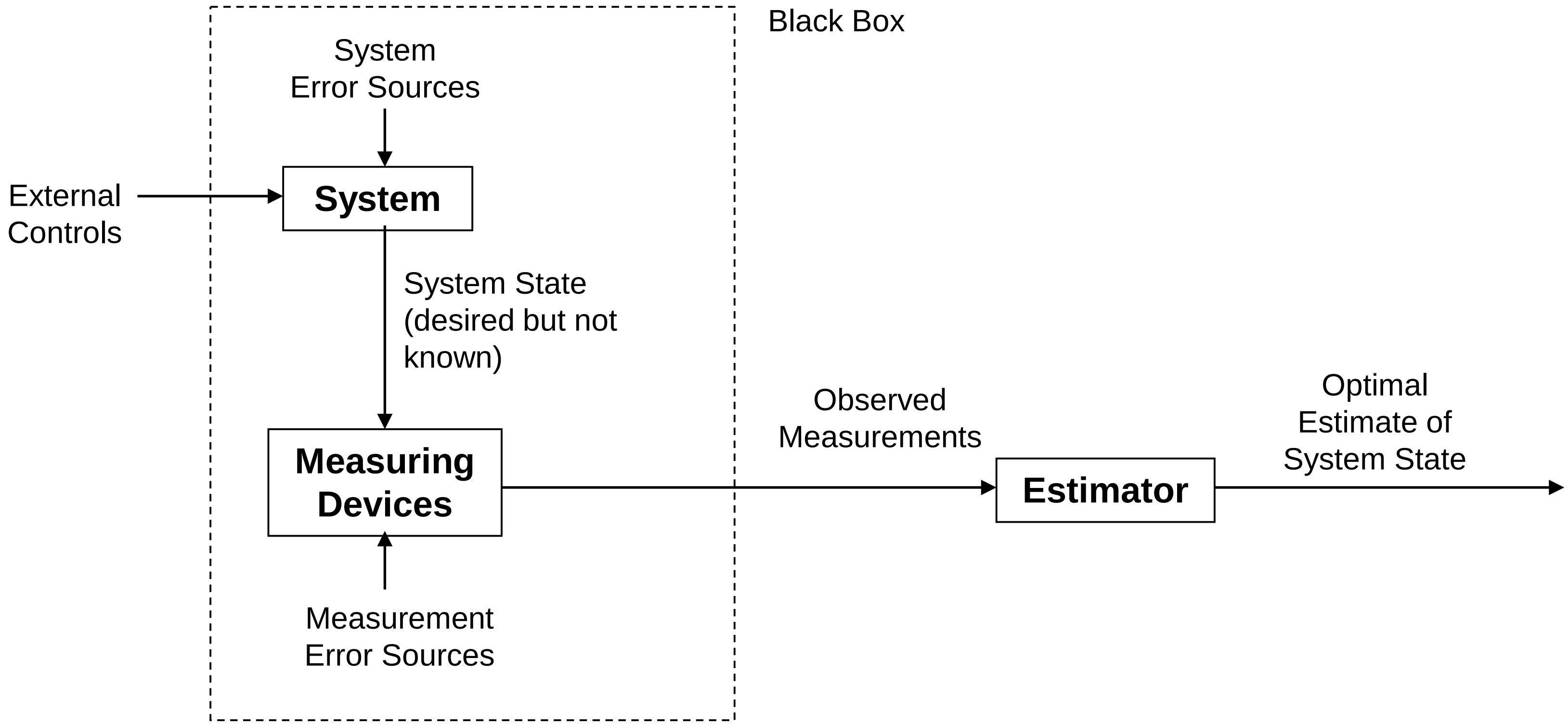
Introduction to Kalman Filters

2018 응용로봇공학

Overview

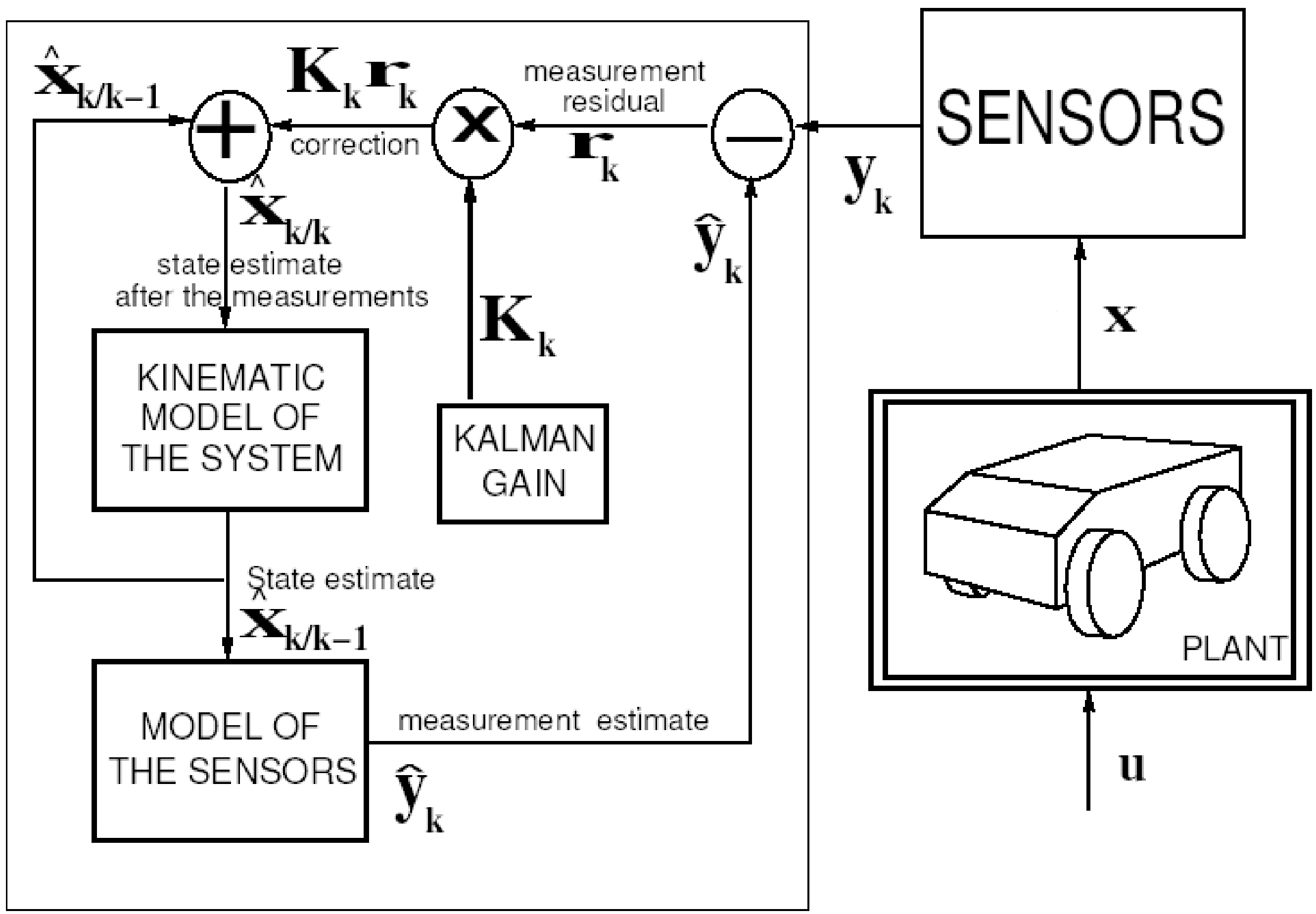
- ✓ The Problem – Why do we need Kalman Filters?
- ✓ What is a Kalman Filter?
- ✓ Conceptual Overview
- ✓ The Theory of Kalman Filter
- ✓ Simple Example

The Problem



- System state cannot be measured directly
- Need to estimate "optimally" from measurements

Kalman Filter Block Diagram



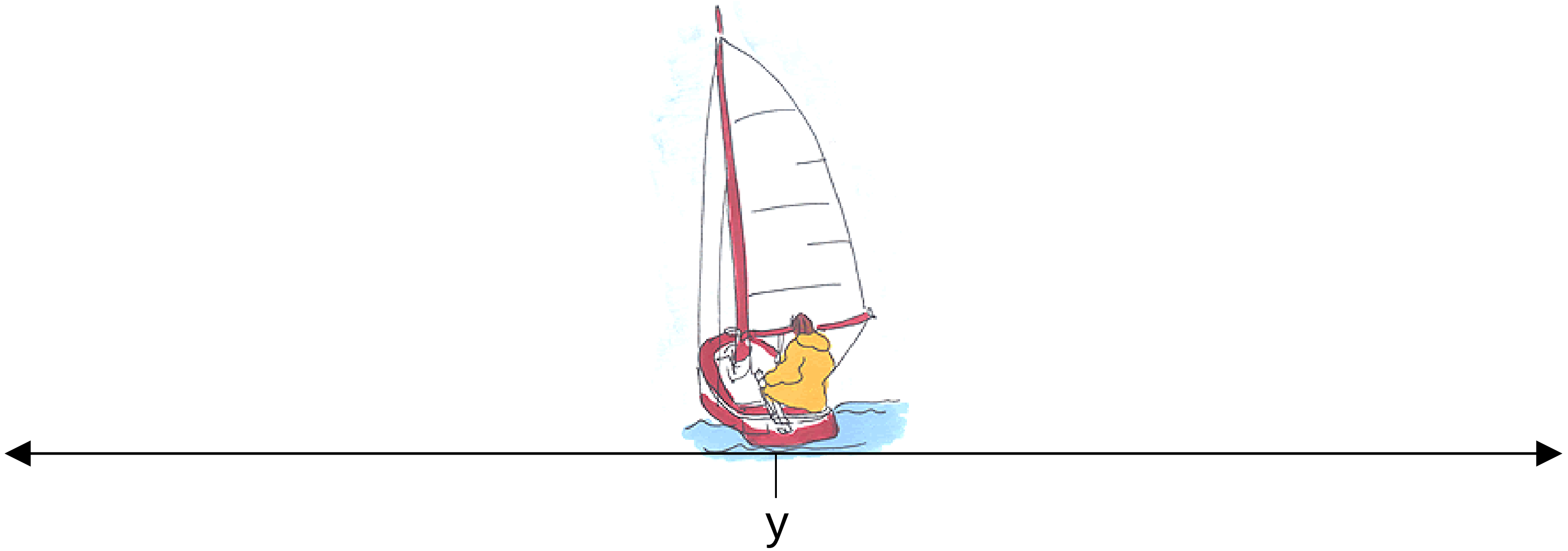
What is a Kalman Filter?

- Recursive data processing algorithm
- Generates optimal estimate of desired quantities given the set of measurements
- Optimal?
  - For linear system and white Gaussian errors, Kalman filter is “best” estimate based on all previous measurements
  - For non-linear system optimality is ‘qualified’
- Recursive?
  - Doesn’t need to store all previous measurements and reprocess all data each time step

Conceptual Overview

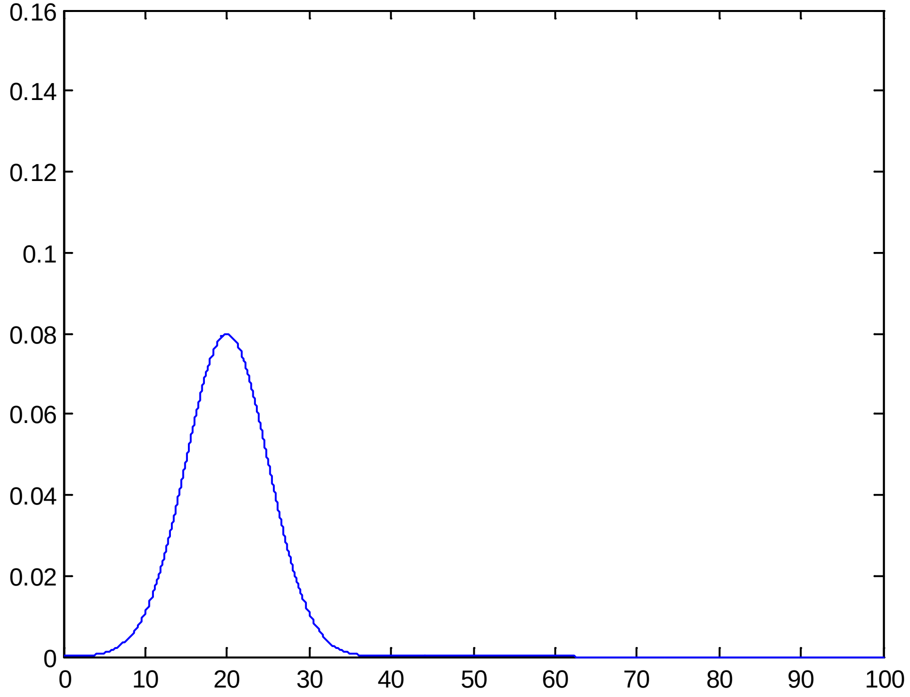
- Simple example to motivate the workings of the Kalman Filter
- Theoretical Justification to come later
  - for now just focus on the concept
- Important: Prediction and Correction

Conceptual Overview



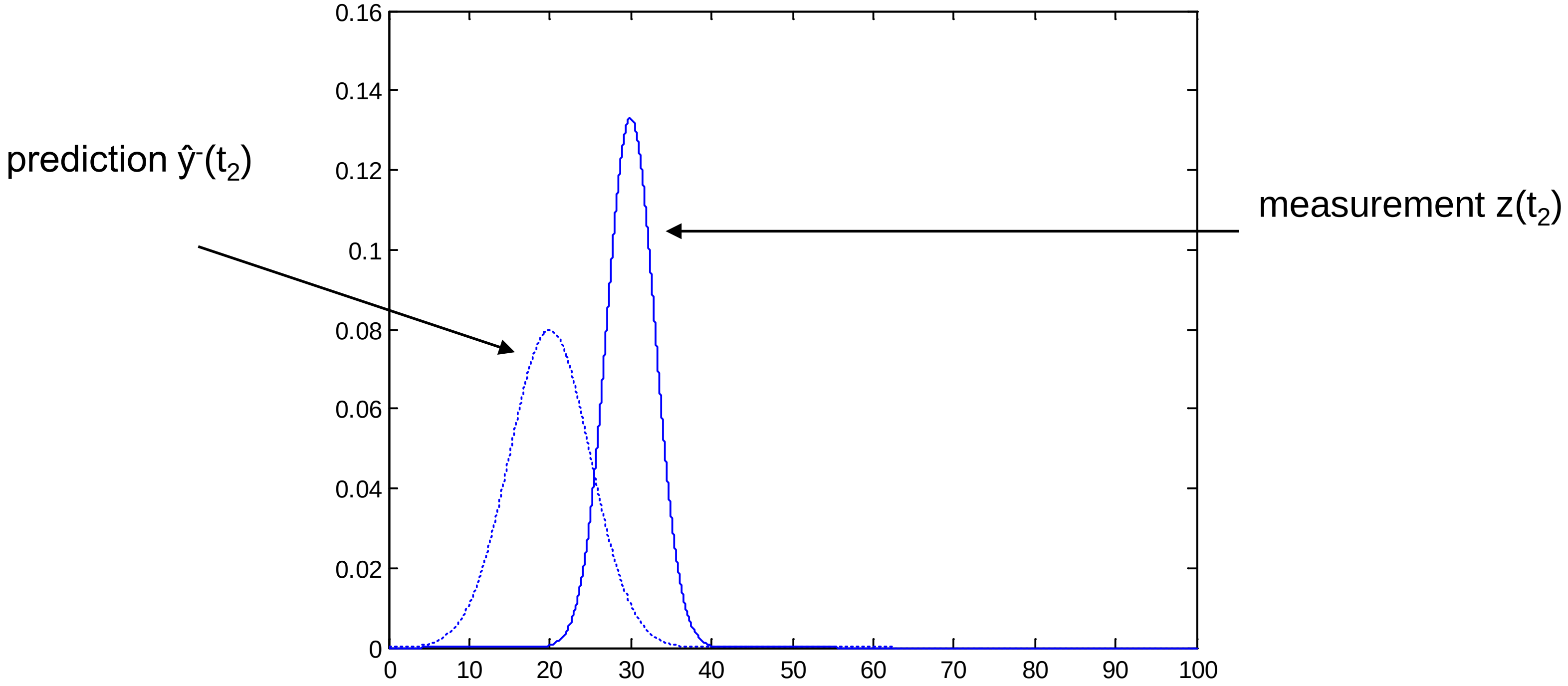
- Lost on the 1-dimensional line
- Position – y(t)
- Assume Gaussian distributed measurements

Conceptual Overview



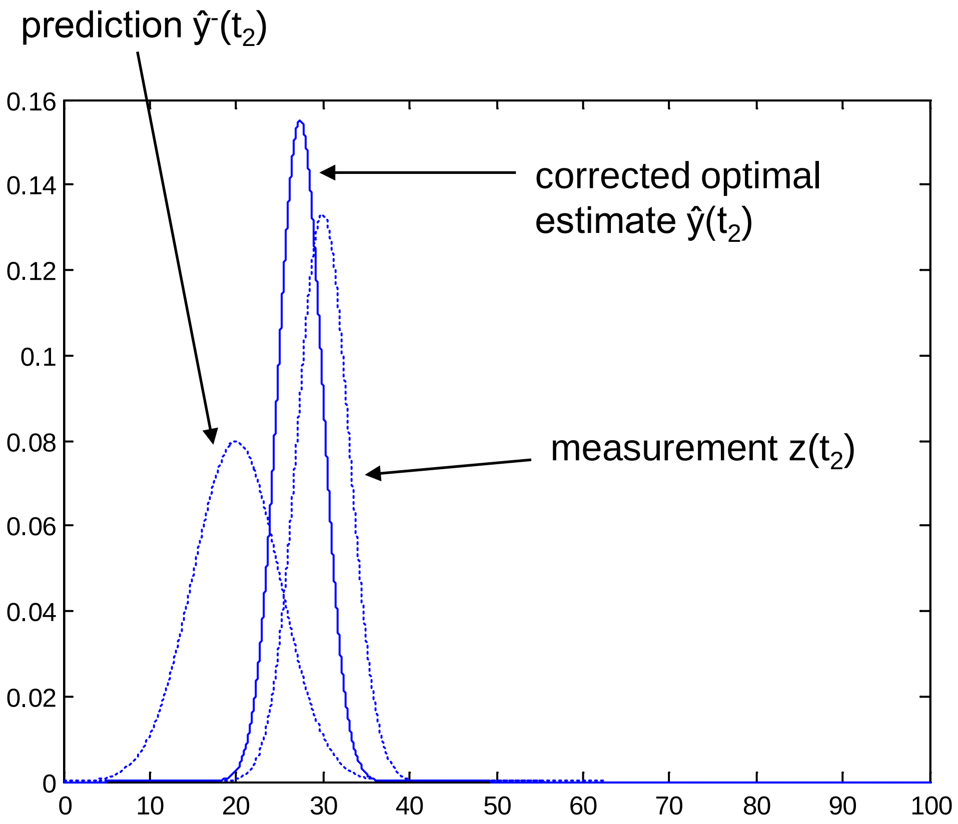
- Sextant Measurement at t_1: Mean = z_1 and Variance = σ_{z1}
- Optimal estimate of position is: $\hat{y}(t_1) = z_1$
- Variance of error in estimate: $\sigma^2_x(t_1) = \sigma^2_{z1}$
- Boat in same position at time t_2 - <u>Predicted</u> position is z_1

Conceptual Overview



- So we have the prediction $\hat{y}^{-}(t_2)$
- GPS Measurement at t_2: Mean = z_2 and Variance = σ_{z2}
- Need to <u>correct</u> the prediction due to measurement to get $\hat{y}(t_2)$
- Closer to more trusted measurement – linear interpolation?

Conceptual Overview



- Corrected mean is the new optimal estimate of position
- New variance is smaller than either of the previous two variances

Conceptual Overview

- Lessons so far:

Make prediction based on previous data - $\hat{y}^-$, σ^-

↓

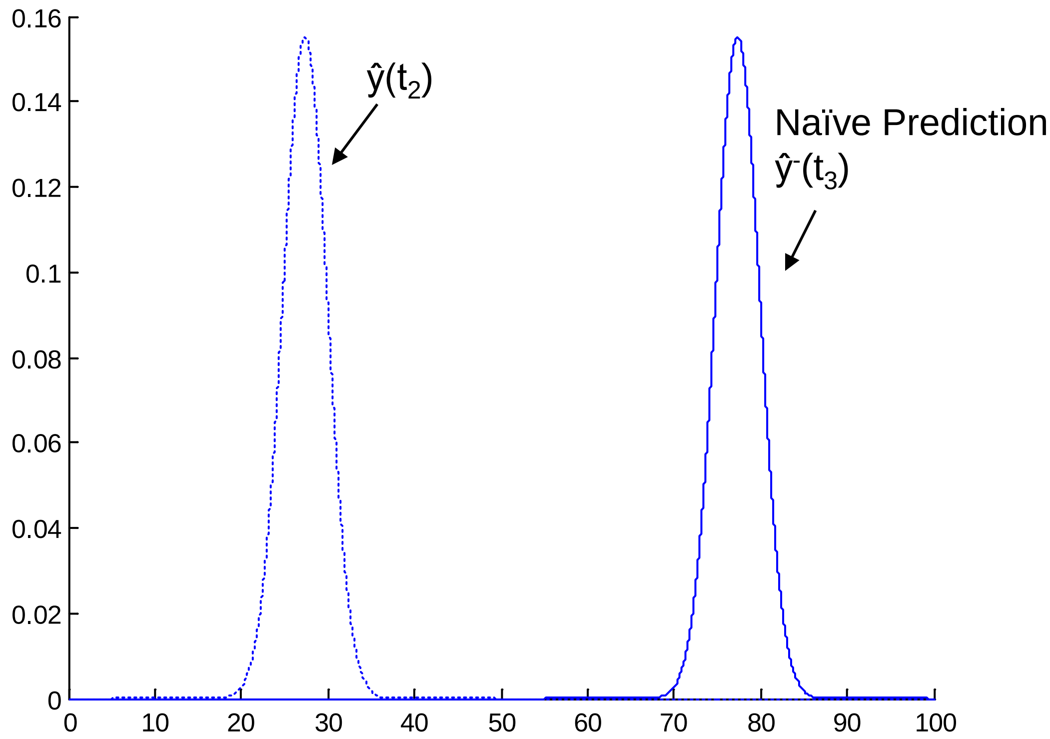
Take measurement – z_k, σ_z

↓

Optimal estimate ($\hat{y}$) = Prediction + (Kalman Gain) * (Measurement - Prediction)

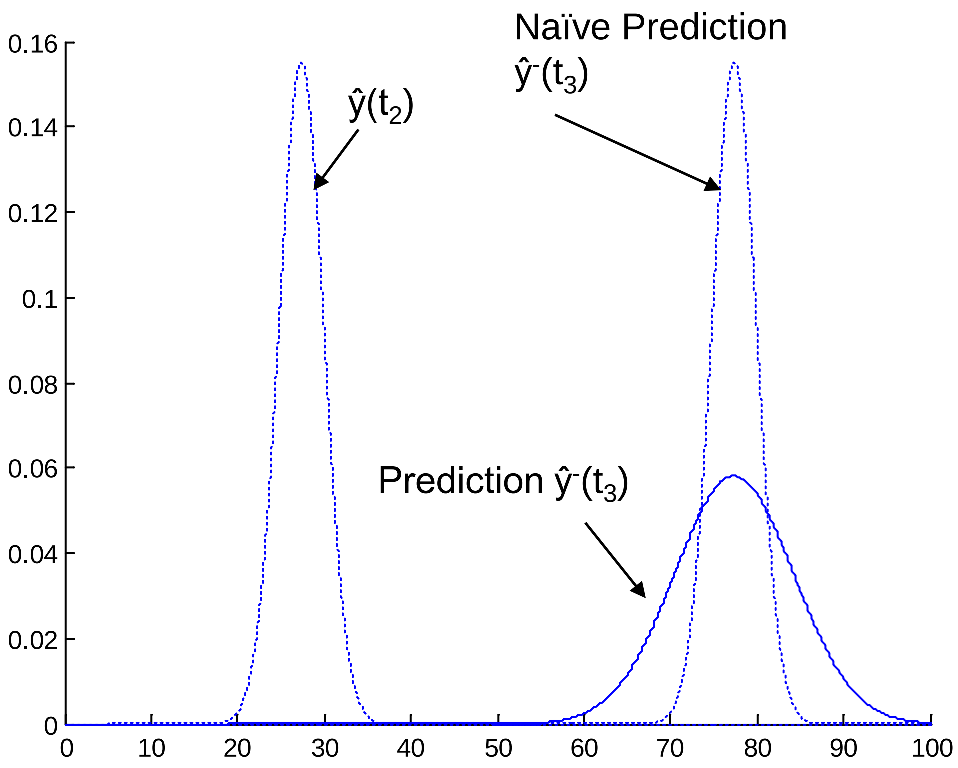
Variance of estimate = Variance of prediction * (1 – Kalman Gain)

Conceptual Overview



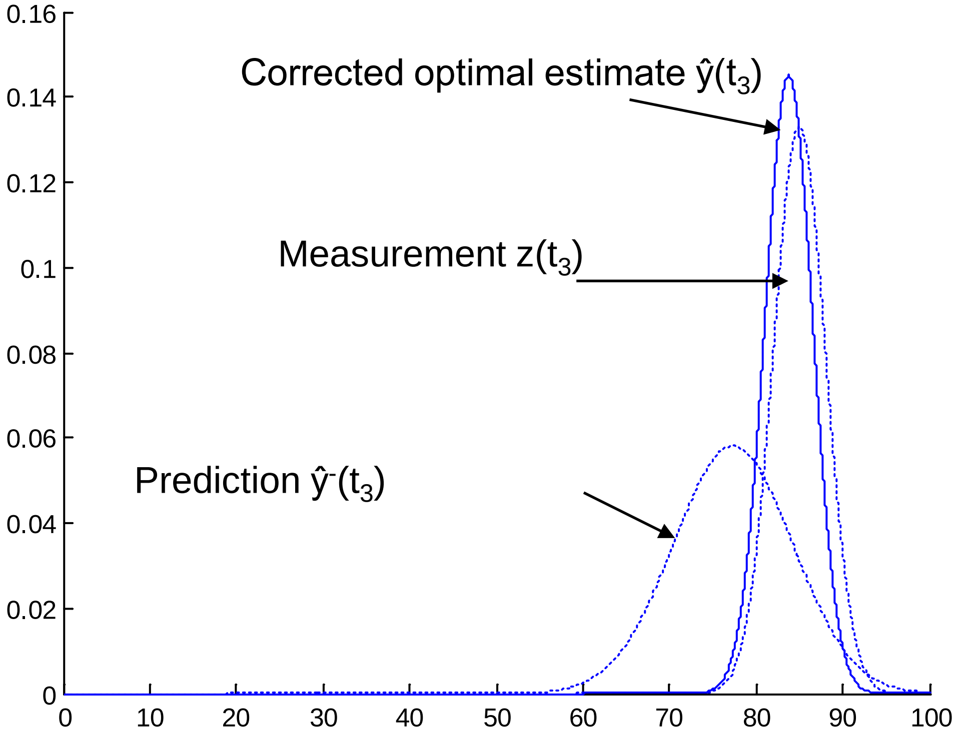
- At time t_3, boat moves with velocity dy/dt=u
- Naïve approach: Shift probability to the right to predict
- This would work if we knew the velocity exactly (perfect model)

Conceptual Overview



- Better to assume imperfect model by adding Gaussian noise
- dy/dt = u + w
- Distribution for prediction moves and spreads out

Conceptual Overview



- Now we take a measurement at t_3
- Need to once again correct the prediction
- Same as before

Conceptual Overview

- Lessons learnt from conceptual overview:
  - Initial conditions ($\hat{y}_{k-1}$ and σ_{k-1})
  - Prediction ($\hat{y}^-_k$, σ^-_k)
    - Use initial conditions and model (eg. constant velocity) to make prediction
  - Measurement (z_k)
    - Take measurement
  - Correction ($\hat{y}_k$, σ_k)
    - Use measurement to correct prediction by ‘blending’ prediction and residual – always a case of merging only two Gaussians
    - Optimal estimate with smaller variance

Kalman Filter Components

Linear discrete time dynamic system (motion model)

State Control input Process noise

$$x_{t+1} = F_t x_t + B_t u_t + G_t w_t$$

State transition function Control input function Noise input function with covariance Q

Measurement equation (sensor model)

Sensor reading State Sensor noise with covariance R

$$z_{t+1} = H_{t+1} x_{t+1} + v_{t+1}$$

Sensor function

Note:Write these down!!!

Theoretical Basis

- Process to be estimated:

$y_k = Ay_{k-1} + Bu_k + w_{k-1}$ Process Noise (w) with covariance Q

$z_k = Hy_k + v_k$ Measurement Noise (v) with covariance R

- Kalman Filter

Predicted: $\hat{y}^-_k$ is estimate based on measurements at previous time-steps

$$\hat{y}^-_k = Ay_{k-1} + Bu_k$$

$$P^-_k = AP_{k-1}A^T + Q$$

Corrected: $\hat{y}_k$ has additional information – the measurement at time k

$$\hat{y}_k = \hat{y}^-_k + K(z_k - H\,\hat{y}^-_k)$$

$$K = P^-_kH^T(HP^-_kH^T + R)^{-1}$$

$$P_k = (I - KH)P^-_k$$

Theoretical Basis

Prediction (Time Update)

(1) Project the state ahead

$$\hat{y}^{-}_{k} = Ay_{k-1} + Bu_{k}$$

(2) Project the error covariance ahead

$$P^{-}_{k} = AP_{k-1}A^{T} + Q$$

Correction (Measurement Update)

(1) Compute the Kalman Gain

$$K = P^{-}_{k}H^{T}(HP^{-}_{k}H^{T} + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{y}_{k} = \hat{y}^{-}_{k} + K(z_{k} - H\,\hat{y}^{-}_{k})$$

(3) Update Error Covariance

$$P_{k} = (I - KH)P^{-}_{k}$$

Blending Factor

- If we are sure about measurements:
  - Measurement error covariance (R) decreases to zero
  - K decreases and weights residual more heavily than prediction

- If we are sure about prediction
  - Prediction error covariance P^-_k decreases to zero
  - K increases and weights prediction more heavily than residual

$\hat{y}^-_k = Ay_{k-1} + Bu_k$

$P^-_k = AP_{k-1}A^T + Q$

$K = P^-_k H^T (HP^-_k H^T + R)^{-1}$

$\hat{y}_k = \hat{y}^-_k + K(z_k - H\,\hat{y}^-_k)$

Kalman Filter 유도과정

Kalman Filter의 기본식

$$x_{k+1} = A_k x_k + w_k \qquad \text{(system equation)}$$

$$z_k = H_k x_k + v_k \qquad \text{(observation equation)}$$

x_k = (n×1) process state vector at time t_k (최적화를 하고자하는 상태변수)

A_k = (n×n) matrix relating x_k to x_{k+1} (한 단계에서의 상태변수와 다음 단계의 상태변수를 연결하는 시스템 매트릭스)

w_k = (n×1) vector (process noise)

z_k = (m×1) vector measurement at time t_k (관측값)

H_k = (m×n) matrix giving the ideal connection between the measurement and the state vector at time t_k

v_k = (m×1) measurement error (measurement noise)

Kalman Filter 유도과정

Prediction error

$$e_k^- = x_k - \hat{x}_k^-$$

Prediction error covariance

$$P_k^- = E[e_k^- e_k^{-T}]$$

Estimation error

$$e_k = x_k - \hat{x}_k$$

Estimation error covariance

$$P_k = E[e_k e_k^T]$$

Noise terms w_k v_k

$$E[w_k w_i^T] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases}$$

$$E[v_k v_i^T] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases}$$

$E[w_k v_i^T] = 0,$ for all k and i

상관없는 양들

Kalman Filter 유도과정-

$$P_k \leftarrow P_k^-, K_k$$

$e_k = x_k - \hat{x}_k$

$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-)$

$($ $z_k = H_k x_k + v_k$ 대입 $)$

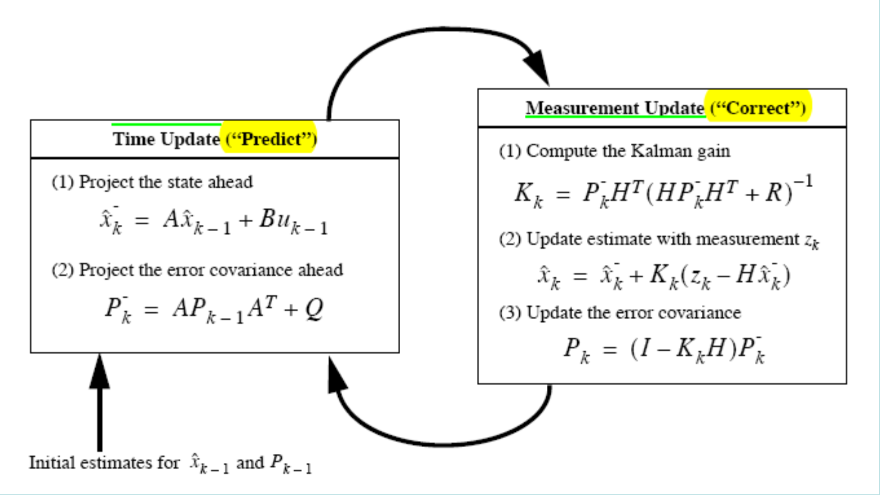
$$
\begin{aligned}
P_k &= E[e_k e_k^T] = E\{[x_k - \hat{x}_k^- - K_k(H_k x_k + v_k - H_k\hat{x}_k^-)][x_k - \hat{x}_k^- - K_k(H_k x_k + v_k - H_k\hat{x}_k^-]^T\} \\
&= E\{[(x_k - \hat{x}_k^-) - K_k(H_k x_k + v_k - H_k\hat{x}_k^-)][(x_k - \hat{x}_k^-) - K_k(H_k x_k + v_k - H_k\hat{x}_k^-)]^T\} \\
&= E\{[e_k^- - K_k(H_k(x_k - \hat{x}_k^-) + v_k)][e_k^- - K_k(H_k(x_k - \hat{x}_k^-) + v_k)]^T\}
\end{aligned}
$$

$\boxed{e_k^- = x_k - \hat{x}^-}$

$$
\begin{aligned}
&= E\{[e_k^- - K_k(H_k e_k^- + v_k)][e_k^- - K_k(H_k e_k^- + v_k)]^T\} \\
&= E\{[e_k^- - K_k H_k e_k^- - K_k v_k][e_k^- - K_k H_k e_k^- - K_k v_k]^T\} \\
&= E\{[e_k^- - K_k H_k e_k^- - K_k v_k][e_k^{-T} - e_k^{-T} H_k^T K_k^T - v_k^T K_k^T]\} \\
&= E\{[e_k^- e_k^{-T} - e_k^- e_k^{-T} H_k^T K_k^T - e_k^- v_k^T K_k^T - K_k H_k e_k^- e_k^{-T} + K_k H_k e_k^- e_k^{-T} H_k^T K_k^T \\
&\quad + K_k H_k e_k^- v_k^T K_k^T - K_k v_k e_k^{-T} + K_k v_k e_k^{-T} H_k^T K_k^T + K_k v_k v_k^T K_k^T\} \\
&= E[e_k^- e_k^{-T}] - E[e_k^- e_k^{-T}] H_k^T K_k^T - \cancel{E[e_k^- v_k^T] K_k^T} - K_k H_k E[e_k^- e_k^{-T}] + K_k H_k E[e_k^- e_k^{-T}] H_k^T K_k^T \\
&\quad + \cancel{K_k H_k E[e_k^- v_k^T] K_k^T} - \cancel{K_k E[v_k e_k^{-T}]} + \cancel{K_k E[v_k e_k^{-T}] H_k^T K_k^T} + K_k E[v_k v_k^T] K_k^T \\
&= P_k^- - P_k^- H_k^T K_k^T - K_k H_k P_k^- + K_k H_k P_k^- H_k^T K_k^T + K_k R_k K_k^T \\
&= \boxed{P_k^- - P_k^- H_k^T K_k^T - K_k H_k P_k^- + K_k (H_k P_k^- H_k^T + R_k) K_k^T}
\end{aligned}
$$

$P_k^- = E[e_k^- e_k^{-T}] = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T]$

Kalman Filter 유도과정- $K_k \leftarrow P_k^-$



matrix differentiation formula

$$\frac{d[trace(ACA^T)]}{dA} = 2AC$$

(C must be symmetric)

$$\frac{d[trace(AB)]}{dA} = B^T$$

(AB must be square)

$$P_k = P_k^- - P_k^- H_k^T K_k^T - K_k H_k P_k^- + K_k (H_k P_k^- H_k^T + R_k) K_k^T$$

$$\begin{aligned} \frac{d(trace\, P_k)}{dK_k} &= -2(H_k P_k^-)^T + 2K_k(H_k P_k^- H_k^T + R_k) \\ &= -2P_k^{-T} H_k^T + 2K_k(H_k P_k^- H_k^T + R_k) \\ &= 0 \end{aligned}$$

- ✓ Trace(P_k) 는 고려된 모든 상태벡터들의 mean-square error들의 합에 해당
- ✓ 결국 이 합이 최적화될 때 각 오차들도 최적화 된다고 할 수 있다

$$K_k = P_k^{-T} H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$P_k^- = P_k^{-T}$$

(대칭행렬이기때문에)

$$\boxed{K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}}$$

Kalman Gain

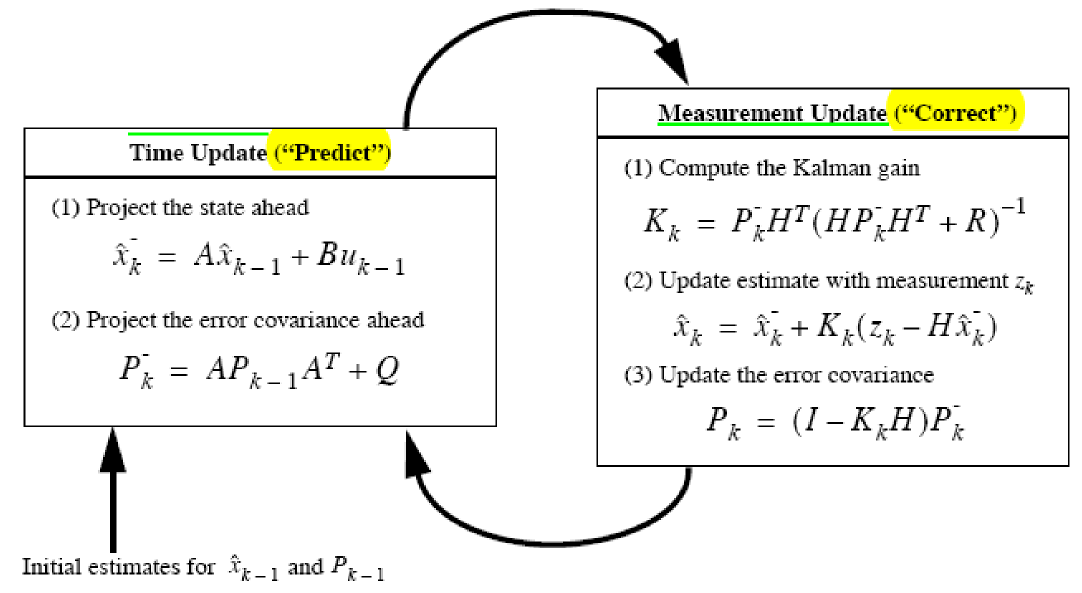
Kalman Filter 유도과정- $P_k \leftarrow P_k^-$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

대입

- Simplest update equation

$$\begin{aligned}
P_k &= P_k^- - P_k^- H_k^T K_k^T - K_k H_k P_k^- + K_k (H_k P_k^- H_k^T + R_k) K_k^T \\
&= P_k^- - P_k^- H_k^T K_k^T - K_k H_k P_k^- + P_k^- H_k^T \cancel{(H_k P_k^- H_k^T + R_k)^{-1}} \cancel{(H_k P_k^- H_k^T + R_k)} K_k^T \\
&= P_k^- - \cancel{P_k^- H_k^T K_k^T} - K_k H_k P_k^- + \cancel{P_k^- H_k^T K_k^T} \\
&= P_k^- - K_k H_k P_k^- \\
&= \boxed{(I - K_k H_k) P_k^-}
\end{aligned}$$

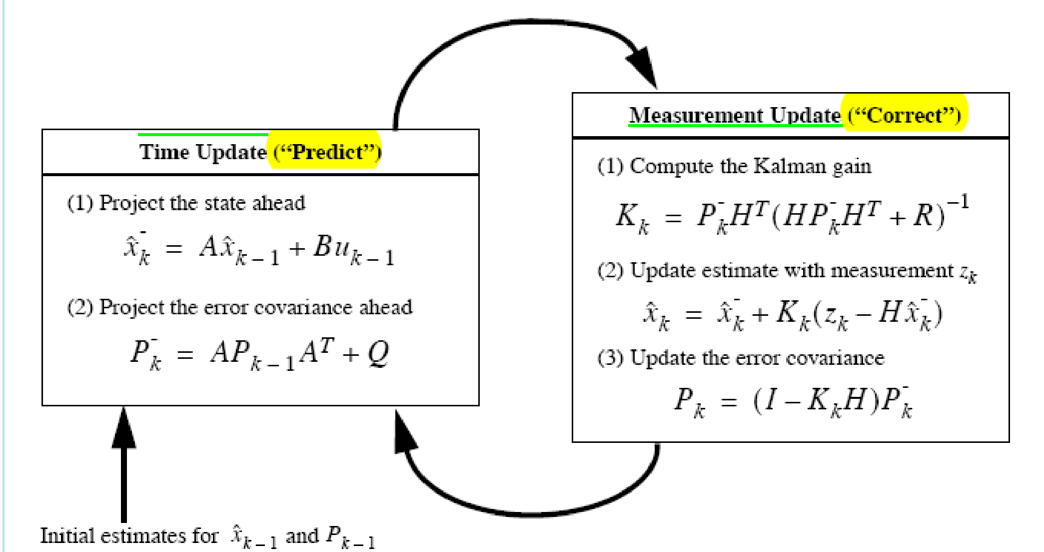


Kalman Filter 유도과정- $P_{k+1}^{-} \leftarrow P_k$

$$\hat{x}_{k+1}^{-} = A_k \hat{x}_k$$

$$
\begin{aligned}
e_{k+1}^{-} &= x_{k+1} - \hat{x}_{k+1}^{-} \\
&= (A_k x_k + w_k) - A_k \hat{x}_k \\
&= A_k (x_k - \hat{x}_k) + w_k \\
&= A_k e_k + w_k
\end{aligned}
$$

$$
\begin{aligned}
P_{k+1}^{-} &= E[e_{k+1}^{-} e_{k+1}^{-T}] = E[(A_k e_k + w_k)(A_k e_k + w_k)^T] \\
&= E[(A_k e_k + w_k)(e_k^T A_k^T + w_k^T)] \\
&= E[A_k e_k e_k^T A_k^T + A_k e_k w_k^T + w_k e_k^T A_k^T + w_k w_k^T] \\
&= A_k E[e_k e_k^T] A_k^T + A_k E[\cancel{e_k w_k^T}] + E[\cancel{w_k e_k^T}] A_k^T + E[w_k w_k^T] \\
&= \boxed{A_k P_k A_k^T + Q_k}
\end{aligned}
$$



그래서....

Step 1: Build a model

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

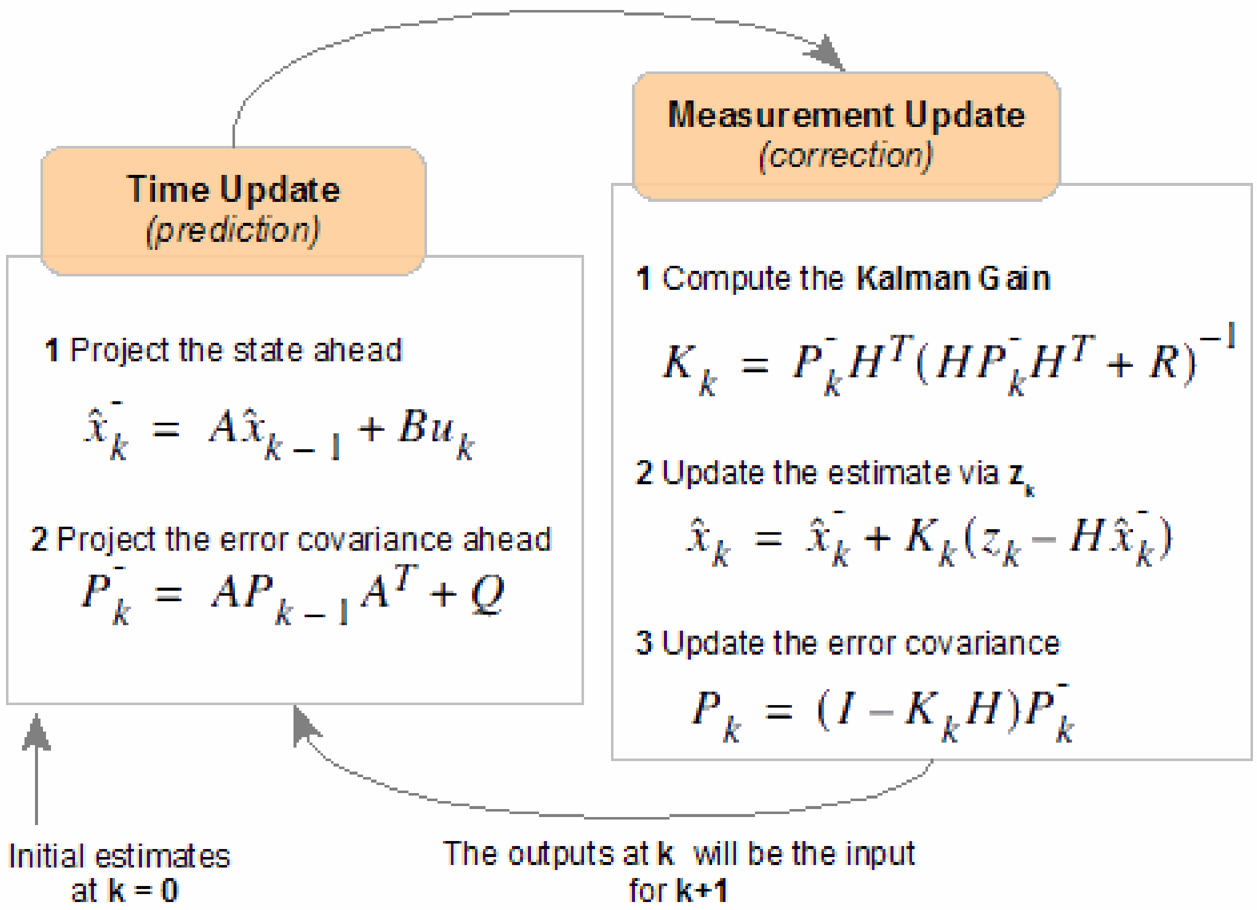
$$z_k = Hx_k + v_k$$

- Any $\mathbf{x_k}$ is a linear combination of its previous value plus a control signal $\mathbf{u_k}$ and a **process noise**.
- The entities **A**, **B** and **H** are in general matrices related to the states. In many cases, we can assume they are numeric value and constant.
- $\mathbf{W_{k-1}}$ is the **process noise** and $\mathbf{v_k}$ is the **measurement noise,** both are considered to be Gaussian.

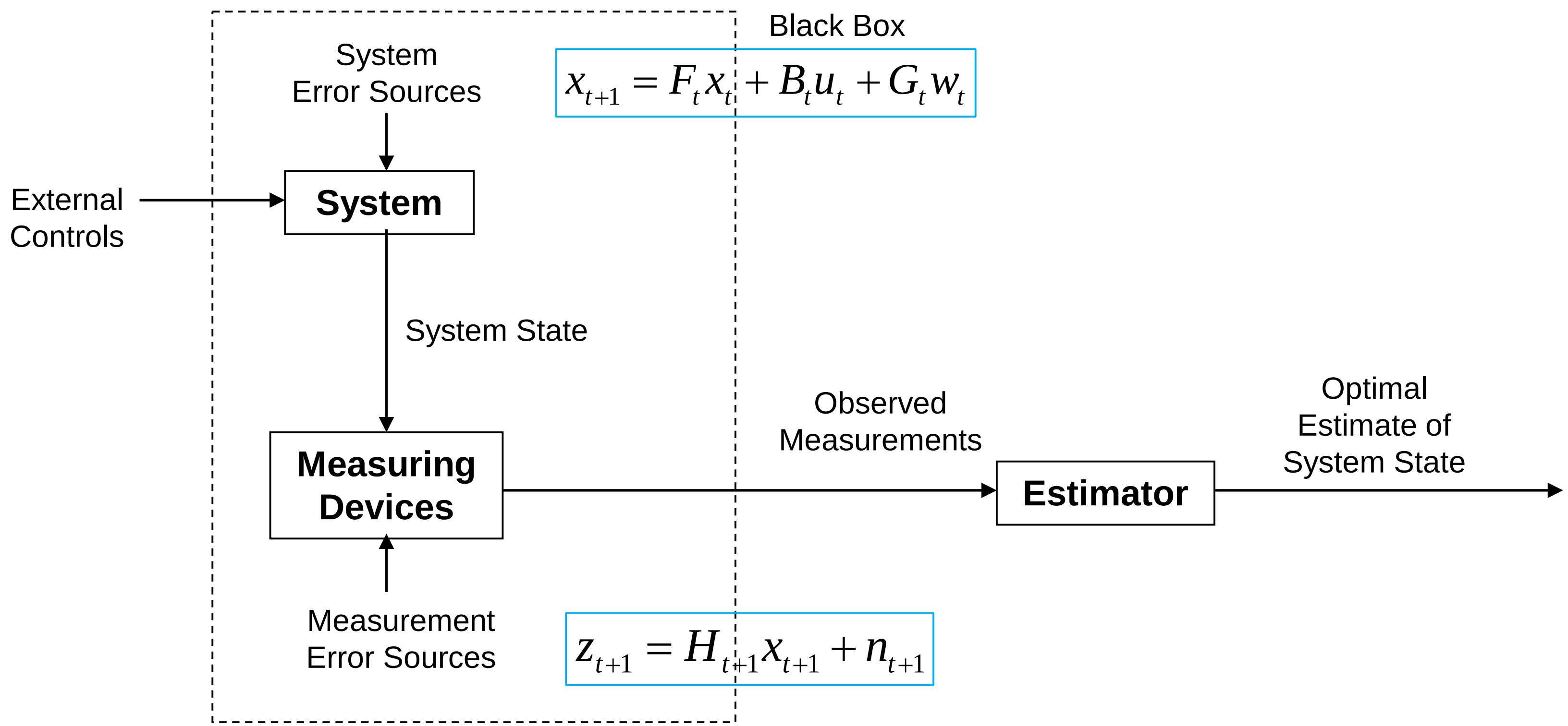
Step 2: Start process

Time Update *(prediction)*	Measurement Update *(correction)*
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$	$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
$P_k^- = AP_{k-1}A^T + Q$	$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
	$P_k = (I - K_k H)P_k^-$

Step 3: Iterate



Quick Example – Constant Model



Quick Example – Constant Model

Prediction

$$\hat{y}^-_k = y_{k-1}$$

$$P^-_k = P_{k-1}$$

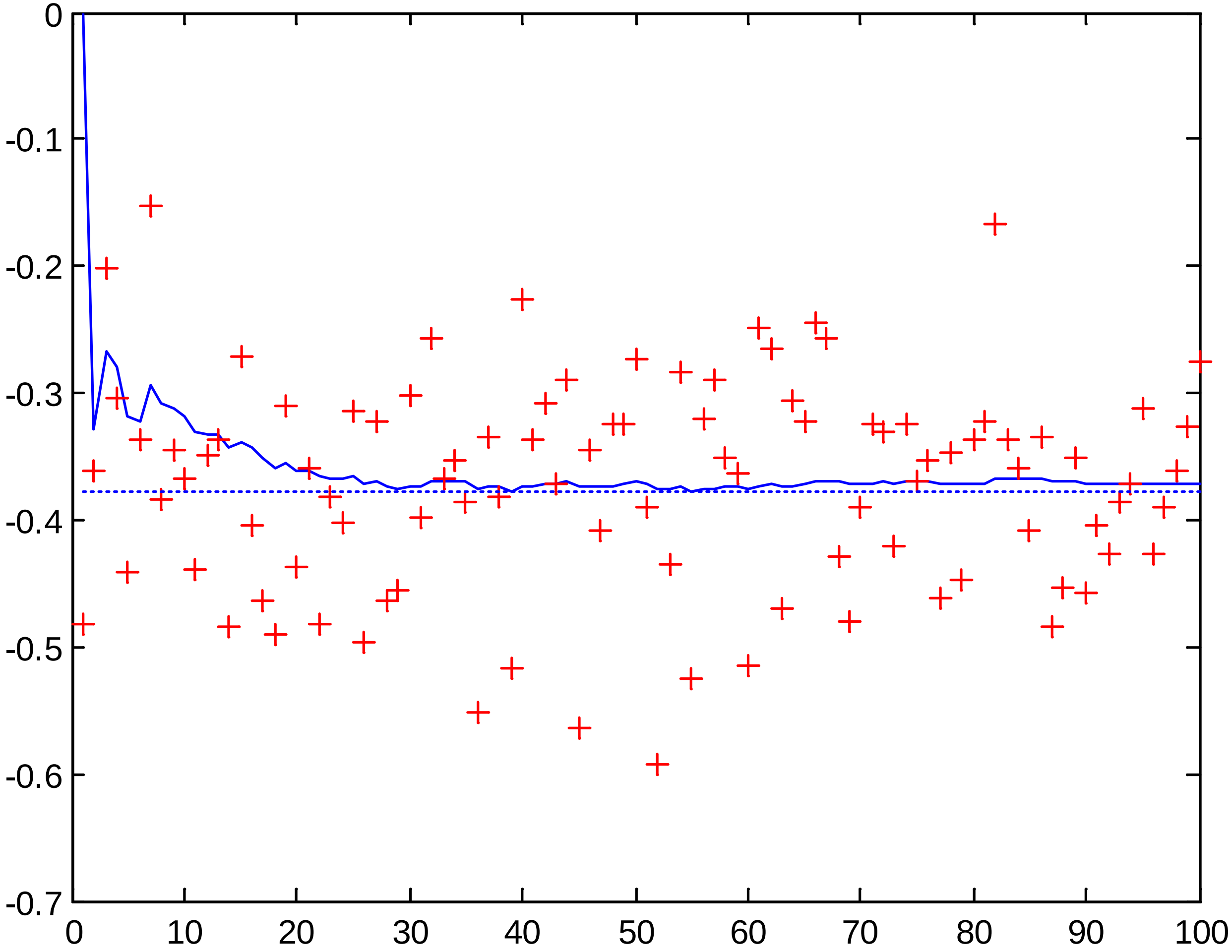
Correction

$$K = P^-_k(P^-_k + R)^{-1}$$

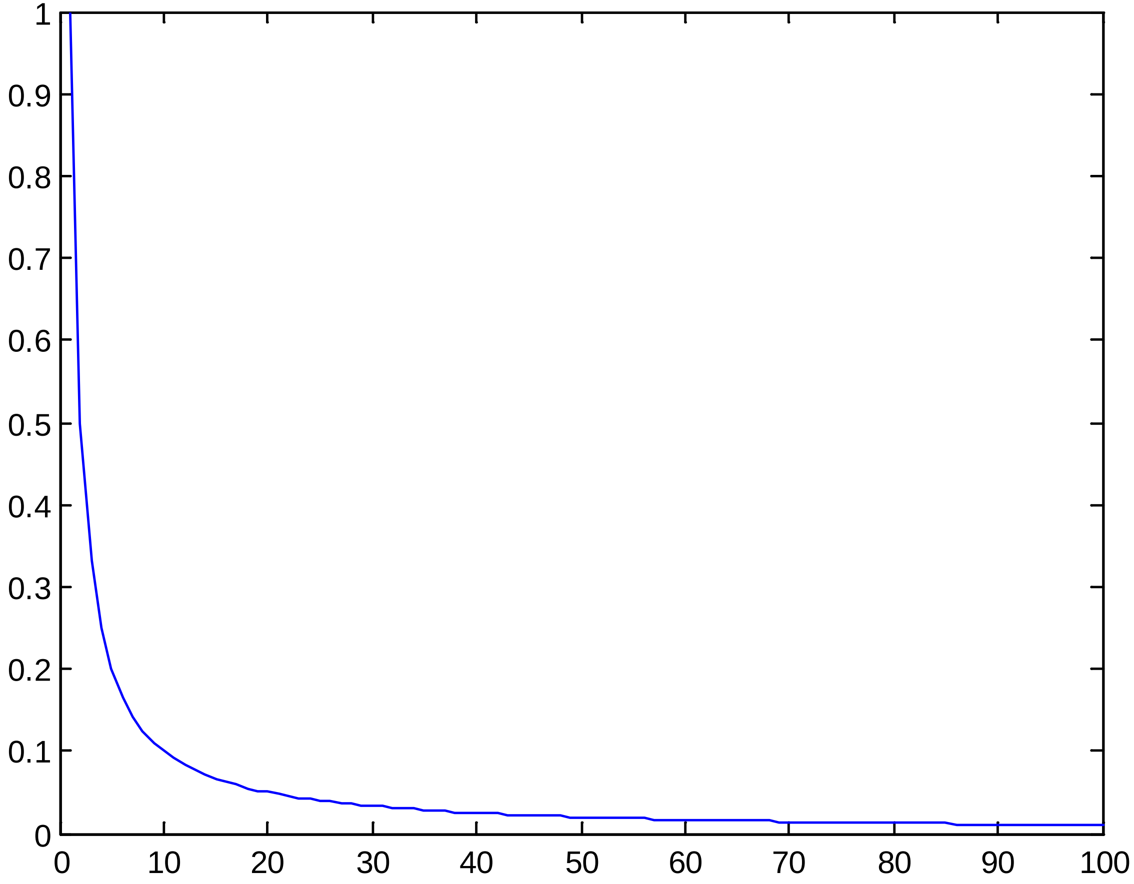
$$\hat{y}_k = \hat{y}^-_k + K(z_k - H\,\hat{y}^-_k)$$

$$P_k = (I - K)P^-_k$$

Quick Example – Constant Model

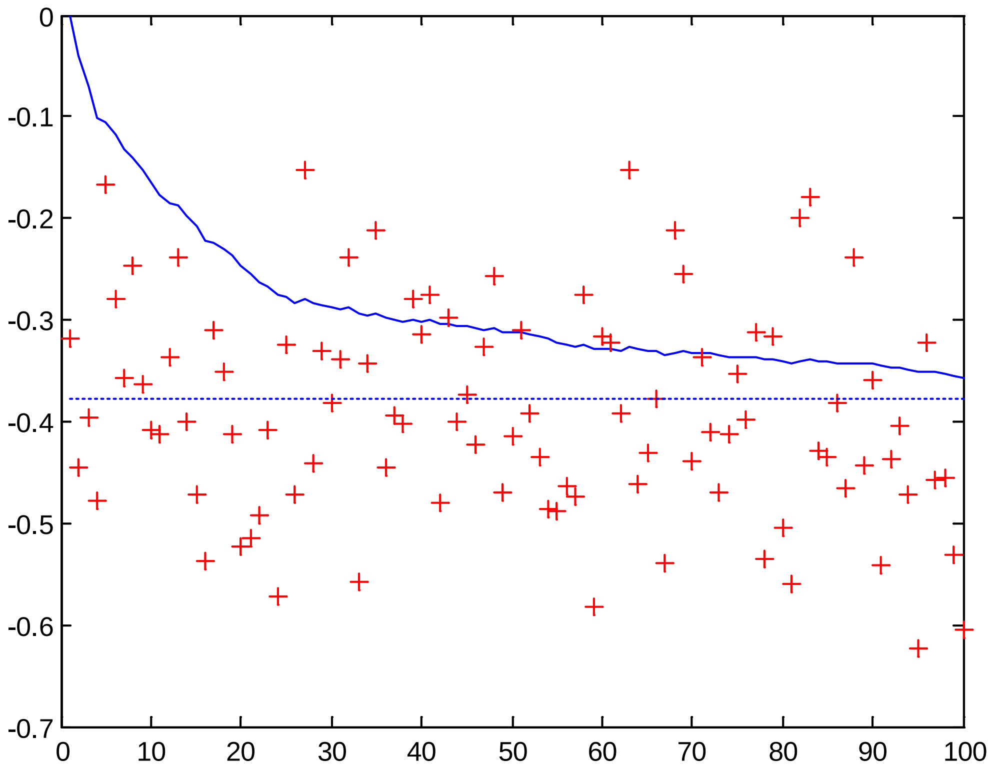


Quick Example – Constant Model



Convergence of Error Covariance - P_k

Quick Example – Constant Model



Larger value of R – the measurement error covariance (indicates poorer quality of measurements)

Filter slower to ‘believe’ measurements – slower convergence