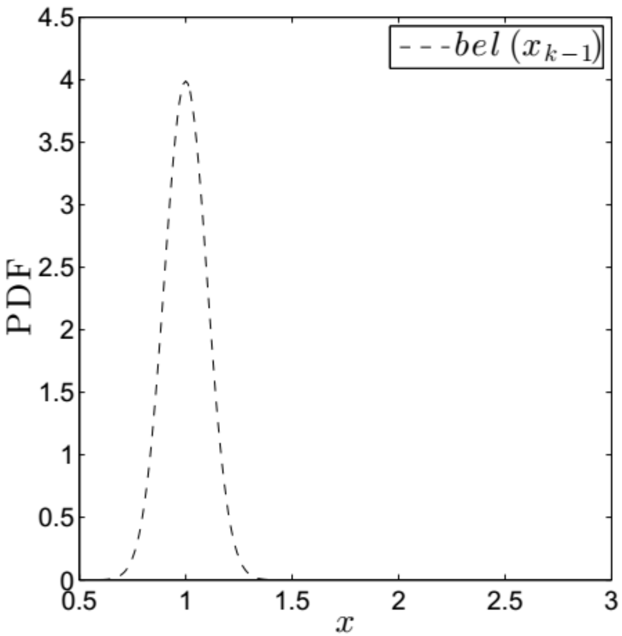
2. In case of a bad sensor R_k is very big, i.e., $R_k = \infty$:

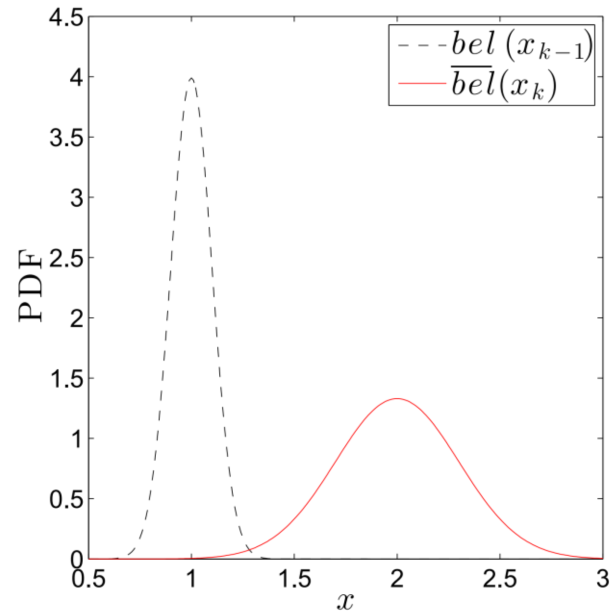
$$K_k = \hat{\Sigma}_k C^T \left(C \hat{\Sigma}_k C^T + R_k \right)^{-1} = \hat{\Sigma}_k C^T \left(C \hat{\Sigma}_k C^T + \infty \right)^{-1}$$

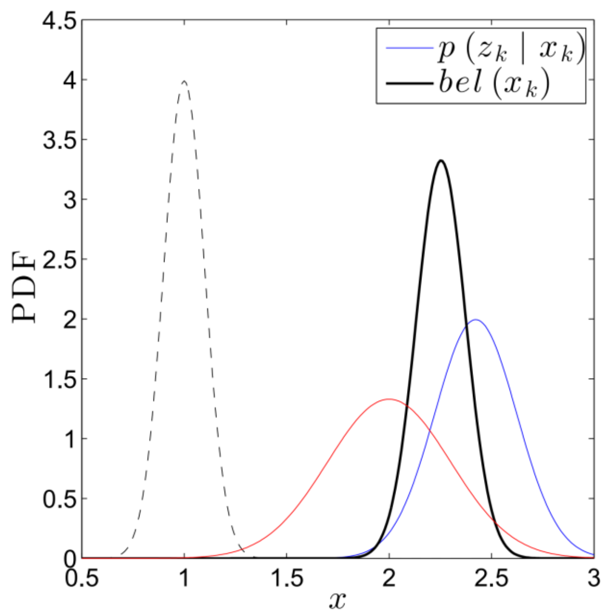
$$K_k = \hat{\Sigma}_k C^T (\infty)^{-1} = 0$$

$\mu_k = \hat{\mu}_k$ → we trust only the prediction model,
we don't trust the exteroceptive sensor at all.

Kalman filter - example







Next topic

Kalman filter applied for map-based localisation