Learning about Kalman filter / Derivation of Equations

Resources

`Kalman Filter from Ground Up`; author Alex Becker; https://www.kalmanfilter.net

Overview

The equations of the `Kalman Filter` and their derivation.

State Extrapolation Equation

Also known under different names such as:

1. prediction equation
2. transition equation
3. state space model / dynamic model

$$\hat{\mathbf{x}}_{\mathbf{n+1,n}} = \mathbf{F} \cdot \hat{\mathbf{x}}_{\mathbf{n,n}} + \mathbf{G} \cdot \mathbf{u}_{\mathbf{n}} + \mathbf{w}_{\mathbf{n}}$$

property	description	dimension
$\hat{\mathbf{x}}_{\mathbf{n+1,n}}$	predicted state at time $n+1$	$n_x \times 1$
$\hat{\mathbf{x}}_{\mathbf{n,n}}$	estimated state at time n	$n_x \times 1$
$\mathbf{u}_{\mathbf{n}}$	driving input , deterministic	$n_u \times 1$
$\mathbf{w}_{\mathbf{n}}$	process noise; eg: models uncertainty of system model	$n_x \times 1$
$\mathbf{F}$	state transition matrix	$n_x \times n_x$
$\mathbf{G}$	control / input matrix	$n_x \times n_u$

The purpose of the `Kalman filter` is to estimated the state vector $\hat{\mathbf{x}}_{\mathbf{n,n}}$ in a way to minimise the uncertainty. Uncertainty is measured by the variance of the state vector.

The book provides an example to clarify the notation used so far:

example / airplane without control input

no control input means $\mathbf{u}_{\mathbf{n}}$.

If we assume an airplane moving in 3 dimensions the state vector provides information about the position, velocity and accelaration. And assuming that cartesial coordinates are used, that state vector $\mathbf{\hat{x}_{n,n}}$ at time n can be expressed like this:

$$\mathbf{\hat{x}_{n,n}} = \left[\hat{x}_{n,n}, \hat{y}_{n,n}, \hat{z}_{n,n}, \dot{x}_{n,n}, \dot{y}_{n,n}, \dot{z}_{n,n}, \ddot{x}_{n,n}, \ddot{y}_{n,n}, \ddot{z}_{n,n}\right]^T$$

The state extrapolation equation can now be expressed by (no noise, no inputs):

$$\mathbf{\hat{x}_{n+1,n}} = \mathbf{F} \cdot \mathbf{\hat{x}_{n,n}} = \begin{bmatrix} \hat{x}_{n+1,n} \\ \hat{y}_{n+1,n} \\ \hat{z}_{n+1,n} \\ \dot{x}_{n+1,n} \\ \dot{y}_{n+1,n} \\ \dot{z}_{n+1,n} \\ \ddot{x}_{n+1,n} \\ \ddot{y}_{n+1,n} \\ \ddot{z}_{n+1,n} \end{bmatrix} = \begin{bmatrix} \hat{x}_{n,n} + \Delta t \cdot \dot{x}_{n,n} + 0.5 \cdot \Delta t^2 \cdot \ddot{x}_{n,n} \\ \hat{y}_{n,n} + \Delta t \cdot \dot{y}_{n,n} + 0.5 \cdot \Delta t^2 \cdot \ddot{y}_{n,n} \\ \hat{z}_{n,n} + \Delta t \cdot \dot{z}_{n,n} + 0.5 \cdot \Delta t^2 \cdot \ddot{z}_{n,n} \\ \dot{x}_{n,n} + \Delta t \cdot \ddot{x}_{n,n} \\ \dot{y}_{n,n} + \Delta t \cdot \ddot{y}_{n,n} \\ \dot{z}_{n,n} + \Delta t \cdot \ddot{z}_{n,n} \\ \ddot{x}_{n,n} \\ \ddot{y}_{n,n} \\ \ddot{z}_{n,n} \end{bmatrix}$$

example / free falling object

The gravitational force acts upon the falling object. Accordingly we have constant acceleration.

The state vector has two elements:

1. the height or altitude

2. the velocity of the object

$$\mathbf{\hat{x}_{n,n}} = \begin{bmatrix} \hat{h}_n \\ \hat{\dot{h}}_n \end{bmatrix}$$

The dynamics are expressed by the state transition matrix $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

For the control matrix $\mathbf{G}$:

$$\mathbf{G} = \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}$$

And the input variable :

$$\mathbf{u_n} = g$$

Accordingly the state extrapolation equation becomes:

$$\hat{\mathbf{x}}_{\mathbf{n+1,n}} = \begin{bmatrix} \hat{h}_{n+1,n} \\ \dot{\hat{h}}_{n+1,n} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{h}_{n,n} \\ \dot{\hat{h}}_{n,n} \end{bmatrix} + g \cdot \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}$$

Covariance Extrapolation Equation

From the state extrapolation equation

$$\hat{\mathbf{x}}_{\mathbf{n+1,n}} = \mathbf{F} \cdot \hat{\mathbf{x}}_{\mathbf{n,n}} + \mathbf{G} \cdot \mathbf{u_n} + \mathbf{w_n}$$

the covariance $Cov(\hat{\mathbf{x}}_{\mathbf{n+1,n}})$ shall be computed.

From the definition of the covariance we have:

$$Cov(\hat{\mathbf{x}}_{\mathbf{n+1,n}}) = E\left(\left(\hat{\mathbf{x}}_{\mathbf{n+1,n}} - E(\hat{\mathbf{x}}_{\mathbf{n+1,n}})\right) \cdot \left(\hat{\mathbf{x}}_{\mathbf{n+1,n}} - E(\hat{\mathbf{x}}_{\mathbf{n+1,n}})\right)^T\right)$$

Computing expectations first:

$$\begin{aligned} E(\hat{\mathbf{x}}_{\mathbf{n+1,n}}) &= E\left(\mathbf{F} \cdot \hat{\mathbf{x}}_{\mathbf{n,n}}\right) + E\left(\mathbf{G} \cdot \mathbf{u_n}\right) + E\left(\mathbf{w_n}\right) \\ &= \mathbf{F} \cdot E\left(\hat{\mathbf{x}}_{\mathbf{n,n}}\right) + \mathbf{G} \cdot \mathbf{u_n} + E\left(\mathbf{w_n}\right) \end{aligned}$$

$$\begin{aligned} \left(\hat{\mathbf{x}}_{\mathbf{n+1,n}} - E(\hat{\mathbf{x}}_{\mathbf{n+1,n}})\right) &= \mathbf{F} \cdot \hat{\mathbf{x}}_{\mathbf{n,n}} + \mathbf{G} \cdot \mathbf{u_n} + \mathbf{w_n} - \mathbf{F} \cdot E\left(\hat{\mathbf{x}}_{\mathbf{n,n}}\right) + E\left(\mathbf{w_n}\right) \\ &= \mathbf{F} \cdot \hat{\mathbf{x}}_{\mathbf{n,n}} + \mathbf{G} \cdot \mathbf{u_n} + \mathbf{w_n} - \mathbf{F} \cdot E\left(\hat{\mathbf{x}}_{\mathbf{n,n}}\right) - \mathbf{G} \cdot \mathbf{u_n} - E\left(\mathbf{w_n}\right) \\ &= \mathbf{F} \cdot \left(\hat{\mathbf{x}}_{\mathbf{n,n}} - E\left(\hat{\mathbf{x}}_{\mathbf{n,n}}\right)\right) + \left(\mathbf{w_n} - E\left(\mathbf{w_n}\right)\right) \end{aligned}$$

In a preliminary step we now compute $\left(\hat{\mathbf{x}}_{\mathbf{n+1,n}} - E(\hat{\mathbf{x}}_{\mathbf{n+1,n}})\right) \cdot \left(\hat{\mathbf{x}}_{\mathbf{n+1,n}} - E(\hat{\mathbf{x}}_{\mathbf{n+1,n}})\right)^T$

$$(\hat{\mathbf{x}}_{\mathbf{n+1,n}} - E(\hat{\mathbf{x}}_{\mathbf{n+1,n}})) \cdot (\hat{\mathbf{x}}_{\mathbf{n+1,n}} - E(\hat{\mathbf{x}}_{\mathbf{n+1,n}}))^T = (\mathbf{F} \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) + (\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}}))) \cdot (\mathbf{F} \cdot ($$

$$= (\mathbf{F} \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) + (\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}}))) \cdot ((\hat{\mathbf{x}}_{\mathbf{n}}$$

$$= \mathbf{F} \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}}))^T \cdot \mathbf{F}^T +$$

$$+ \mathbf{F} \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) \cdot (\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}}))^T + (\mathbf{w}_{\mathbf{n}} -$$

Taking expectations on all terms yields the covariance:

$$Cov(\hat{\mathbf{x}}_{\mathbf{n+1,n}}) = \mathbf{F} \cdot E((\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}}))^T) \cdot \mathbf{F}^T + E((\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}})) \cdot (\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}$$

$$+ \mathbf{F} \cdot E((\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) \cdot (\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}}))^T) + E((\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}})) \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}}))$$

some simplication are now possible:

$$Cov(\hat{\mathbf{x}}_{\mathbf{n+1,n}}) = \mathbf{F} \cdot Cov(\hat{\mathbf{x}}_{\mathbf{n,n}}) \cdot \mathbf{F}^T + Cov(\mathbf{w}_{\mathbf{n}})$$

$$+ \mathbf{F} \cdot E((\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) \cdot (\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}}))^T) + E((\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}})) \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}}))$$

we still need to work on the expression

$$\mathbf{D} = \mathbf{D}_{\mathbf{1}} + \mathbf{D}_{\mathbf{1}}^T = \mathbf{F} \cdot E((\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) \cdot (\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}}))^T) + E((\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}})) \cdot (\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})$$

and begin with the first term:

$$\mathbf{D}_{\mathbf{1}} = \mathbf{F} \cdot E((\hat{\mathbf{x}}_{\mathbf{n,n}} - E(\hat{\mathbf{x}}_{\mathbf{n,n}})) \cdot (\mathbf{w}_{\mathbf{n}} - E(\mathbf{w}_{\mathbf{n}}))^T)$$

$$\mathbf{D}_{\mathbf{1}} = \mathbf{F} \cdot E(\hat{\mathbf{x}}_{\mathbf{n,n}} \cdot \mathbf{w}_{\mathbf{n}}{}^T) - \mathbf{F} \cdot E(\hat{\mathbf{x}}_{\mathbf{n,n}}) \cdot E(\mathbf{w}_{\mathbf{n}}{}^T) - \mathbf{F} \cdot E(\hat{\mathbf{x}}_{\mathbf{n,n}}) \cdot E(\mathbf{w}_{\mathbf{n}}{}^T) + \mathbf{F} \cdot E(\hat{\mathbf{x}}_{\mathbf{n,n}}) \cdot E(\mathbf{w}_{\mathbf{n}})^T$$

$$= \mathbf{F} \cdot (E(\hat{\mathbf{x}}_{\mathbf{n,n}} \cdot \mathbf{w}_{\mathbf{n}}{}^T) - E(\hat{\mathbf{x}}_{\mathbf{n,n}}) \cdot E(\mathbf{w}_{\mathbf{n}}{}^T))$$

Assumption

$\hat{\mathbf{x}}_{\mathbf{n,n}}$ and $\mathbf{w}_{\mathbf{n}}$ are statistical independent. Then

$E(\hat{\mathbf{x}}_{\mathbf{n,n}} \cdot \mathbf{w}_{\mathbf{n}}{}^T) = E(\hat{\mathbf{x}}_{\mathbf{n,n}}) \cdot E(\mathbf{w}_{\mathbf{n}}{}^T)$ and accordingly $\mathbf{D}_{\mathbf{1}} = \mathbf{0}$.

$$Cov(\hat{\mathbf{x}}_{\mathbf{n+1,n}}) = \mathbf{F} \cdot Cov(\hat{\mathbf{x}}_{\mathbf{n,n}}) \cdot \mathbf{F}^T + Cov(\mathbf{w}_{\mathbf{n}})$$

summary & notation

formula	description	changed notation
$Cov(\mathbf{\hat{x}_{n+1,n}})$	covariance matrix of extrapolated state	$\mathbf{P_{n+1,n}}$
$Cov\left(\mathbf{\hat{x}_{n,n}}\right)$	covariance matrix of estimated state	$\mathbf{P_{n,n}}$
$Cov\left(\mathbf{w_n}\right)$	covariance matrix of process noise	$\mathbf{Q_n}$

With these changes in the notation the covariance update equation becomes:

$$\mathbf{P_{n+1,n}} = \mathbf{F} \cdot \mathbf{P_{n,n}} \cdot \mathbf{F}^T + \mathbf{Q_n}$$

While the mathematics are not overly complicated there still remain some issues:

1. how to derive the state transition matrix for a specific problem
2. how should an initial value of covariance matrix $\mathbf{P_{0,0}}$ be chosen ?
3. what are the criteria that determine the appropriate choice of the covariance matrix $\mathbf{Q_n}$ of process noise ?

Process Noise

The concept of process noise is a bit vague. In the state update equation process noise is modelled as a random vector $\mathbf{w}_n$ that affects the extrapolated / predicted state. The book provides this definition:

$\mathbf{w}_n$ is an *unmeasurable* input that affects the state.

Other literatur just mentions that $\mathbf{w}_n$ is noise and uncorrelated to $\mathbf{\hat{x}_{n,n}}$.

Let us assume that process noise only affects the accelaration. The process noise vector can be expressed by:

$$\mathbf{w}_n = \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix}$$

Element w is a zero mean random variable with variance σ_w.

For the covariance matrix of $\mathbf{w}_n$ we get:

$$Cov\left(\mathbf{w}_n \cdot \mathbf{w}_n^T\right) = \mathbf{Q}_w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_w^2 \end{bmatrix}$$

For the covariance of $\mathbf{F} \cdot \mathbf{w}_n$ we obtain:

$$\mathbf{Q} = \mathbf{F} \cdot \mathbf{Q}_w \cdot \mathbf{F}^T$$

For a one dimensional constant motion model the state transition matrix $\mathbf{F}$ becomes:

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & 0.5 \cdot \Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Q} = \mathbf{F} \cdot \mathbf{Q}_w \cdot \mathbf{F}^T$$

$$= \begin{bmatrix} 1 & \Delta t & 0.5 \cdot \Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_w^2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ \Delta t & 1 & 0 \\ 0.5 \cdot \Delta t^2 & \Delta t & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \Delta t & 0.5 \cdot \Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.5 \cdot \Delta t^2 & \Delta t & 1 \end{bmatrix} \cdot \sigma_w^2$$

$$= \begin{bmatrix} 0.25 \cdot \Delta t^4 & 0.5 \cdot \Delta t^3 & 0.5 \cdot \Delta t^2 \\ 0.5 \cdot \Delta t^3 & \Delta t^2 & \Delta t \\ 0.5 \cdot \Delta t^2 & \Delta t & 1 \end{bmatrix} \cdot \sigma_w^2$$

Two dimensional case

$$\mathbf{w}_n = \begin{bmatrix} 0 \\ 0 \\ w_x \\ 0 \\ 0 \\ w_y \end{bmatrix}$$

$$Cov\left(\mathbf{w}_n \cdot \mathbf{w}_n^T\right) = \mathbf{Q}_w = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Var(w_x) & 0 & 0 & Cov(w_x, w_y) \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Cov(w_x, w_y) & 0 & 0 & Var(w_y \end{bmatrix}$$

Setting $Var(w_x) = Var(w_y) = \sigma_w^2$ and $Cov(w_x, w_y) = 0$:

$$Cov\left(\mathbf{w}_n \cdot \mathbf{w}_n^T\right) = \mathbf{Q}_w = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \sigma_w^2$$

For a two dimensional constant motion model the state transition matrix $\mathbf{F}$ becomes:

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & 0.5 \cdot \Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0.5 \cdot \Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{F} \cdot \mathbf{Q}_w = \begin{bmatrix} 1 & \Delta t & 0.5 \cdot \Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0.5 \cdot \Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \sigma_w^2 = \left[\begin{matrix} 0 & 0 & 0.5 \cdot \Delta t^2 \\ 0 & 0 & \Delta t \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}\right.$$

$$\mathbf{F} \cdot \mathbf{Q}_w \cdot \mathbf{F}^T = \begin{bmatrix} 0 & 0 & 0.5 \cdot \Delta t^2 & 0 & 0 & 0 \\ 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \cdot \Delta t^2 \\ 0 & 0 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \Delta t & 1 & 0 & 0 & 0 & 0 \\ 0.5 \cdot \Delta t^2 & \Delta t & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \Delta t & 1 & 0 \\ 0 & 0 & 0 & 0.5 \cdot \Delta t^2 & \Delta t & 1 \end{bmatrix} \cdot \sigma_w^2$$

$$\mathbf{F} \cdot \mathbf{Q}_w \cdot \mathbf{F}^T = \begin{bmatrix} 0.25 \cdot \Delta t^4 & 0.5 \cdot \Delta t^3 & 0.5 \cdot \Delta t^2 & 0 & 0 & 0 \\ 0.5 \cdot \Delta t^3 & \Delta t^2 & \Delta t & 0 & 0 & 0 \\ 0.5 \cdot \Delta t^2 & \Delta t & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 \cdot \Delta t^4 & 0.5 \cdot \Delta t^3 & 0.5 \cdot \Delta t^2 \\ 0 & 0 & 0 & 0.5 \cdot \Delta t^3 & \Delta t^2 & \Delta t \\ 0 & 0 & 0 & 0.5 \cdot \Delta t^2 & \Delta t & 1 \end{bmatrix} \cdot \sigma_w^2$$

Measurement

At time step n a new measurement $\mathbf{z_n}$ is available. The measurement equation in vector notation is:

$$\mathbf{z_n} = \mathbf{H} \cdot \mathbf{x_n} + \mathbf{v_n}$$

property	description	dimension
$\mathbf{z_n}$	measurement	$n_z \times 1$
$\mathbf{H}$	observation matrix	$n_z \times n_x$
$\mathbf{x_n}$	true system state (different from estimated state)	$n_x \times 1$
$\mathbf{v_n}$	measurement noise	$n_z \times 1$

The observation matrix $\mathbf{H}$ accounts for the fact that most measurements do not reflect the state directly . Thus the term `hidden state` is frequently found in the literature.

The book provides an example of measuring a hidden state.

A distance x_n is measured using an echo-meter. Therefore the measurement is about a delay t_n. The speed of a propagating sound wave is c. So what is actually measured is $t_n = 2 \cdot \frac{x_n}{c}$.

On the other hand the number of elements n_z in measurement $\mathbf{z_n}$ may be different from the number n_x of states $\mathbf{x_n}$. Thus only a **selection** of state variables may contribute to $\mathbf{z_n}$.

And there may be a sitation where a **linear combination** of state variables contribute to an element of the measurement vector.

Summary / What we have learned so far

Basically this is a re-write of chapter 8.4 (interim summary) of

`Kalman Filter from Ground Up`; author Alex Becker; https://www.kalmanfilter.net

prediction equations

state extrapolation equation

$$\mathbf{\hat{x}_{n+1,n}} = \mathbf{F} \cdot \mathbf{\hat{x}_{n,n}} + \mathbf{G} \cdot \mathbf{u_n} + \mathbf{w_n}$$

and based on the state update equation the covariance update equation has been derived:

covariance extrapolation equation

$$\mathbf{P_{n+1,n}} = \mathbf{F} \cdot \mathbf{P_{n,n}} \cdot \mathbf{F}^T + \mathbf{Q_n}$$

auxiliary equations

The measurement equation belongs to this category .

measurement equation

$$\mathbf{z_n} = \mathbf{H} \cdot \mathbf{x_n} + \mathbf{v_n}$$

Covariance equations

covariance of measurement uncertainty

$$\mathbf{R_n} = E\left(\mathbf{v_n} \cdot \mathbf{v_n}^T\right)$$

covariance of process noise

$$\mathbf{Q_n} = E\left(\mathbf{w_n} \cdot \mathbf{w_n}^T\right)$$

estimation uncertainty

$$\mathbf{P_{n,n}} = E\left(\left(\mathbf{x_n} - \mathbf{\hat{x}_{n,n}}\right) \cdot \left(\mathbf{x_n} - \mathbf{\hat{x}_{n,n}}\right)^T\right)$$

The State Update Equation

The book just states the `state update equation` without deriving it beforehand. The derivation is done in later chapters of the book. According to my personal taste I would have preferred to see a derivation in the first place. So perhaps I will find other resources that might fill this gap.

$$\mathbf{\hat{x}_{n,n}} = \mathbf{\hat{x}_{n,n-1}} + \mathbf{K_n} \cdot \left(\mathbf{z_n} - \mathbf{H} \cdot \mathbf{\hat{x}_{n,n-1}}\right)$$

Basically this equation combines the predicted state $\mathbf{\hat{x}_{n,n-1}}$ with some kind of measurement error $\left(\mathbf{z_n} - \mathbf{H} \cdot \mathbf{\hat{x}_{n,n-1}}\right)$ with some weighting matrix $\mathbf{K_n}$. This weighting is called the `Kalman Gain Matrix` or simply `Kalman Gain`.

The goal is to evaluate $\mathbf{K_n}$ is a way to minimise the trace of covariance matrix .

Starting point is the state update equation:

$$\mathbf{\hat{x}_{n,n}} = \mathbf{\hat{x}_{n,n-1}} + \mathbf{K_n} \cdot \left(\mathbf{z_n} - \mathbf{H} \cdot \mathbf{\hat{x}_{n,n-1}}\right)$$

Inserting the expression for the measurement vector $\mathbf{z_n}$

$$\mathbf{z_n} = \mathbf{H} \cdot \mathbf{x_n} + \mathbf{v_n}$$

$$\mathbf{\hat{x}_{n,n}} = \mathbf{\hat{x}_{n,n-1}} + \mathbf{K_n} \cdot \left(\mathbf{H} \cdot \mathbf{x_n} + \mathbf{v_n} - \mathbf{H} \cdot \mathbf{\hat{x}_{n,n-1}}\right)$$

Compute the estimation error $\mathbf{e_n}$:

$$\begin{aligned}
\mathbf{e_n} &= \mathbf{x_n} - \mathbf{\hat{x}_{n,n}} \\
&= \mathbf{x_n} - \mathbf{\hat{x}_{n,n-1}} - \mathbf{K_n} \cdot \left(\mathbf{H} \cdot \mathbf{x_n} + \mathbf{v_n} - \mathbf{H} \cdot \mathbf{\hat{x}_{n,n-1}}\right) \\
&= \left(\mathbf{x_n} - \mathbf{\hat{x}_{n,n-1}}\right) - \mathbf{K_n} \cdot \mathbf{H} \cdot \left(\mathbf{x_n} - \mathbf{\hat{x}_{n,n-1}}\right) - \mathbf{K_n} \cdot \mathbf{v_n} \\
&= \left(\mathbf{I} - \mathbf{K_n} \cdot \mathbf{H}\right) \cdot \left(\mathbf{x_n} - \mathbf{\hat{x}_{n,n-1}}\right) - \mathbf{K_n} \cdot \mathbf{v_n}
\end{aligned}$$

Now that we have found an expression for the estimation error $\mathbf{e_n}$ we are going to compute the covariance matrix $\mathbf{P_{n,n}}$:

$$
\begin{aligned}
\mathbf{P_{n,n}} &= E\left(\mathbf{e_n}\cdot\mathbf{e_n}^T\right)\\
&= E\left(\left(\left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)\cdot\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)-\mathbf{K_n}\cdot\mathbf{v_n}\right)\cdot\left(\left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)\cdot\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)-\mathbf{K_n}\cdot\mathbf{v_n}\right)^T\right)\\
&= \left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)\cdot E\left(\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)\cdot\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)^T\right)\cdot\left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)^T+\mathbf{K_n}\cdot E\left(\mathbf{v_n}\cdot\mathbf{v_n}^T\right)\cdot\mathbf{K_n}^T\\
&- \left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)\cdot E\left(\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)\cdot\mathbf{v_n}^T\right)\cdot\mathbf{K_n}^T-\mathbf{K_n}\cdot E\left(\mathbf{v_n}\cdot\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)^T\right)\cdot\left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)^T
\end{aligned}
$$

quantity

$$
E\left(\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)\cdot\mathbf{v_n}^T\right)=\mathbf{0}
$$

since measurements at time n are uncorrelated / independent from extrapolated state $\mathbf{\hat{x}_{n,n-1}}$.

$$
\begin{aligned}
\mathbf{P_{n,n}} &= E\left(\mathbf{e_n}\cdot\mathbf{e_n}^T\right)\\
&= \left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)\cdot E\left(\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)\cdot\left(\mathbf{x_n}-\mathbf{\hat{x}_{n,n-1}}\right)^T\right)\cdot\left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)^T+\mathbf{K_n}\cdot E\left(\mathbf{v_n}\cdot\mathbf{v_n}^T\right)\cdot\mathbf{K_n}^T\\
\mathbf{P_{n,n}} &= \left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)\cdot\mathbf{P_{n,n-1}}\cdot\left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)^T+\mathbf{K_n}\cdot\mathbf{R_n}\cdot\mathbf{K_n}^T
\end{aligned}
$$

which is the covariance update equation expressed via `Kalman gain` $\mathbf{K_n}$.

What is left is to derive the equation for the `Kalman gain`.

Kalman Gain / Derivation

We start with the covariance update equation and do some re-arragements of the equation:

$$
\begin{aligned}
\mathbf{P_{n,n}} &= \left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)\cdot\mathbf{P_{n,n-1}}\cdot\left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)^T+\mathbf{K_n}\cdot\mathbf{R_n}\cdot\mathbf{K_n}^T\\
&= \left(\mathbf{I}-\mathbf{K_n}\cdot\mathbf{H}\right)\cdot\mathbf{P_{n,n-1}}\cdot\left(\mathbf{I}-\mathbf{H}^T\cdot\mathbf{K_n}^T\right)+\mathbf{K_n}\cdot\mathbf{R_n}\cdot\mathbf{K_n}^T\\
&= \left(\mathbf{P_{n,n-1}}-\mathbf{K_n}\cdot\mathbf{H}\cdot\mathbf{P_{n,n-1}}\right)\cdot\left(\mathbf{I}-\mathbf{H}^T\cdot\mathbf{K_n}^T\right)+\mathbf{K_n}\cdot\mathbf{R_n}\cdot\mathbf{K_n}^T\\
&= \mathbf{P_{n,n-1}}-\mathbf{P_{n,n-1}}\cdot\mathbf{H}^T\cdot\mathbf{K_n}^T-\mathbf{K_n}\cdot\mathbf{H}\cdot\mathbf{P_{n,n-1}}+\mathbf{K_n}\cdot\mathbf{H}\cdot\mathbf{P_{n,n-1}}\cdot\mathbf{H}^T\cdot\mathbf{K_n}^T+\mathbf{K_n}\cdot\mathbf{R_n}\cdot\mathbf{K_n}\\
&= \mathbf{P_{n,n-1}}-\mathbf{P_{n,n-1}}\cdot\mathbf{H}^T\cdot\mathbf{K_n}^T-\mathbf{K_n}\cdot\mathbf{H}\cdot\mathbf{P_{n,n-1}}+\mathbf{K_n}\cdot\left(\mathbf{H}\cdot\mathbf{P_{n,n-1}}\cdot\mathbf{H}^T+\mathbf{R_n}\right)\cdot\mathbf{K_n}^T
\end{aligned}
$$

The last equation is also in the book as table 8.6.

We need to minimise the variance of $\mathbf{P_{n,n}}$ subjects to the matrix elements of `Kalman gain` $\mathbf{K_n}$. The aggregate variances of $\mathbf{P_{n,n}}$ is just the trace $tr\left(\mathbf{P_{n,n}}\right)$ of this matrix.

For $tr\left(\mathbf{P_{n,n}}\right)$ we get:

$$\begin{aligned} tr\left(\mathbf{P_{n,n}}\right) &= tr\left(\mathbf{P_{n,n-1}}\right) - tr\left(\mathbf{P_{n,n-1}} \cdot \mathbf{H}^T \cdot \mathbf{K_n}^T\right) - tr\left(\mathbf{K_n} \cdot \mathbf{H} \cdot \mathbf{P_{n,n-1}}\right) + tr\Big(\mathbf{K_n} \cdot \Big(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T \cdot \\ &= tr\left(\mathbf{P_{n,n-1}}\right) - 2 \cdot tr\left(\mathbf{P_{n,n-1}} \cdot \mathbf{H}^T \cdot \mathbf{K_n}^T\right) + tr\left(\mathbf{K_n} \cdot \left(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + \mathbf{R_n}\right) \cdot \mathbf{K_n}^T\right) \\ &= tr\left(\mathbf{P_{n,n-1}}\right) - 2 \cdot tr\left(\mathbf{K_n} \cdot \mathbf{H} \cdot \mathbf{P_{n,n-1}}\right) + tr\left(\mathbf{K_n} \cdot \left(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + \mathbf{R_n}\right) \cdot \mathbf{K_n}^T\right) \end{aligned}$$

In this derivation we used the property $tr(\mathbf{A}) = tr\left(\mathbf{A^T}\right)$ and the fact that covariance matrices are symmetric.

Before we differentiate these traces with respect to $\mathbf{K_n}$ two useful differentiation rules shall be derived:

---

differentiation rule #1

$$\frac{d}{d\mathbf{A}}(tr(\mathbf{A} \cdot \mathbf{B}))$$

Dimensions of matrices:

$\mathbf{A};\, a_{m,n} \in \mathrm{R}^{m \times n}$

$\mathbf{B};\, b_{n,m} \in \mathrm{R}^{n \times m}$

$$tr(\mathbf{A} \cdot \mathbf{B}) = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j} \cdot b_{j,i}$$

Differentiation with respect to some matrix element $a_{k,l}$ yields:

$$\frac{d}{d\,a_{k,l}}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j} \cdot b_{j,i}\right) = b_{l,k}$$

$$\frac{d}{d\mathbf{A}}(tr(\mathbf{A} \cdot \mathbf{B})) = \begin{bmatrix} b_{1,1} & \cdots & b_{n,1} \\ \vdots & \ddots & \vdots \\ b_{1,m} & \cdots & b_{n,m} \end{bmatrix} = \mathbf{B}^T$$

differentiation rule #2

$$\frac{d}{d\mathbf{A}}\left(tr\left(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{A}^T\right)\right)$$

Dimensions of matrices:

$\mathbf{A}; a_{m,n} \in \mathrm{R}^{m \times n}$

$\mathbf{B}; b_{n,n} \in \mathrm{R}^{n \times n};\ \mathit{symmetric}$

We define matrices $\mathbf{D};\ \ \in \mathrm{R}^{m \times m}$ and $\mathbf{C};\ \ \in \mathrm{R}^{m \times n}$ like this:

$$\mathbf{D} = \underbrace{\mathbf{A}}_{\mathbf{C}} \cdot \mathbf{B} \cdot \mathbf{A}^T$$

$$= \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} \cdot \begin{bmatrix} b_{1,1} & \cdots & b_{1,n} \\ \vdots & \ddots & \vdots \\ b_{n,1} & \cdots & b_{n,n} \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} & \cdots & a_{m,1} \\ \vdots & \ddots & \vdots \\ a_{1,n} & \cdots & a_{m,n} \end{bmatrix}$$

Let $c_{k,l}$ denote a matrix element of $\mathbf{C}$ with :

$$c_{k,l} = \sum_{j=1}^{n} a_{k,j} \cdot b_{j,l}$$

Similarly let $d_{h,g}$ denote a matrix element of $\mathbf{D}$ with :

$$d_{h,g} = \sum_{i=1}^{n} c_{h,i} \cdot a_{g,i}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} a_{h,j} \cdot b_{j,i} \cdot a_{g,i}$$

For the trace $tr(\mathbf{D})$ we get:

$$tr(\mathbf{D}) = \sum_{h=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{h,j} \cdot b_{j,i} \cdot a_{h,i}$$

Differentiation with respect to some matrix element $a_{r,s}$ yields:

$$\frac{d}{d\,a_{r,s}} \left(\sum_{h=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{h,j} \cdot b_{j,i} \cdot a_{h,i} \right) = \sum_{i=1}^{n} b_{s,i} \cdot a_{r,i} + \sum_{j=1}^{n} a_{r,j} \cdot b_{j,s}$$

$$= \sum_{i=1}^{n} b_{s,i} \cdot a_{r,i} + \sum_{j=1}^{n} a_{r,j} \cdot b_{s,j}$$

$$= \sum_{j=1}^{n} a_{r,j} \cdot b_{s,j}$$

$$\frac{d}{d\mathbf{A}} \left(tr\left(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{A}^T \right) \right) = 2 \cdot \mathbf{A} \cdot \mathbf{B}$$

These two rules for differentiation are now applied.

We need to calculate:

$$\frac{d}{d\mathbf{K_n}}\Big(tr\big(\mathbf{P_{n,n}}\big)\Big) = \frac{d}{d\mathbf{K_n}}\Big(tr\big(\mathbf{P_{n,n-1}}\big)\Big) - 2 \cdot \frac{d}{d\mathbf{K_n}}\Big(tr\big(\mathbf{K_n} \cdot \mathbf{H} \cdot \mathbf{P_{n,n-1}}\big)\Big) + \frac{d}{d\mathbf{K_n}}\Big(tr\big(\mathbf{K_n} \cdot \big(\mathbf{H} \cdot \mathbf{P_{n,n-1}}$$

Since the covariance matrix $\mathbf{P_{n,n-1}}$ does not depend on the `Kalman gain` the derivative is 0.

$$\frac{d}{d\mathbf{K_n}}\left(tr\left(\mathbf{P_{n,n-1}}\right)\right) = \mathbf{0}$$

For the next equation we apply rule#1:

$$\begin{aligned} \frac{d}{d\mathbf{K_n}}\left(tr\left(\mathbf{K_n} \cdot \mathbf{H} \cdot \mathbf{P_{n,n-1}}\right)\right) &= \left(\mathbf{H} \cdot \mathbf{P_{n,n-1}}\right)^T \\ &= \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T \end{aligned}$$

and here we apply rule#2:

$$\frac{d}{d\mathbf{K_n}}\left(tr\left(\mathbf{K_n} \cdot \left(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + \mathbf{R_n}\right) \cdot \mathbf{K_n}^T\right)\right) = 2 \cdot \mathbf{K_n} \cdot \left(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + \mathbf{R_n}\right)$$

Applying these equation we get:

$$\frac{d}{d\mathbf{K_n}}\left(tr\left(\mathbf{P_{n,n}}\right)\right) = -2 \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + 2 \cdot \mathbf{K_n} \cdot \left(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + \mathbf{R_n}\right)$$

The `Kalman gain` is found by setting all derivatives to zero:

$$\frac{d}{d\mathbf{K_n}}\left(tr\left(\mathbf{P_{n,n}}\right)\right) = \mathbf{0}$$

Which gives us an equation

$$\mathbf{P_{n,n-1}} \cdot \mathbf{H}^T = \mathbf{K_n} \cdot \left(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + \mathbf{R_n}\right)$$

which is solved for the `Kalman Gain` $\mathbf{K_n}$:

$$\mathbf{K_n} = \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T \cdot \left(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + \mathbf{R_n}\right)^{-1}$$

What remains to be done is to find an **efficient** way to evaluate the inverse matrix

$$\left(\mathbf{H} \cdot \mathbf{P_{n,n-1}} \cdot \mathbf{H}^T + \mathbf{R_n}\right)^{-1}$$

Final Summary

The important equations around the `Kalman filter` are again summarised here. As before the presentation here is a re-write of chapter 8.9 of `Kalman Filter from Ground Up`; author Alex Becker; https://www.kalmanfilter.net.

In essence the filtering operation involves a prediction step and a correction step.

The prediction step involves 2 equations + 2 intial conditions:

equation	description
$\hat{\mathbf{x}}_{n+1,n} = \mathbf{F} \cdot \hat{\mathbf{x}}_{n,n} + \mathbf{G} \cdot \mathbf{u}_n$	state extrapolation / prediction
$\mathbf{P}_{n+1,n} = \mathbf{F} \cdot \mathbf{P}_{n,n} \cdot \mathbf{F}^T$	extrapolation of uncertainty
$\hat{\mathbf{x}}_{0,0}$	assumptions about the initial state
$\mathbf{P}_{0,0}$	assumption of covariance of initial state

The correction step involve 3 equations:

equation	description
$\mathbf{K}_{\mathbf{n}} = \mathbf{P}_{\mathbf{n},\mathbf{n-1}} \cdot \mathbf{H}^T \cdot \left(\mathbf{H} \cdot \mathbf{P}_{\mathbf{n},\mathbf{n-1}} \cdot \mathbf{H}^T + \mathbf{R}_{\mathbf{n}}\right)^{-1}$	`Kalman Gain`
$\hat{\mathbf{x}}_{\mathbf{n},\mathbf{n}} = \hat{\mathbf{x}}_{\mathbf{n},\mathbf{n-1}} + \mathbf{K}_{\mathbf{n}} \cdot \left(\mathbf{z}_{\mathbf{n}} - \mathbf{H} \cdot \hat{\mathbf{x}}_{\mathbf{n},\mathbf{n-1}}\right)$	state update equation
$\mathbf{P}_{\mathbf{n},\mathbf{n}} = \left(\mathbf{I} - \mathbf{K}_{\mathbf{n}} \cdot \mathbf{H}\right) \cdot \mathbf{P}_{\mathbf{n},\mathbf{n-1}} \cdot \left(\mathbf{I} - \mathbf{K}_{\mathbf{n}} \cdot \mathbf{H}\right)^T + \mathbf{K}_{\mathbf{n}} \cdot \mathbf{R}_{\mathbf{n}} \cdot \mathbf{K}_{\mathbf{n}}{}^T$	update uncertainty of estimate

And here is an overview of the `workflow` (eg. the sequence in which these equations are applied in each iteration):

initialisation

Based on some prior knowledge we choose the initial state estimate $\hat{\mathbf{x}}_{0,0}$ and the initial covariance $\mathbf{P}_{0,0}$ of this estimate.

Using these initial properties the extrapolated state and the extrapolated uncertainty are evaluated:

state extrapolation

$$\hat{\mathbf{x}}_{1,0} = \mathbf{F} \cdot \hat{\mathbf{x}}_{0,0} + \mathbf{G} \cdot \mathbf{u}_1$$

uncertainty extrapolation

$$\mathbf{P}_{1,0} = \mathbf{F} \cdot \mathbf{P}_{0,0} \cdot \mathbf{F}^T$$

computation of Kalman gain

$$\mathbf{K_1} = \mathbf{P_{1,0}} \cdot \mathbf{H}^T \cdot \left(\mathbf{H} \cdot \mathbf{P_{1,0}} \cdot \mathbf{H}^T + \mathbf{R_1}\right)^{-1}$$

state estimation / state update

$$\hat{\mathbf{x}}_{\mathbf{1,1}} = \hat{\mathbf{x}}_{\mathbf{1,0}} + \mathbf{K_1} \cdot \left(\mathbf{z_1} - \mathbf{H} \cdot \hat{\mathbf{x}}_{\mathbf{1,0}}\right)$$

update uncertainty of estimated state $\hat{\mathbf{x}}_{\mathbf{1,1}}$

$$\mathbf{P_{1,1}} = \left(\mathbf{I} - \mathbf{K_1} \cdot \mathbf{H}\right) \cdot \mathbf{P_{1,0}} \cdot \left(\mathbf{I} - \mathbf{K_1} \cdot \mathbf{H}\right)^T + \mathbf{K_1} \cdot \mathbf{R_1} \cdot \mathbf{K_1}^T$$

Having found $\hat{\mathbf{x}}_{\mathbf{1,1}}$ and $\mathbf{P_{1,1}}$ the process is repeated.

We calculate $\hat{\mathbf{x}}_{2,1}$ from $\hat{\mathbf{x}}_{\mathbf{1,1}}$ and possibly a new input $\mathbf{u}_2$ and $\mathbf{P_{2,1}}$ from $\mathbf{P_{1,1}}$.

Then we are able to use these values to compute:

1. the Kalman gain $\mathbf{K_2}$

2. the state estimate $\hat{\mathbf{x}}_{\mathbf{2,2}}$

3. and the uncertainty $\mathbf{P_{2,2}}$.

and so on ...

Conclusion

Most of the derivation of the formulas which define the `Kalman filter` are clearly described in the book in great detail without leaving out intermediate steps. This helps to really understand the derivation. Many other book are very concise. You must then try to reconstruct the intermediate steps. Not only is this time consuming you may likely fail to do so. This a common source of frustation with many textbooks which try to be overly concise in how specific topics are presented.

My only critisism is the fact that the state estimation formula has not been derived. The `Kalman gain` used in this formula seems to come from **nowhere**.

In [ ]:

```

```