The Kalman Filter

Henrik I Christensen

Contextual Robotics
UC San Diego
La Jolla, CA
hichristensen@ucsd.edu

Outline

Introduction

- Recapitulation of system models
- Integration of stochastic variables
- How to perform fusion in a more general sense
- Doing this in a non-linear system

Outline

State space model

$$\begin{aligned} s_t &= F s_{t-1} + G u_t + w_t \\ z_t &= H s_t + v_t \end{aligned}$$

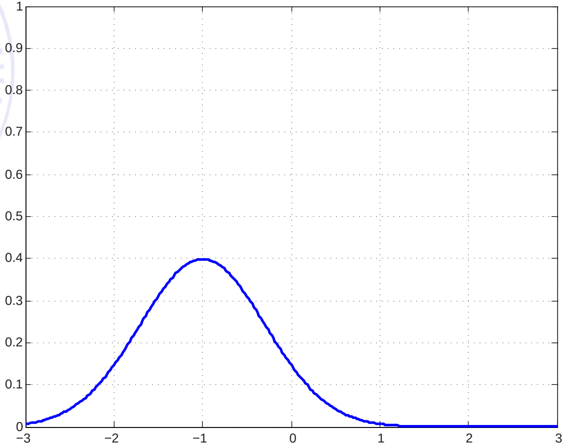
- where F is the system model, G is the deterministic input, H is a prediction of where features are in the world, w is the system noise, and v is the measurement noise
- $p(w) \sim N(0, Q)$
- $p(v) \sim N(0, R)$

State state model example

$$
\begin{aligned}
x_t &= x_{t-1} + v_{t-1}T + \frac{1}{2}a_{t-1}T^2 \\
v_t &= v_{t-1} + a_{t-1}T \\
a_t &= a_{t-1} \\
s_t &= \begin{bmatrix} x_t \\ v_t \\ a_t \end{bmatrix} \\
\mathbf{F} &= \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \\
\mathbf{G} &= [001]^T
\end{aligned}
$$

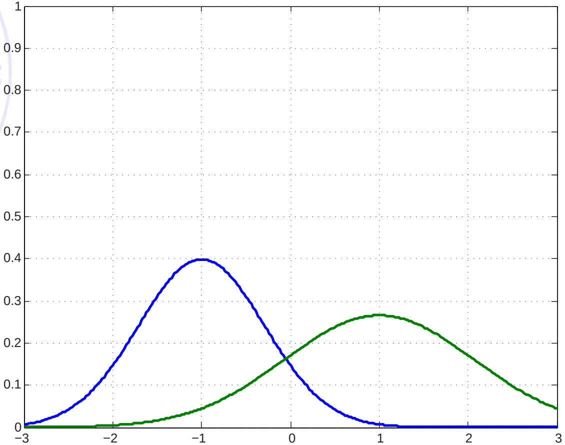
Outline

A small example - I



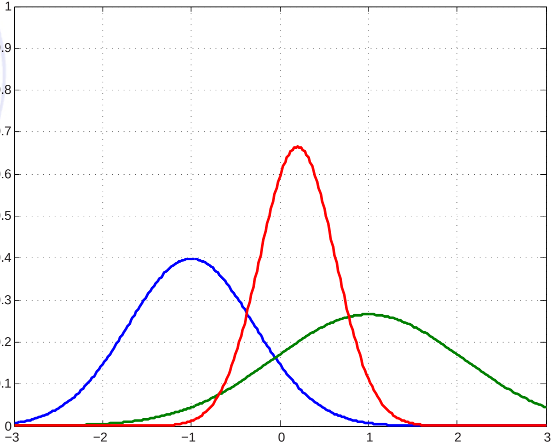
$N(-1, 1)$

A small example - II



$$N(-1,1) \& N(1,1.5)$$

A small example - III - Joint value?



$N(?, ?)$

Outline

Fusion of stochastic variables

- Assume two measurement x_1 and x_2 with associated uncertainties σ_1 and σ_2. How does one generate an optimal estimate $\hat{x}$?
- Doing a weighted least square:

$$S = \sum_{i=1,2} w_i(\hat{x} - x_i)^2$$

  what are the optimal weights w_i?
- From $\frac{\partial S}{\partial \hat{x}} = 0$ we get ...

Fusion of stochastic variables

$$\hat{x} = \frac{\sum w_i q_i}{\sum w_i}$$

with $w_i = \frac{1}{\sigma_i^2}$ we get

$$\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2$$

and

$$\sigma_{\hat{x}} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

Weighted updating

- The update can be rewritten to

$$\hat{x} = x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(x_2 - x_1)$$

- I.e. the updating = the value + a correction term

Outline

The Kalman model

- Composed of a prediction and an update of the estimate
- Prediction is an estimate of what the system ought to be given our history / state.
- Update is based on the difference between expectation and actual measurements.

Kalman Prediction

$$
\begin{aligned}
s_{t|t-1} &= F s_{t-1|t-1} + G u_t \\
\Sigma_{t|t-1} &= F \Sigma_{t-1|t-1} F^T + Q_t
\end{aligned}
$$

where Q is the uncertainty of the model / system noise

Kalman Updating

$$
\begin{aligned}
s_{t|t} &= s_{t|t-1} + K_t(z_t - Hs_{t|t-1}) \\
K_t &= \Sigma_{t|t-1} H^T S_t^{-1} \\
S_t &= H\Sigma_{t|t-1} H^T + R_t \\
\Sigma_{t|t} &= (I - K_t H)\Sigma_{t|t-1}
\end{aligned}
$$

Kalman Updating

$$
\begin{aligned}
s_{t|t} &= s_{t|t-1} + K_t(z_t - Hs_{t|t-1}) \\
K_t &= \Sigma_{t|t-1} H^T S_t^{-1} \\
S_t &= H\Sigma_{t|t-1} H^T + R_t \\
\Sigma_{t|t} &= (I - K_t H)\Sigma_{t|t-1}
\end{aligned}
$$

A bit of analysis

- Consider prior estimation error: $e_{t|t-1} = s_t - s_{t|t-1}$ and
- posteriori error $e_{t|t} = s_t - s_{t|t}$
- The error covariances are them:

$$\begin{aligned} \Sigma_{t|t-1} &= E[e_{t|t-1} e_{t|t-1}^T] \\ \Sigma_{t|t} &= E[e_{t|t} e_{t|t}^T] \\ K_t &= \Sigma_{t|t-1} H^T (\Sigma_{t|t-1} H^T + R_t)^{-1} \end{aligned}$$

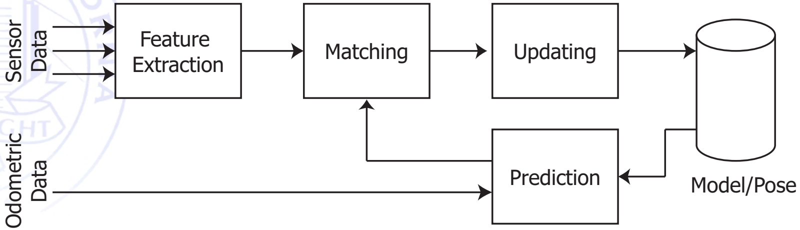
$$lim_{R \to 0} K_t = H^{-1}$$

$$lim_{\sigma_{t|t-1} \to 0} K_t = 0$$

Kalman interpretation

- F is the autonomous evolution
- H is the measurement prediction $p(z|s)$ ie a prediction of where features in the world are in the sensory frame
- Σ is the uncertainty in the pose/state estimate $s_{t|t}$.
- S_t is the uncertainty in the sensory measurements
- R_t is the sensor noise
- Q_t is the uncertainty in the system model. How good is the model?

Example use of the Kalman filter

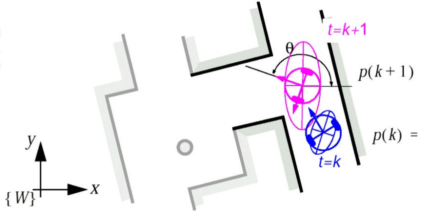


- Prediction / Update as described above
- Matching based on the Mahalanobis distance

$$M = z_t S_t^{-1} z_t^T$$

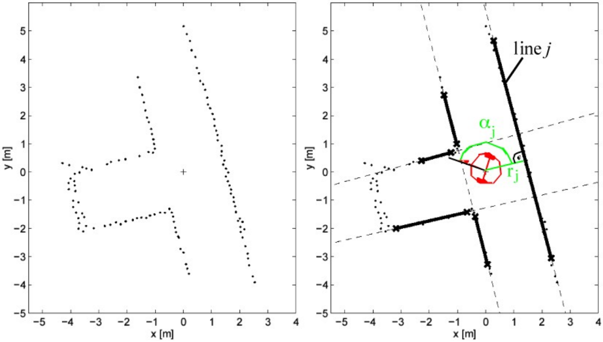
- A validation gate may be used: $M < \rho$

Kalman Example – Prediction

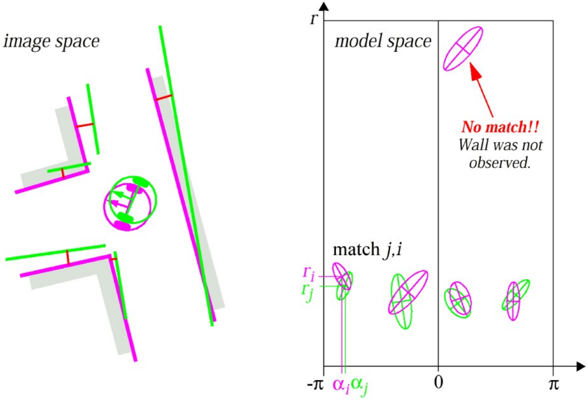


- Kinematic prediction of position and uncertainty
- Standard model

Kalman Example – Features

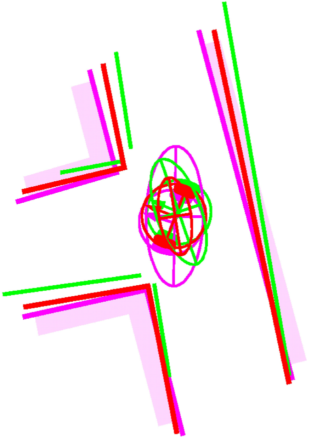


Kalman Example – Matching



using nearest neighbour and predicted features matching is easy

Kalman Example – Updating



- Matched features generates an error in estimates (purple - model), (green - measured), and (red - update)
- Updating is now easy

Outline

Non-linear systems

- Many systems are non-linear, in these cases a linearization can be used
- Consider

$$
\begin{aligned}
s_t &= f(s_{t-1}, u_{t-1}, w_{t-1}) \\
z_t &= h(s_t, v_k)
\end{aligned}
$$

A linearization of the system

$$
\begin{aligned}
s_{t|t} &\approx f(s_{t-1|t-1}, u_{t-1}, 0) + F(s_{t-1} - s_{t-1|t-1}) + W w_{t-1} \\
z_t &\approx h(s_{t|t}, 0) + H(s_t - s_{t|t}) + V v_k
\end{aligned}
$$

where

$$
\begin{aligned}
F_{ij} &= \frac{\partial f_i}{\partial s_j}(s_{t-1|t-1}, u_{t-1}, 0) \\
W_{ij} &= \frac{\partial f_i}{\partial w_j}(s_{t-1|t-1}, u_{t-1}, 0) \\
H_{ij} &= \frac{\partial h_i}{\partial s_j}(s_{t|t}, 0) \\
V_{ij} &= \frac{\partial h_i}{\partial v_j}(s_{t|t}, 0)
\end{aligned}
$$

The EKF computation

Time updating / prediction:

$$
\begin{aligned}
s_{t|t-1} &= f(s_{t-1|t-1}, u_{t-1}, 0) \\
\Sigma_{t|t-1} &= F_t \Sigma_{t-1|t-1} F_t^T + W_t Q_{t-1} W_t^T
\end{aligned}
$$

Measurement update computation

$$
\begin{aligned}
K_t &= \Sigma_{t|t-1} H_t^T (H_t \Sigma_{t|t-1} H_t^T + V_t R_t V_t)^{-1} \\
s_{t|t} &= s_{t|t-1} + K_t(z_t - h(s_{t|t-1}, 0)) \\
\Sigma_{t|t} &= (I - K_t H_t)\Sigma_{t|t-1}
\end{aligned}
$$

Outline

Kalman – Discussion

- Makes an assumption of a single pose estimate
- By far the most frequently used model
- Easy to compute.
- Estimation of F and H can be difficult. For non-linear systems that are Jacobians that must be computed for each step.