16-662 – Robot Autonomy

Kalman Filter

Katharina Muelling
kmuelling@nrec.ri.cmu.edu
NSH 4521
What we have learned so far?

• Motion planning assuming exact position information of objects and robots

• In most real world tasks exact world models are not available

• Sensor carry only partial information and are noisy

• State information can be captured partially by multiple sensors which are not time synchronized
Today’s Lecture

• Class Content
  – How to recover state information from sensor data
  – Probabilistic state distributions
  – Kalman Filter
  – Extended Kalman Filter
Why are Kalman Filter cool?

Motivating Example

Filter noisy sensor information to predict ball trajectories
Why are Kalman Filter cool?

Motivating Example

State estimation from multiple sensors
Why are Kalman Filter cool?

Motivating Example

Indirect measurement of state information
Why are Kalman Filter cool?

Motivating Example
Flying to the Moon

No way to measure trajectory of the rocket

Sensors: sextant, gyroscope
Problem Definition

**Given** noisy, indirect measurements $y$ and some control input $u$: Recover the true state $x$ that led to the observation

Example
Problem Definition

Given noisy, indirect measurements $y$ and some control input $u$: Recover the true state $x$ that led to the observation.

Example

How can we deal with the uncertainty?

1. Represent initial estimate as probability distribution
2. Sensor feedback
Probabilistic State Estimation

General Idea
Compensate for lost information by injecting new measurements into system

How?
Probabilistic process is two step process

Predict → Update

Given state estimate (prior) in form of a pdf: propagate state estimate with generative model
Correct prediction by merging the predicted pdf with new sensor measurement
Probabilistic robotics: dynamics of robot in environment characterized by
1. Transition distribution
2. Measurement distribution

**Belief**: Posterior distribution over states in environment given all previous sensor data and control input
Assumption 1: Discrete-time linear system

\[ \mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{\varepsilon}_t \]

\[ \mathbf{y}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{\delta}_t \]
Kalman Filter

Assumption 1: All noise is Gaussian

\[ x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \]
\[ y_t = H_t x_t + \delta_t \]

\[ \varepsilon_t \sim N(0, R) \]
\[ \delta_t \sim N(0, Q) \]

\[ p(x) \]

\[ \sigma \text{ increases or decreases spread of distribution} \]

\[ N(\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2} (x - \mu)^2\right\} \]
Kalman Filter

Assumption 1: All noise is Gaussian

\[ x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \]
\[ y_t = H_t x_t + \delta_t \]

\[ \varepsilon_t \sim N(0, R) \]
\[ \delta_t \sim N(0, Q) \]

Everything stays Gaussian!
Kalman Filter

Linear Kalman Filter

Input:

\[ x_{t-1}, u_t, y_t \]

Prediction:

\[ \hat{x}_t = Ax_{t-1} + Bu_t \]
\[ S_t = A S_{t-1} A^T + R_t \]
Kalman Filter

Initial Guess

\[ x_{t-1} \]
Kalman Filter

Prediction

\[ \hat{x}_t = Ax_{t-1} + Bu_t \]

\[ S_t = A S_{t-1} A^T + R_t \]

Move Forward

\( x_{t-1} \)

\( \hat{x}_t \)
Kalman Filter

Linear Kalman Filter

Update:

\[
K_t = \hat{\Sigma}_t H^T (H\hat{\Sigma}_t H^T + Q_t)^{-1}
\]

\[
x_t = \hat{x}_t + K (y_t - H\hat{x}_t)
\]

\[
\hat{\Sigma}_t = (I - K_t H)\hat{\Sigma}_t
\]

Kalman gain: Decides what to trust more, prediction or measurement

Innovation: Difference between measured and expected y
Kalman Filter

Update

\[ K_t = \hat{\Sigma}_t H^T (H\hat{\Sigma}_t H^T + Q_t)^{-1} \]

\[ x_t = \hat{x}_t + K (y_t - H\hat{x}_t) \]

\[ \hat{\Sigma}_t = (I - K_t H)\hat{\Sigma}_t \]
Kalman Filter

Linear Kalman Filter

Update:

\[
K_t = \hat{\Sigma}_t H^T (H\hat{\Sigma}_t H^T + Q_t)^{-1}
\]

\[
x_t = \hat{x}_t + K(y_t - H\hat{x}_t)
\]

\[
\Sigma_t = (I - K_t H)\hat{\Sigma}_t
\]

Assume we are very confident in our measurements, i.e., Q=0

\[
K_t = \hat{\Sigma}_t H^T (H\hat{\Sigma}_t H^T)^{-1} \rightarrow K_t H = I
\]

\[
x_t = \hat{x}_t + K(y_t - H\hat{x}_t) = \hat{x}_t + Ky_t - \hat{x}_t = H^{-1}y_t
\]

\[
\Sigma_t = (I - K_t H)\hat{\Sigma}_t = 0
\]
Kalman Filter

Linear Kalman Filter

Update:

\[
K_t = \hat{\Sigma}_t H^T (H\hat{\Sigma}_t H^T + Q_t)^{-1}
\]

\[
x_t = \hat{x}_t + K (y_t - H\hat{x}_t)
\]

\[
\Sigma_t = (I - K_t H)\hat{\Sigma}_t
\]

Assume we do not trust our measurements, i.e., \(Q=\infty\)

\[
K_t = \hat{\Sigma}_t H^T 0 = 0
\]

\[
x_t = \hat{x}_t
\]

\[
\Sigma_t = (I - 0H)\hat{\Sigma}_t = \hat{\Sigma}_t
\]
Kalman Filter

Linear Kalman Filter

Input: \[ x_{t-1}, u_t, y_t \]

Prediction: \[ \hat{x}_t = Ax_{t-1} + Bu_t \]
\[ \hat{\Sigma}_t = A\Sigma_{t-1}A^T + R_t \]

Update: \[ K_t = \hat{\Sigma}_tH^T (H\hat{\Sigma}_tH^T + Q_t)^{-1} \]
\[ x_t = \hat{x}_t + K_t(y_t - H\hat{x}_t) \]
\[ \Sigma_t = (I - K_tH)\hat{\Sigma}_t \]

State estimate: \[ \hat{x}_t, t \]

Kalman gain

Innovation
1) The real world is not linear!
2) Our belief might not be Gaussian
Problems?

Non-linear Dynamic Systems lead to non-linear functions

\[ x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \quad \text{and} \quad y_t = H_t x_t + \delta_t \]

\[ x_t = g(u_t, x_{t-1}) + \varepsilon \quad \text{and} \quad y_t = h(x_t) + \delta_t \]

Why is this a problem?
Extended Kalman Filter

Why are non-linear functions a problem?

Figure by S. Thrun, CS-223
Extended Kalman Filter

Why are non linear functions a problem?

Mapping Gaussian function with non linear function is not Gaussian

Kalman Filter cannot be applied anymore

Figure by S. Thrun, CS-223
Extended Kalman Filter

EKF solves the problem on non-linear functions through **local linearization** and then applying the KF

**Prediction**

\[
g(u_t, x) \approx g(u_t, x_{t-1}) + \frac{\partial g(u_t, x_{t-1})}{\partial x} (x - x_{t-1})
\]

\[
\approx \begin{bmatrix} G \\
\end{bmatrix}
\]

**Update**

\[
h(x) \approx h(\hat{x}_t) + \frac{\partial h(\hat{x}_t)}{\partial x} (x - \hat{x}_t)
\]

\[
\approx \begin{bmatrix} H \\
\end{bmatrix}
\]
Jacobian

Given: vector valued function $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$g(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{bmatrix}$$

The Jacobian is defined as

$$G = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \ldots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \ldots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}$$
Extended Kalman Filter

Figure by S. Thrun, CS-223
Extended Kalman Filter

Predict

\[ \hat{x}_t = g(u_t, x_{t-1}) \]

\[ \hat{\Sigma}_t = G_t \Sigma_t G_t^T + R_t \]

Update

\[ K_t = \hat{\Sigma}_t H_t^T (H_t \hat{\Sigma}_t H_t^T + Q_t)^{-1} \]

\[ x_t = \hat{x}_t + K_t(y_t - h(\hat{x}_t)) \]

\[ \Sigma_t = (I - K_t H_t) \hat{\Sigma}_t \]