AI Robotics

Kalman Filter

MISSISSIPPI STATE UNIVERSITY™

Overview

Robot Model

Sensor Measurements:
$Z_{0:t} = \{z_0, \dots, z_t\}$

Robot State (or pose):
$X_{0:t} = \{X_0, \dots, X_t\}$

sense

act

Landmark or Map:
$m = \{m_1, \dots, m_n\}$

Robot Controls:
$u_{0:t} = \{u_0, u_1, \dots, u_t\}$

State Estimation

Sensor Measurements:
$Z_{0:t} = \{z_0, \dots, z_t\}$

Robot State (or pose):
$X_{0:t} = \{X_0, \dots, X_t\}$

sense

Known

Unknown

act

Robot Controls:
$u_{0:t} = \{u_0, u_1, \dots, u_t\}$

Probabilistic State Estimation

$$P(x_t|u_1, z_1, u_2, z_2, \cdots, u_t, z_t) = P(x_t|z_{1:t}, u_{1:t})$$

Robot State

sense

Sensor Model

$$P(z_t|x_t)$$

Motion Model

act

$$P(x_t|x_{t-1}, u_t)$$

History

Rudolf E. Kálmán

From Wikipedia, the free encyclopedia

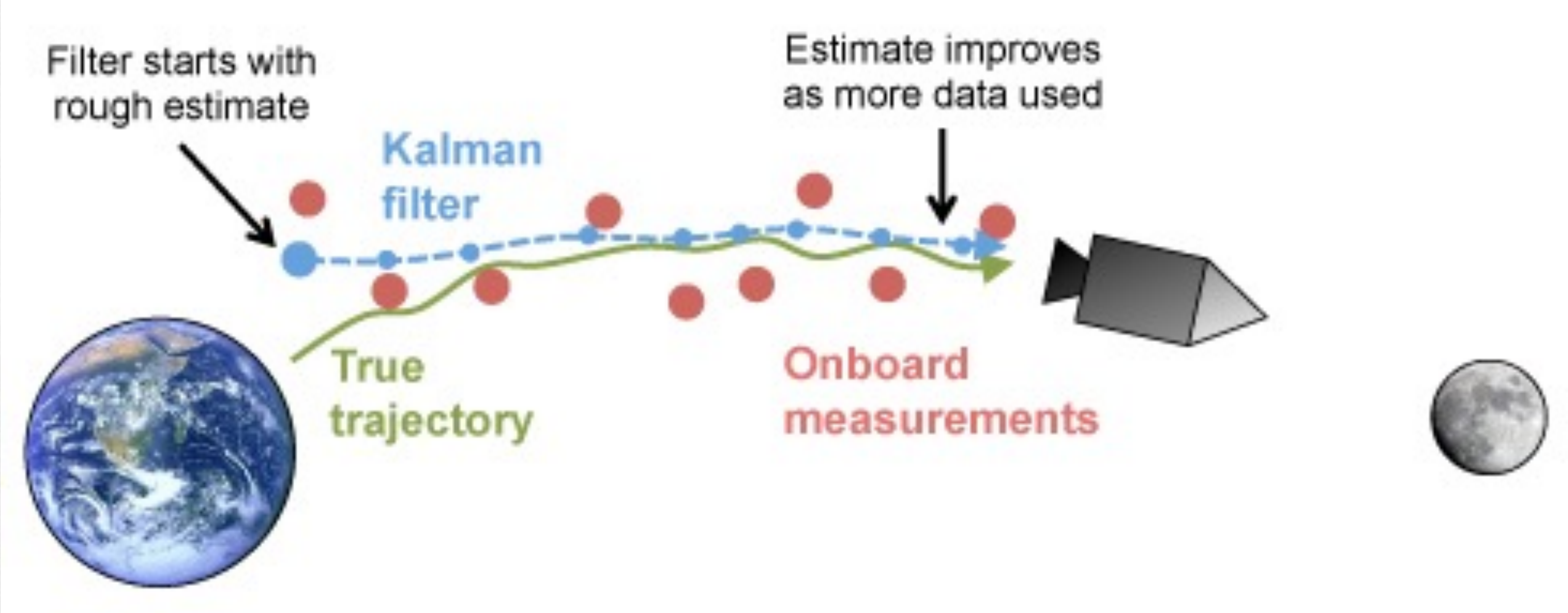
Rudolf Emil Kálmán[3] (May 19, 1930 – July 2, 2016) was a Hungarian-American electrical engineer, mathematician, and inventor. He is most noted for his co-invention and development of the Kalman filter, a mathematical algorithm that is widely used in signal processing, control systems, and guidance, navigation and control. For this work, U.S. President Barack Obama awarded Kálmán the National Medal of Science on October 7, 2009.[4]

Contents [hide]

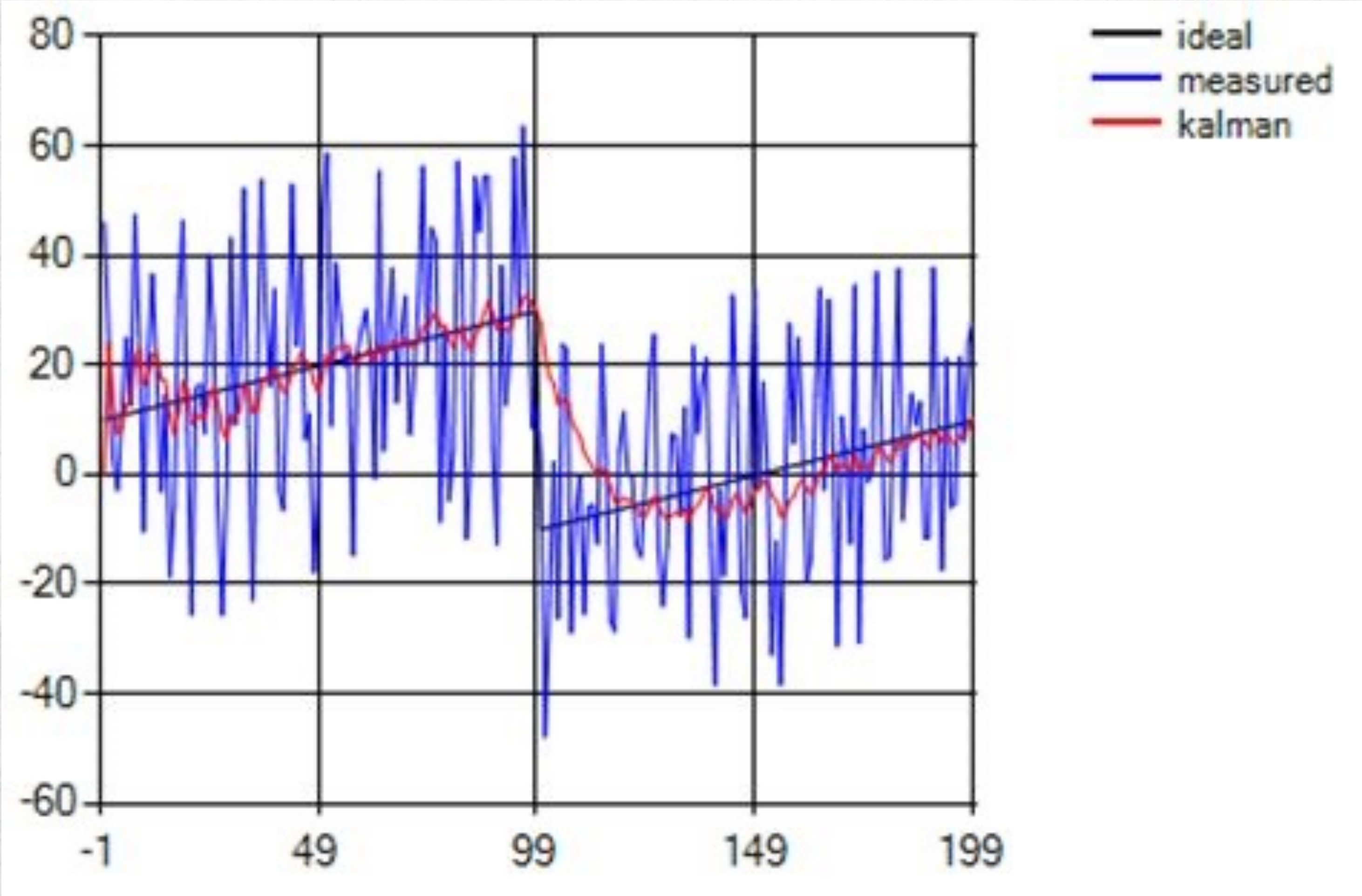
Rudolf E. Kálmán

Born	Rudolf Emil Kálmán[1] May 19, 1930 Budapest, Hungary
Died	July 2, 2016 (aged 86)[2] Gainesville, Florida

Rocket Science



State Estimation



Theory

Bayes Filter

$$Bel(x_t) = P(x_t \mid u_{1:t}, z_{1:t})$$

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

Sensor Model **Transition Model** **Belief State**

$$P(x_t \mid u_{1:t}, z_{1:t}) = \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) P(x_{t-1} \mid u_{1:t-1}, z_{1:t-1}) dx_{t-1}$$

Bayes Filter Algorithm

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

Previous State

Current control input

Current sensor measurement

Algorithm **Bayes_filter**($Bel(x_{t-1})$, u_t, z_t):

$$Bel'(x_t) = \int P(x_t \mid u_t, x_{t-1}) \, Bel\,(x_{t-1}) dx$$

Prediction Step: Motion Update

$$Bel(x_t) = \eta \, P(z_t \mid x_t) \, Bel'(x_t)$$

Correction Step: Measurement Update

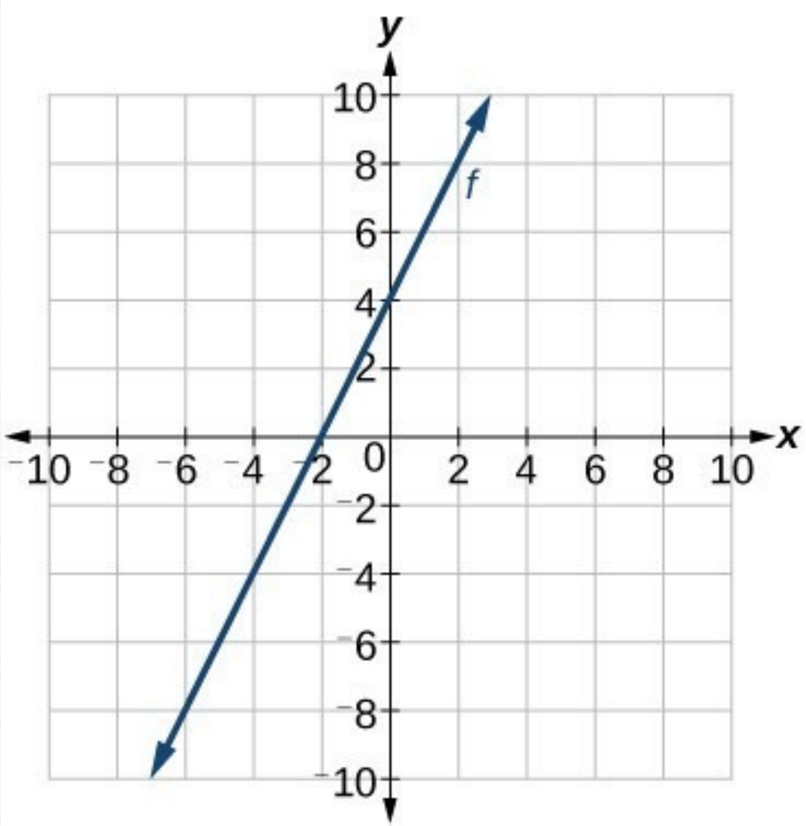
$$return \quad Bel(x_t)$$

Current State

Linear Dynamics

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

$$z_t = C_t x_t$$



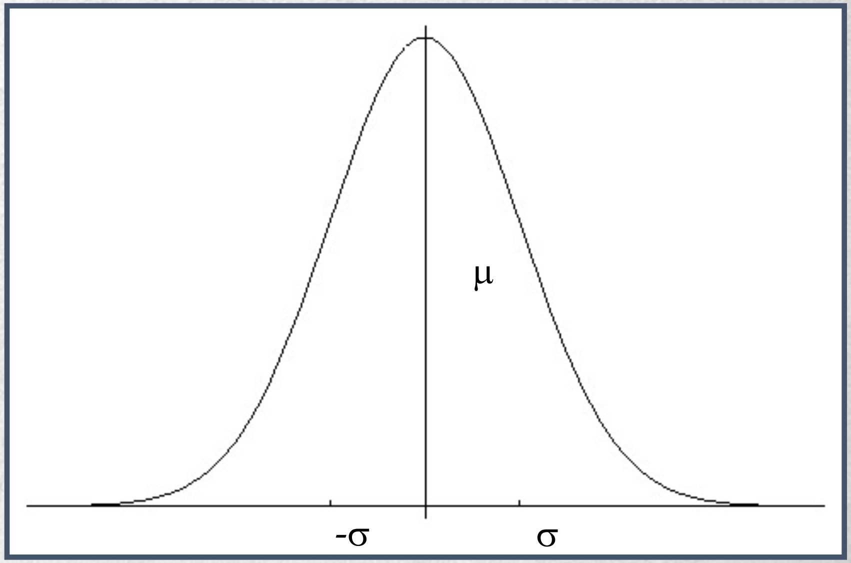
Kalman Filters

- Represent beliefs as Gaussian Distributions

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

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

Univariate



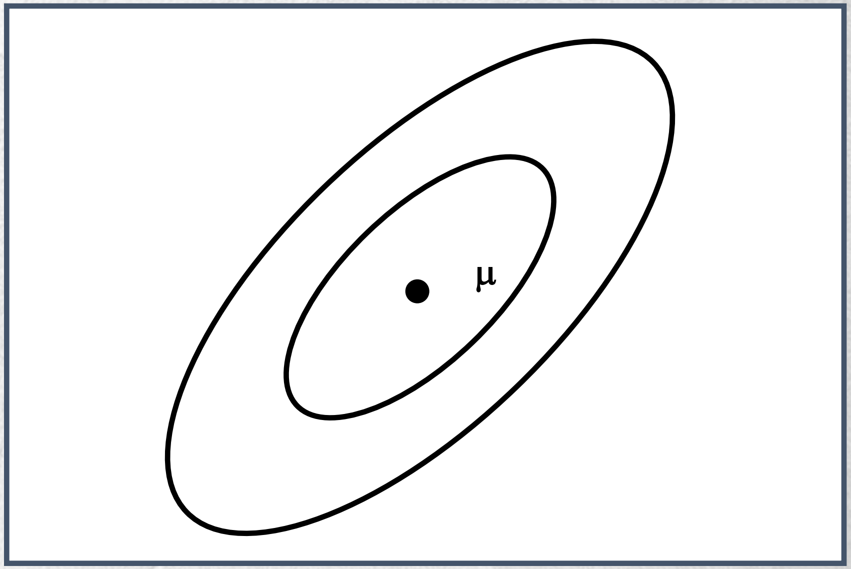
Kalman Filters

- Represent beliefs as Gaussian Distributions

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

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

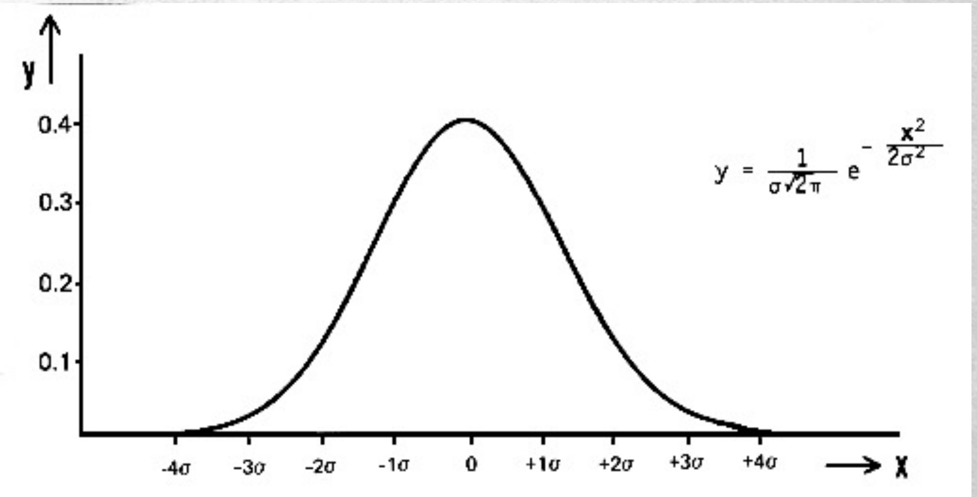
Multivariate



Gaussian Noise

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

$$z_t = C_t x_t + \delta_t \ .$$



Components of a Kalman Filter

A State Transition Matrix (n x n): describes how the state evolves from t to t-1 without controls or noise.

B Control Input Matrix (n x l): describes how the control u_t changes the state from t to t-1.

C Matrix (k x n) that describes how to map the state x_t to an observation z_t.

ε

δ

Random variables representing the motion and measurement noise that are assumed to be independent and normally distributed with covariance R and Q respectively.

Probabilistic State Estimation

$$P(x_t|u_1, z_1, u_2, z_2, \cdots, u_t, z_t) = P(x_t|z_{1:t}, u_{1:t})$$

Robot State

sense

Sensor Model

$$p(z_t \mid x_t) \quad = \quad \det(2\pi Q_t)^{-\frac{1}{2}} \; \exp\left\{-\tfrac{1}{2}(z_t - C_t\, x_t)^T \, Q_t^{-1} \, (z_t - C_t\, x_t)\right.$$

Motion Model

act

$$\begin{aligned} & p(x_t \mid u_t, x_{t-1}) & (3.4) \\ & = \det(2\pi R_t)^{-\frac{1}{2}} \; \exp\left\{-\tfrac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right\} \end{aligned}$$

Multivariate Gaussians

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

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

- We stay in the "Gaussian world" as long as we start with Gaussians and perform only linear transformations.
- This means there is a closed form way to keep track of our beliefs

Derivation

Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

$$\boxed{bel(x_0) = N\left(x_0; \mu_0, \Sigma_0\right)}$$

Linear Gaussian Systems: Dynamics

- Dynamics are linear function of state and control plus additive noise:

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

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

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

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

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

Linear Gaussian Systems: Dynamics

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

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

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

$$\Downarrow$$

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

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

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

Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

$$\boxed{z_t = C_t x_t + \delta_t}$$

$$\boxed{p(z_t \mid x_t) = N\left(z_t ; C_t x_t , Q_t\right)}$$

$$\boxed{\begin{array}{cccc} bel(x_t) = & \eta & p(z_t \mid x_t) & \overline{bel}(x_t) \\ & & \Downarrow & \Downarrow \\ & & \sim N\left(z_t ; C_t x_t , Q_t\right) & \sim N\left(x_t ; \overline{\mu}_t , \overline{\Sigma}_t\right) \end{array}}$$

Linear Gaussian Systems: Observations

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

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

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

$$\Downarrow$$

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

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

Kalman Filter Equations

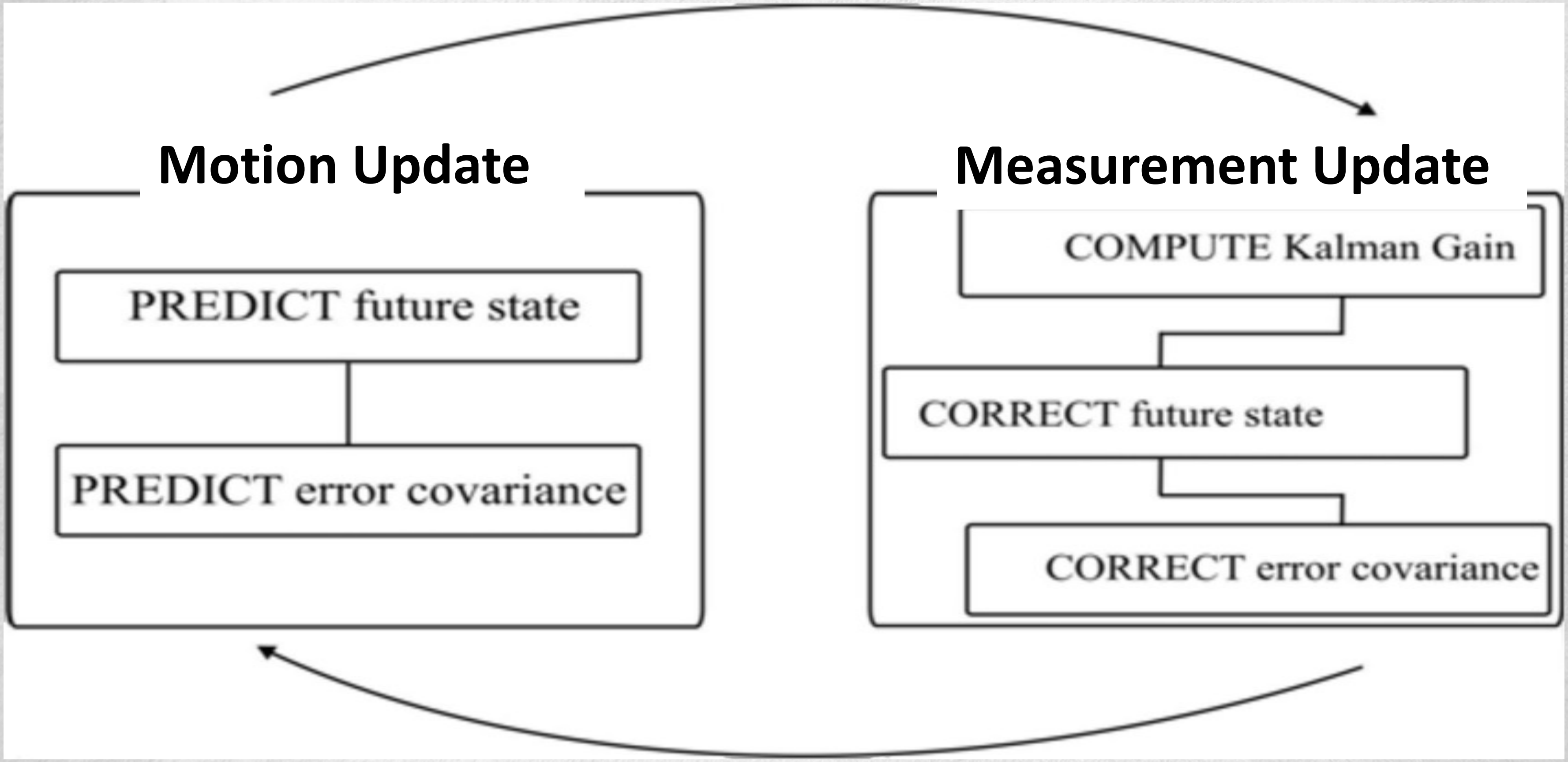
Kalman Filter Algorithm

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

1. Prediction (Motion update):
2. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
3. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

4. Correction (Sensor update):
5. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
6. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
7. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
8. Return μ_t, Σ_t

Kalman Filter Cycle

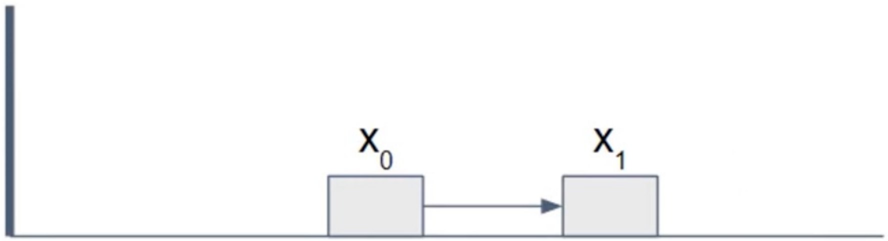


Simple Example

$A = 1$ $\quad C = 1$ $\quad \mu_0 = 5$ $\quad R = 0.1$

$B = 1$ $\quad \Sigma_0 = 0$ $\quad Q = 0.3$

- We will look at a 1D robot that can move along the x-axis.
- The robot has just 1 noisy sensor that gives us the distance to the wall located at x=0.



Simple Example

Motion Update

$$\bar{\mu}_1 = A\mu_0 + Bu_1$$

$$\bar{\mu}_1 = 1\,(5) + 1\,(2.5)$$

$$\boxed{\bar{\mu}_1 = 7.5}$$

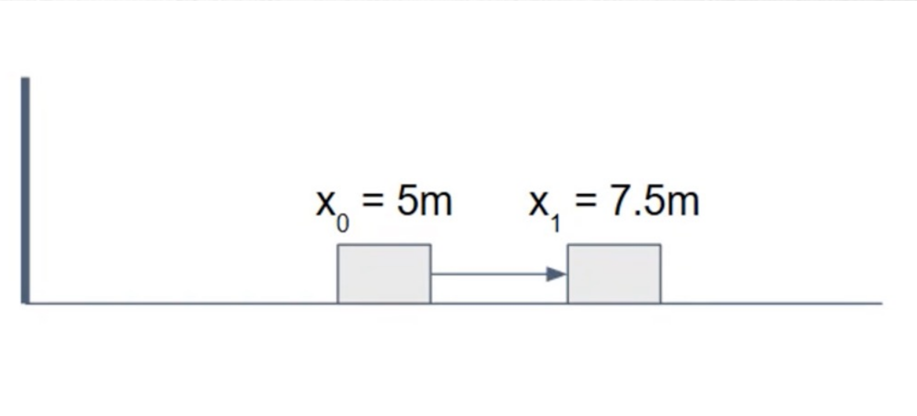
$$\bar{\Sigma}_1 = A\Sigma_0 A^T + R$$

$$\bar{\Sigma}_1 = 1(0)(1) + 0.1$$

$$\boxed{\bar{\Sigma}_1 = 0.1}$$

$$u_1 = +2.5$$

Input force causes movement to the right



Uncertainty increases

Simple Example

Measurement Update

$$K_1 = \bar{\Sigma}_1 C^T (C\bar{\Sigma}_1 C^T + Q)^{-1}$$

$$K_1 = 0.1(1)(1(0.1)(1) + 0.3)^{-1}$$

$$K_1 = 0.25$$

$$\Sigma_1 = (I - K_1 C)\bar{\Sigma}_1$$

$$\Sigma_1 = (1 - 0.25)\, 0.1$$

$$\boxed{\Sigma_1 = 0.075}$$

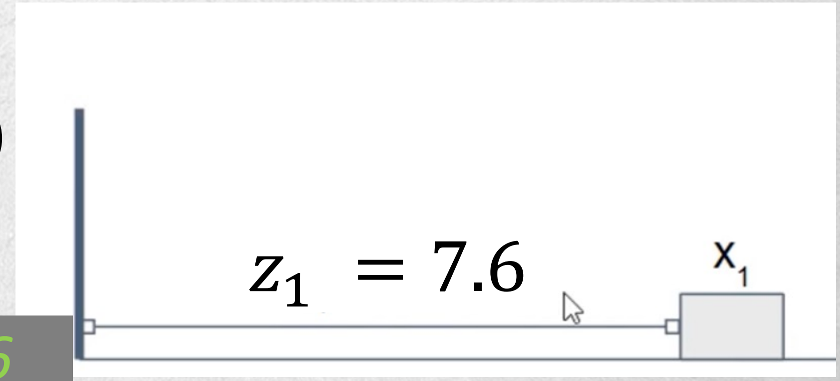
Uncertainty decreases

$$\mu_1 = \bar{\mu}_1 + K_1(z_1 - C_1\bar{\mu}_1)$$

$$\mu_1 = 7.5 + 0.25\,(7.6 - 1(7.5))$$

$$\boxed{\mu_1 = 7.525}$$

New estimate is in between 7.5 & 7.6



Simple Example

Motion Update

$$\bar{\mu}_2 = A\mu_1 + Bu_2$$

$$\bar{\mu}_2 = 1\,(7.525) + 1\,(2.5)$$

$$\boxed{\bar{\mu}_2 = 10.025}$$

$$\bar{\Sigma}_2 = A\Sigma_1 A^T + R$$

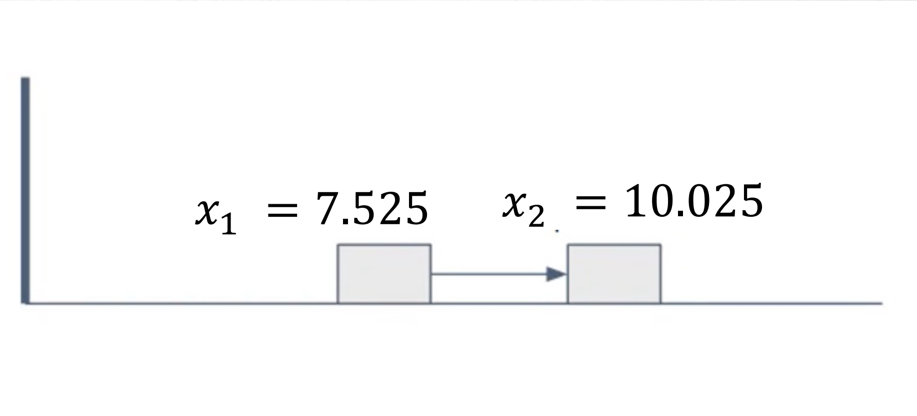
$$\bar{\Sigma}_2 = 1(0.075)(1) \; + 0.1$$

$$\boxed{\bar{\Sigma}_2 = 0.175}$$

Uncertainty increases

$$u_2 = +\,2.5$$

Input force causes movement to the right



Simple Example

Measurement Update

$$K_2 = \bar{\Sigma}_2 C^T (C\bar{\Sigma}_2 C^T + Q)^{-1}$$

$$K_2 = 0.175(0.175 + 0.3)^{-1}$$

$$K_2 = 0.368$$

$$\Sigma_2 = (I - K_2 C)\bar{\Sigma}_2$$

$$\Sigma_2 = (1 - 0.368)\, 0.175$$

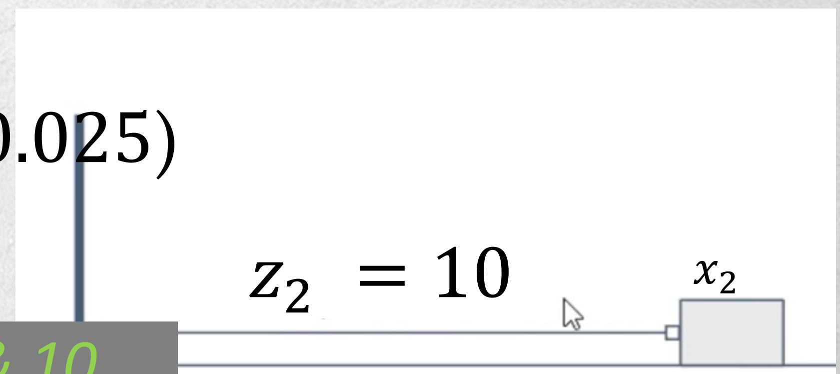
$$\boxed{\Sigma_1 = 0.111}$$

Uncertainty decreases

$$\mu_2 = \bar{\mu}_2 + K_2(z_2 - C\bar{\mu}_2)$$

$$\mu_2 = 10.025 + 0.368\,(10 - 10.025)$$

$$\boxed{\mu_1 = 10.016}$$



New estimate is in between 10.025 & 10

Analysis

Kalman Gain

Kalman Gain is a weighting between noise in the motion model vs noise in the sensor model

$$K_t = \bar{\Sigma}_t C^T (C \bar{\Sigma}_t C^T + Q_t)^{-1}$$

Case Study

1) Sensor model is perfect (Q = 0)

$$K_t = \bar{\Sigma}_t C^T (C\,\bar{\Sigma}_t C^T + Q_t)^{-1}$$

$$K_t = \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + 0\,)^{-1}$$

$$K_t = \bar{\Sigma}_t C^T (C^T)^{-1} (\bar{\Sigma}_t)^{-1}\,(C)^{-1}$$

$$K_t = C^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - C\,\bar{\mu}_t)$$

$$\mu_t = \bar{\mu}_t + C^{-1}(z_t - C\bar{\mu}_t)$$

$$\mu_t = \bar{\mu}_t + C^{-1}z_t - \bar{\mu}_t$$

$$\boxed{\mu_t = C^{-1}z_t}$$

State estimate only depends on sensor reading

Case Study

1) Sensor model is perfect (Q = 0)

$$K_t = \bar{\Sigma}_t C^T (C\,\bar{\Sigma}_t C^T + Q_t)^{-1}$$

$$K_t = \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + 0\,)^{-1}$$

$$K_t = \bar{\Sigma}_t C^T (C^T)^{-1} (\bar{\Sigma}_t)^{-1}\,(C)^{-1}$$

$$K_t = C^{-1}$$

$$\Sigma_t = (I - K_t C)\bar{\Sigma}_t$$

$$\Sigma_t = (I - C^{-1} C)\bar{\Sigma}_t$$

$$\boxed{\Sigma_t = 0}$$

State estimate has zero uncertainty

Case Study

2) Sensor model is garbage (Q = ∞)

$$K_t = \bar{\Sigma}_t C^T (C\,\bar{\Sigma}_t C^T + Q_t)^{-1}$$

$$K_t = \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + \infty\,)^{-1}$$

$$K_t = 0$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - C\,\bar{\mu}_t)$$

$$\mu_t = \bar{\mu}_t + 0\,(z_t - C\,\bar{\mu}_t)$$

$$\boxed{\mu_t = \bar{\mu}_t}$$

State estimate only depends on motion update

Case Study

2) Sensor model is garbage ($Q = \infty$)

$$K_t = \bar{\Sigma}_t C^T (C \bar{\Sigma}_t C^T + Q_t)^{-1}$$

$$K_t = \bar{\Sigma}_t C^T (C \bar{\Sigma}_t C^T + \infty)^{-1}$$

$$K_t = 0$$

$$\Sigma_t = (I - K_t C)\bar{\Sigma}_t$$

$$\Sigma_t = (I - 0\, C)\bar{\Sigma}_t$$

$$\boxed{\Sigma_t = \bar{\Sigma}_t}$$

Uncertainty in state estimate is unchanged after measurement update

Case Study

3) Motion model is perfect & initial uncertainty is 0 (R = 0, $\Sigma_{t-1} = 0$)

$$\bar{\Sigma}_t = A\Sigma_{t-1}A^T + R$$

$$\bar{\Sigma}_t = 0$$

$$K_t = \bar{\Sigma}_t C^T (C\,\bar{\Sigma}_t C^T + Q_t)^{-1}$$

$$K_t = 0$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - C\,\bar{\mu}_t)$$

$$\mu_t = \bar{\mu}_t + 0\,(z_t - C\,\bar{\mu}_t)$$

$$\boxed{\mu_t = \bar{\mu}_t}$$

State estimate only depends on motion update

Case Study

4) Motion model is garbage (R = ∞)

$$\bar{\Sigma}_t = A\Sigma_{t-1}A^T + R$$

$$\bar{\Sigma}_t = \infty$$

$$K_t = \bar{\Sigma}_t C^T (C\,\bar{\Sigma}_t C^T + Q_t)^{-1}$$

$$K_t = C^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - C\,\bar{\mu}_t)$$

$$\mu_t = \bar{\mu}_t + C^{-1}(z_t - C\,\bar{\mu}_t)$$

$$\mu_t = \bar{\mu}_t + C^{-1}z_t - \bar{\mu}_t$$

$$\boxed{\mu_t = C^{-1}z_t}$$

State estimate only depends on sensor reading

Kalman Filter Summary

- Highly efficient: Polynomial in measurement dimensionality k and state dimensionality n:
  $$O(k^{2.376} + n^2)$$
- Optimal for linear Gaussian systems!
- Usually used for Tracking Problem, and with landmarks in the environment
- Problem:
  - Most robotics systems are nonlinear!

References

1. http://bilgin.esme.org/BitsAndBytes/KalmanFilterforDummies
2. https://www.kalmanfilter.net/default.aspx
3. https://en.wikipedia.org/wiki/Kalman_filter