Kalman Filter Estimation and Prediction Process, MATLAB Examples

Thursday, March 16, 2023    9:28 AM

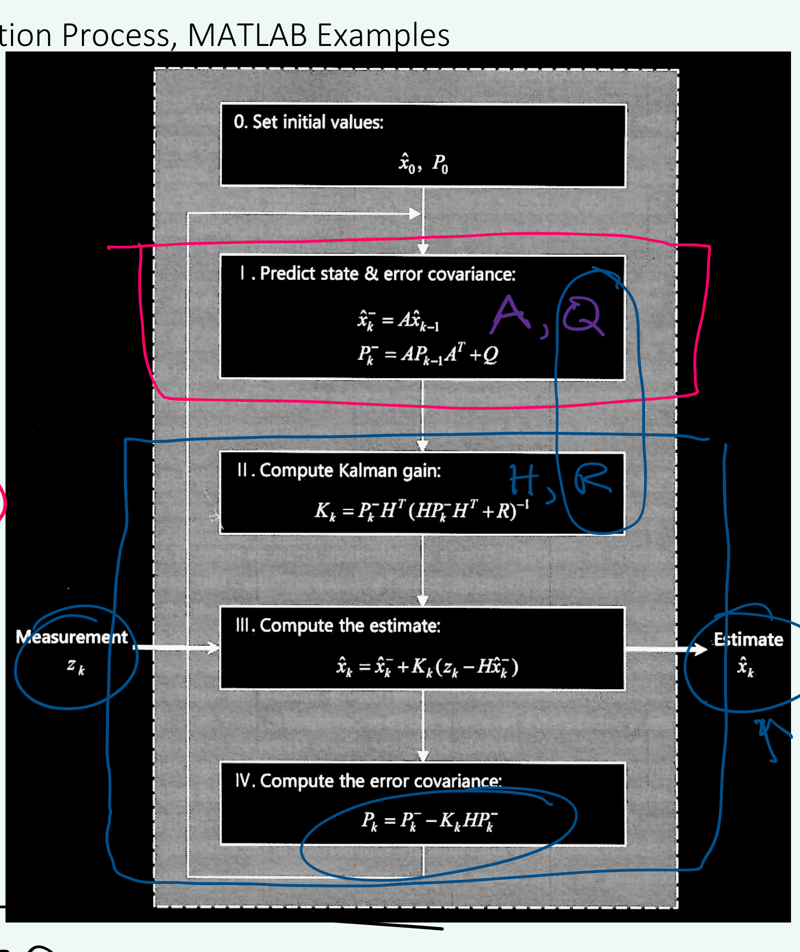
Kalman filter algorithm:

I. Prediction step

$x_k$ , $\hat{x}_k$ , $\hat{x}_k^-$

$x_k \sim N(\hat{x}_k, P_k)$

II-IV Estimation step




## Estimation step

Computation of an estimate ($x_k$ is $n \times 1$ column vector) — time $t_k$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \underbrace{H\hat{x}_k^-}_{\text{"predicted measurement"}})$$

true measurement

Re-write:

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

$n = 1$ , $H = 1$

$$\hat{x}_k = (1 - K_k)\hat{x}_k^- + K_k z_k$$

Let's substitute $\alpha$ for $1 - K_k$

$$\hat{x}_k = \alpha \hat{x}_k^- + (1-\alpha) z_k \quad \text{Kalman filter}$$

$$\bar{x}_k = \alpha \bar{x}_{k-1} + (1-\alpha) x_k \quad \text{Low pass filter (from last time)}$$

* Kalman filter estimation process is like a low pass filter w/ a dynamically changing $\alpha$ (or equivalently gain $K$)

The gain gets updated at each time $k$ according to

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

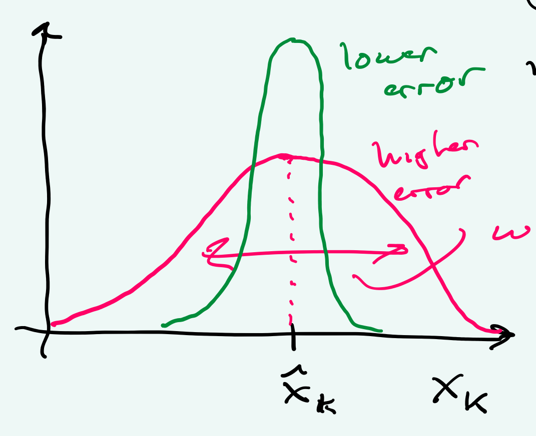
Error covariance updated as

$$P_k = P_k^- - K_k H P_k^-$$

Error covariance is a measure of the inaccuracy of the estimate

true state $x_k \sim \underbrace{N(\hat{x}_k, P_k)}$

normal distribution with mean $\hat{x}_k$ w/ covariance $P_k$ (spread of the "bell curve")




width related to $P_k$

if $P_k$ is "large", error is large

if $P_k$ is "small", error is small

Note: H and R only show up in estimation step

## Prediction step

Computation of the prediction

$\hat{x}_k$ at time $t_k$ $\xrightarrow{\text{predict}}$ $\hat{x}_{k+1}$ at time $t_{k+1}$

$n \times n$ matrix

$$\hat{x}_{k+1}^- = A\hat{x}_k$$
$$P_{k+1}^- = A P_k A^T + Q$$

"process noise" or "state transition noise"

"estimate" or "a priori" estimate

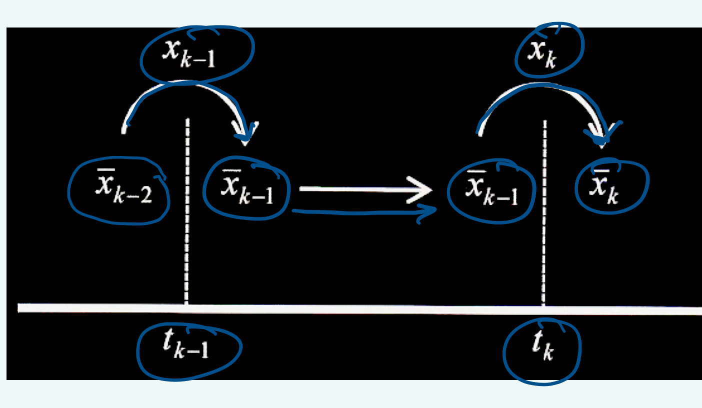
| | |
|---|---|
| $\hat{x}_k$ | Estimate of the state variable |
| $\hat{x}_k^-$ | Prediction of the state variable |
| $P_k$ | Estimate of the error covariance |
| $P_k^-$ | Prediction of the error covariance |

Note: A and Q only in prediction step

How do prediction and estimation fit together?

Low pass filter

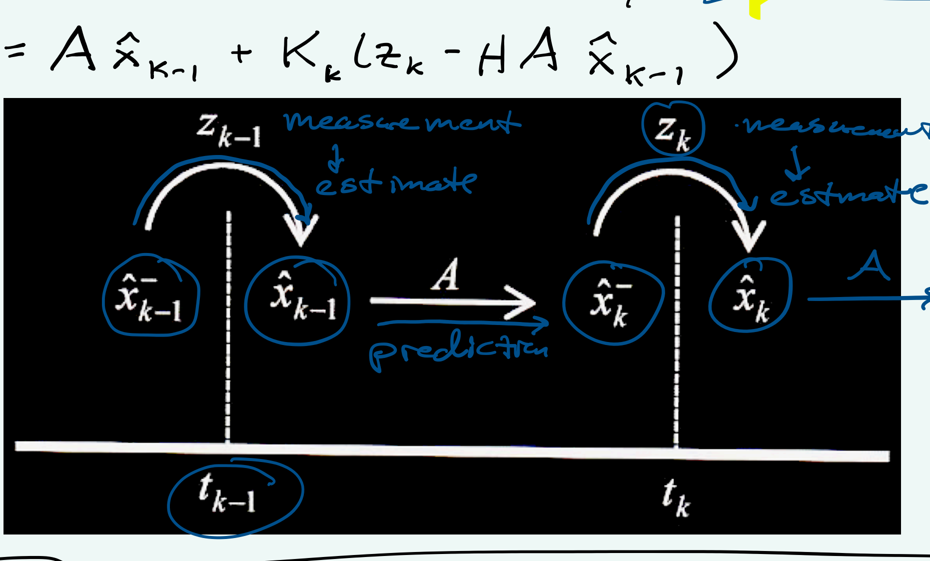
$$\bar{x}_k = \alpha \bar{x}_{k-1} + (1-\alpha) x_k$$




Kalman filter

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad \boxed{\hat{x}_k^- = A\hat{x}_{k-1}}$$

$$\hat{x}_k = A\hat{x}_{k-1} + K_k(z_k - HA\hat{x}_{k-1})$$




## System Model

The problem expressed in mathematics:

matrices : A, H, Q, R

Kalman filter deals with a linear state model

state: $\boxed{x_{k+1} = A x_k} + w_k$ — linear + process noise

measurement $\boxed{z_{k+1} = H x_k} + v_k$ — linear + measurement noise

$x_k$ = state variable ($n \times 1$ column vector)

$z_k$ = measurement ($m \times 1$ column vector)

$A$ = state transition matrix ($n \times n$ matrix)

$H$ = state-to-measurement matrix ($m \times n$ matrix)

$w_k$ = state transition noise ($n \times 1$ column vector)

$v_k$ = measurement noise ($m \times 1$ column vector)

Covariance of the noise

$Q$ = covariance matrix of $w_k$ ($n \times n$ diagonal matrix)

$R$ = covariance matrix of $v_k$ ($m \times m$ diagonal matrix)

if $n > 1$, $w_k = \begin{bmatrix} w_k^1 \\ w_k^2 \\ \vdots \\ w_k^n \end{bmatrix} \rightarrow \begin{matrix} x_k^1 \\ x_k^2 \\ \vdots \\ x_k^n \end{matrix} \rightarrow \underbrace{\begin{matrix} \sigma_1^2 \\ \sigma_2^2 \\ \\ \sigma_n^2 \end{matrix}}_{\text{variances}}$

$$Q = \begin{bmatrix} \sigma_1^2 & 0 & & \vdots \\ 0 & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n^2 \end{bmatrix} = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$$

$\sigma_i$ is the standard deviation

$\sigma_i^2$ is the variance

from trial and error

Similarly for R

(sensor specifications can give R as well)

$\sigma = \pm 2\,m \rightarrow \sigma^2 = 4\ (m^2)$

## Simple example (voltage estimate from a battery)

similar to last time, voltage is a constant

$m = 1$

z  measurement is voltage and has noise

x  state is also just the voltage (assume no noise)

$n = 1$

$x_{k+1} = A x_k$

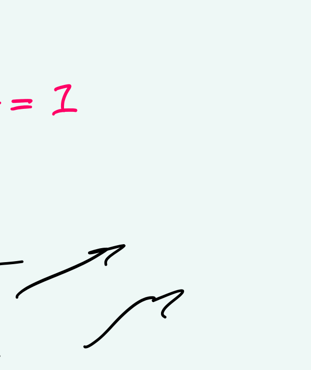
$x_{k+1} = x_k$ (voltage stays the same)

$z_k = H x_k$

$z_k = x_k + v_k$ (measurement is voltage + meas. noise)

Translate into matrices

$A = 1$, $H = 1$ ⁞ $Q = 0$, $R = 4$




Need initialization "guess"

$\hat{x}_0^- = 14$ , $P_0^- = 6$

MATLAB example

## Estimating velocity from position

$m = 1$

Setup system model ($n = 2$)

$x_k = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}_k$ , $x_{k+1} = A x_k + w_k$

$z_k = H x_k + v_k$

timestep btwn measurements

$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ , $H = [1\ \ 0]$ ($1 \times 2$ matrix)

($2 \times 2$ matrix)

$$x_{k+1} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}_k$$

$$= \begin{bmatrix} \text{position}_k + \Delta t \cdot \text{velocity}_k \\ \text{velocity}_k \end{bmatrix}$$

example $\Delta t = 0.02\ s$ for sonar altitude data

$$z_k = H x_k + v_k = [1\ \ 0] \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}_k + v_k$$

$$z_k = \text{position}_k + v_k$$

Noise is harder: $Q = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$ , $R = 10$

Next time, we'll do the MATLAB example