KALMAN FILTER
Univariate and multivariate analysis

Olga Nínive Marín Jiménez

Contents

1 Introduction

In this document we are going to discuss about the Kalman Filter. We start by looking at the simpler version, the Uni-variate Kalman Filter, and talk about its important parts, such as how it makes predictions and adjusts them with new information. Then, we step into the Multivariate Kalman Filter, which handles more complex situations. Besides, we use examples like estimating building height and predicting how vehicles move to show how this filter works in the real world.

2 Uni-variate Kalman Filter

The Kalman Filter's goal is to estimate and predict system states in the presence of uncertainty. Moreover, treats measurements (z_n), current state estimation ($\hat{x}_n$), and next state estimation (predictions) ($\hat{x}_{n+1}$) as normally distributed random variables which are described by mean and variance.

The Kalman Filter output is a random variable with the mean as the state estimate (x) and the variance as the estimation uncertainty (p), so it provides the estimate and the confidence level of it.

2.1 Equations

The Kalman Filter computations are based on five equations.

Two prediction equations:

- State Extrapolation Equation - predicting or estimating the future state based on the known present estimation.

$$\hat{x}_{n+1,n} = \hat{x}_{n,n}$$

(for constant dynamics)

$$\hat{x}_{n+1,n} = \hat{x}_{n,n} + \Delta t\hat{\dot{x}}_{n,n}$$
$$\hat{\dot{x}}_{n+1,n} = \hat{\dot{x}}_{n,n}$$

(for constant velocity dynamics)

- Covariance Extrapolation Equation - the measure of uncertainty in our prediction.

$$p^x_{n+1,n} = p^x_{n,n}$$

(for constant dynamics)

$$p^x_{n+1,n} = p^x_{n,n} + \Delta t^2 p^v_{n,n}$$
$$p^v_{n+1,n} = p^v_{n,n}$$

(for constant velocity dynamics)

Kalman Gain Equation - required for computation of the update equations. The Kalman Gain is a "weighting" parameter for the measurement and the past estimations. It defines the weight of the past estimation and the weight of the measurement in estimating the current state. We get this value by trying to minimize

$$p_{n,n} = w_1^2 r_n + (1 - w_1)^2 p_{n,n-1} \qquad \frac{dp_{n,n}}{dw_1} = 0$$

$$K_n = \frac{p_{n,n-1}}{p_{n,n-1} + r_n} \qquad 0 \le K_n \le 1$$

Two update equations:

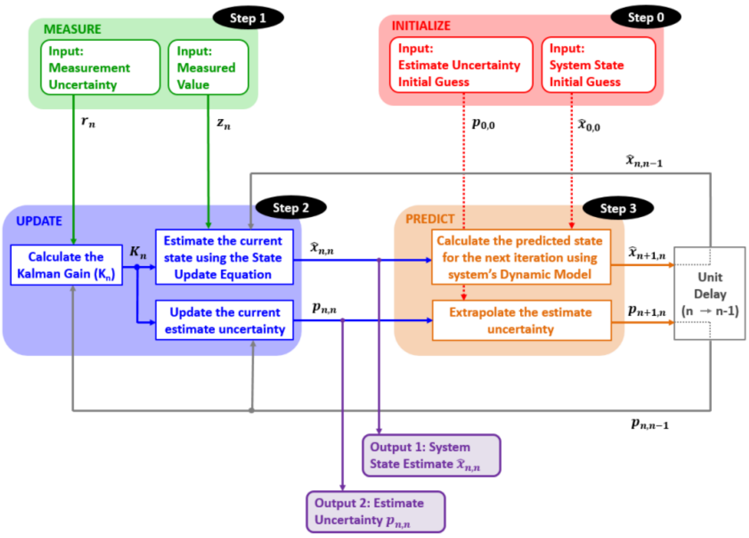
- State Update Equation - estimating the current state based on the known past estimation and present measurement.

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n(z_n - \hat{x}_{n,n-1})$$

- Covariance Update Equation - the measure of uncertainty in our estimation. Besides, the estimate uncertainty is constantly decreasing with each filter iteration since $(1 - K_n) \leq 1$

$$p_{n,n} = (1 - K_n)p_{n,n-1}$$

Bellow is the detailed description of the Kalman Filter algorithm that will be better understand with the examples later.



2.2 Kalman Gain intuition

The Kalman Gain tells us how much the measurement changes the estimate. That is, the Kalman Gain (K_n) is the measurement weight, and the $(1 - K_n)$ term is the weight of the current state estimate.

$$\hat{x}_{n,n} = (1 - K_n)\hat{x}_{n,n-1} + K_n z_n$$

- When the Kalman Gain is close to zero then the measurement uncertainty is high and the estimate uncertainty is low. Hence we give a significant weight to the estimate and a small weight to the measurement.
- When the Kalman Gain is close to one then the measurement uncertainty is low and the estimate uncertainty is high. Hence we give a low weight to the estimate and a significant weight to the measurement.
- When the measurement uncertainty equals the estimate uncertainty, then the Kalman gain equals 0.5.

2.3 Adding process noise

The uncertainty of the dynamic model is the Process Noise (it produces estimation errors) and its variance is denoted by q. Therefore, all the equation remain the same except for the covariance extrapolation equation that shall include it:

$$p_{n+1,n} = p_{n,n} + q_n \qquad p^v_{n+1,n} = p^v_{n,n} + q_n$$

2.4 Applied examples

Here we have some examples using the uni-variate Kalman Filter for estimating the height of a building, the temperature of the liquid in a tank or the temperature of a heating liquid. This 4 examples can be found by clicking here.

3 Multivariate Kalman Filter

For the Multivariate Kalman Filter the output is a multivariate random variable and has a covariance matrix that describes the squared uncertainty of it. The uncertainty variables of the multivariate Kalman Filter are:

$P_{n,n}$ is a covariance matrix that describes the squared uncertainty of an estimate

$P_{n+1,n}$ is a covariance matrix that describes the squared uncertainty of a prediction

R_n is a covariance matrix that describes the squared measurement uncertainty

Q is a covariance matrix that describes the process noise

3.1 State Extrapolation Equation

It predicts the next system state based on the knowledge of the current state, it extrapolates the state vector from the present (time step n) to the future (time step n + 1). Besides, describes the model of the dynamic system.

$$\hat{x}_{n+1,n} = F\hat{x}_{n,n} + Gu_n + w_n$$

Where:

$\hat{x}_{n+1,n}$ is a predicted system state vector at time step $n + 1$, dimension n_x x 1

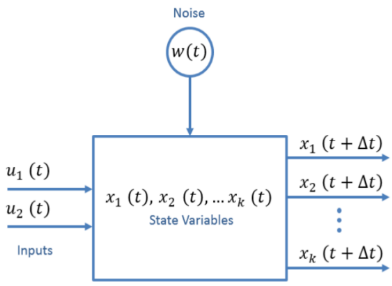
$\hat{x}_{n,n}$ is an estimated system state vector at time step n, dimension n_x x 1

u_n is a control variable or input variable - a measurable (deterministic) input to the system, dimension n_u x 1

w_n is a process noise or disturbance - an unmeasurable input that affects the state, dimension n_x x 1

F is a state transition matrix, dimension n_x x n_x

G is a control matrix or input transition matrix (mapping control to state variables), dimension n_x x n_u



The state variables may represent attributes of the system that we wish to know. The process noise w_n does not typically appear directly in the equations of interest. Instead, this term is used to model the uncertainty in the Covariance Extrapolation Equation.

3.1.1 Linear time-invariant systems

The Linear Kalman Filter assumes the Linear Time-Invariant (LTI) system model. That is, linear systems are described by systems of equations in which the variables are never multiplied with each other but only with constants and then summed up. Besides, a time-invariant system has a system function that is not a direct function of time, that is, although the system's output could change with time, the system function is not time-dependent.

3.2 Covariance Extrapolation Equation

$$P_{n+1,n} = FP_{n,n}F^T + Q$$

Where:

$P_{n+1,n}$ is the squared uncertainty of a prediction (covariance matrix) for the next state

$P_{n,n}$ is the squared uncertainty of an estimate (covariance matrix) of the current state

Q is a process noise matrix

F is a state transition matrix

For the derivation of this equation we use $COV(x) = E((x - \mu_x)(x - \mu_x)^T)$. Therefore,

$$P_{n+1,n} = E((\hat{x}_{n+1,n} - \mu_{x_{n+1,n}})(\hat{x}_{n+1,n} - \mu_{x_{n+1,n}})^T)$$

Then we replace in this equation the value of $\hat{x}_{n+1,n}$ from the state extrapolation equation we get

$$P_{n+1,n} = E((F\hat{x}_{n,n} + G\hat{u}_{n,n} - F\mu_{x_{n,n}} - G\hat{u}_{n,n})(F\hat{x}_{n,n} + G\hat{u}_{n,n} - F\mu_{x_{n,n}} - G\hat{u}_{n,n})^T)$$

$$P_{n+1,n} = E(F(\hat{x}_{n,n} - \mu_{x_{n,n}})(F(\hat{x}_{n,n} - \mu_{x_{n,n}}))^T)$$

Using the matrix transpose property: $(AB)^T = B^T A^T$

$$P_{n+1,n} = E(F(\hat{x}_{n,n} - \mu_{x_{n,n}})(\hat{x}_{n,n} - \mu_{x_{n,n}})^T F^T)$$

$$P_{n+1,n} = FE((\hat{x}_{n,n} - \mu_{x_{n,n}})(\hat{x}_{n,n} - \mu_{x_{n,n}})^T)F^T$$

$$P_{n+1,n} = FP_{n,n}F^T$$

3.2.1 Constructing the process noise matrix Q

We have seen in the examples and in the theory above that the process noise variance has a critical influence on the Kalman Filter performance. Too small q causes a lag error. If the q value is too high, the Kalman Filter follows the measurements and produces noisy estimations.

The process noise can be independent between different state variables. In this case, the process noise covariance matrix Q is a diagonal matrix:

$$Q = \begin{bmatrix} q_{11} & 0 & \cdots & 0 \\ 0 & q_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_{kk} \end{bmatrix}$$

The process noise can also be dependent. For example, the constant velocity model assumes zero acceleration ($a = 0$). However, a random variance in acceleration σ_a^2 causes a variance in velocity and position. In this case, the process noise is correlated with the state variables. Two models:

- Discrete noise model
  - Assumes that the noise is different at each period but is constant between periods. Better to use it when Δt is very small.
  - For the constant velocity model and using:

$$\begin{aligned} V(v) = \sigma_v^2 = E\left(v^2\right) - \mu_v^2 = E\left((a\Delta t)^2\right) - (\mu_a \Delta t)^2 \\ = \Delta t^2 \left(E\left(a^2\right) - \mu_a^2\right) = \Delta t^2 \sigma_a^2 \end{aligned}$$

$$V(x) = \sigma_x^2 = E\left(x^2\right) - \mu_x^2 = E\left(\left(\frac{1}{2} a \Delta t^2\right)^2\right) - \left(\frac{1}{2}\mu_a \Delta t^2\right)^2$$

$$= \frac{\Delta t^4}{4}\left(E\left(a^2\right) - \mu_a^2\right) = \frac{\Delta t^4}{4}\sigma_a^2$$

$$COV(x,v) = COV(v,x) = E(xv) - \mu_x \mu_v$$
$$= E\left(\frac{1}{2} a \Delta t^2 a \Delta t\right) - \left(\frac{1}{2}\mu_a \Delta t^2 \mu_a \Delta t\right) = \frac{\Delta t^3}{2}\left(E\left(a^2\right) - \mu_a^2\right) = \frac{\Delta t^3}{2}\sigma_a^2$$

$$Q = \begin{bmatrix} V(x) & COV(x,v) \\ COV(v,x) & V(v) \end{bmatrix} = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}$$

  - There are two methods for a faster construction of the Q matrix:
    * Projection using the state transition matrix. If the dynamic model does not include a control input: $Q = FQ_aF^T$, being:

$$Q_a = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \sigma_a^2 \qquad F = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

So

$$Q = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix} \sigma_a^2$$

* Projection using the control matrix. If the dynamic model includes a control input. $Q = G\sigma_a^2 G^T$.

- Continuous noise model
  - Assumes that the noise changes continuously over time. Better to use it when Δt is high.
  - We need to integrate the discrete process noise covariance matrix Q over time.

$$Q_c = \int_0^{\Delta t} Q\,dt \quad = \quad \sigma_a^2 \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} \end{bmatrix}$$

3.3 Measurement equation

The measurement value z_n represents a true system state in addition to the random measurement noise v_n, caused by the measurement device. The measurement noise variance r_n can be constant or different for each measurement.

$$z_n = Hx_n + v_n$$

Where:

z_n is a measurement vector, dimension n_z x 1

x_n is a true system state (hidden state), dimension n_x x 1

v_n is a random noise vector, dimension n_z x 1

H is an observation matrix, dimension n_z x n_x

3.3.1 The observation matrix

In many cases, the measured value is not the desired system state. There is a need for a transformation of the system state (input) to the measurement (output). Moreover, the purpose of the observation matrix H is to convert the system state into outputs using linear transformations. For example for scaling, when we have to do state selection or when we need a combination of states.

3.4 Covariance Equations

The terms w and v, which correspond to the process and measurement noise, do not typically appear directly in the calculations since they are unknown, instead they are used to model the uncertainty (or noise) in the equations themselves. All covariance equations are covariance matrices in the form of: $E(ee^T)$

3.4.1 Measurement uncertainty

$$R_n = E(v_n v_n^T)$$

Where:

R_n is the covariance matrix of the measurement

v_n is the measurement error

3.4.2 Process noise uncertainty

$$Q_n = E(w_n w_n^T)$$

Where:

Q_n is the covariance matrix of the process noise

w_n is the process noise

3.4.3 Estimation uncertainty

$$P_{n,n} = E(e_n e_n^T) = E((x_n - \hat{x}_{n,n})(x_n - \hat{x}_{n,n})^T)$$

Where:

$P_{n,n}$ is the covariance matrix of the estimation error

e_n is the estimation error

x_n is the true system state (hidden state)

$\hat{x}_{n,n}$ is the estimated system state vector at time step n

3.5 State Update Equation

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

Where:

$\hat{x}_{n,n}$ is an estimated system state vector at time step n, dimension n_x x 1

$\hat{x}_{n,n-1}$ is a predicted system state vector at time step n - 1, dimension n_x x 1

K_n is a Kalman Gain, dimension n_x x n_z

z_n is a measurement, dimension n_z x 1

H is an observation matrix, dimension n_z x n_x

3.6 Covariance Update Equation

$$P_{n,n} = (I - K_n H)P_{n,n-1}(I - K_n H)^T + K_n R_n K_n^T$$

Where:

$P_{n,n}$ is the covariance matrix of the current state estimation

$P_{n,n-1}$ is the prior estimate covariance matrix of the current state (predicted at the previous state)

K_n is a Kalman Gain

H is the observation matrix

R_n is the measurement noise covariance matrix

I is an Identity Matrix n x n

For the derivation of the equation we use:

	Equation	Notes
1	$\hat{\boldsymbol{x}}_{n,n} = \hat{\boldsymbol{x}}_{n,n-1} + \boldsymbol{K}_n(\boldsymbol{z}_n - \boldsymbol{H}\hat{\boldsymbol{x}}_{n,n-1})$	State Update Equation
2	$\boldsymbol{z}_n = \boldsymbol{H}\boldsymbol{x}_n + \boldsymbol{v}_n$	Measurement Equation
3	$\boldsymbol{P}_{n,n} = E\left(\boldsymbol{e}_n\boldsymbol{e}_n^T\right)$ $= E\left(\left(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n,n}\right)\left(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n,n}\right)^T\right)$	Estimate Covariance
4	$\boldsymbol{R}_n = E\left(\boldsymbol{v}_n\boldsymbol{v}_n^T\right)$	Measurement Covariance

It could be more simplified:

$$P_{n,n} = (I - K_nH)P_{n,n-1}$$

To derive this simplified form of the Covariance Update Equation, plug the Kalman Gain Equation into the Covariance Update Equation. Although this equation is numerically unstable. That is, a minor error in computing the Kalman Gain (due to round-off) can lead to huge computation errors. Also, the subtraction $(I - K_nH)$ can lead to non symmetric matrices due to floating-point error.

3.7 The Kalman Gain

$$K_n = P_{n,n-1}H^T(HP_{n,n-1}H^T + R_n)^{-1}$$

Where:

K_n is the Kalman Gain

$P_{n,n-1}$ is the prior estimate covariance matrix of the current state (predicted at the previous step)

H is the observation matrix

R_n is the measurement noise covariance matrix

For the derivation of the equation we need to rearrange the Covariance Update Equation to:

$$P_{n,n} = P_{n,n-1} - P_{n,n-1}H^TK_n^T - K_nHP_{n,n-1} + K_n(HP_{n,n-1}H^T + R_n)K_n^T$$

Then as The Kalman Filter is an optimal filter we need to seek a Kalman Gain that minimizes the estimate variance. To minimize the estimate variance, we need to minimize the main diagonal of the covariance matrix $P_{n,n}$, we need to minimize $tr(P_{n,n})$. So, we differentiate $tr(P_{n,n})$ with respect to K_n and set the result to zero.

3.8 Applied examples

Here we have some examples using the multivariate Kalman Filter for estimating the vehicle location and/or velocity and acceleration or a rocket altitude. This 3 examples extensively can be found by clicking here.

4 Summary

The Kalman Filter operates in a "predict-correct" loop.

Once initialized, the Kalman Filter predicts the system state at the next step. It also provides the uncertainty of the prediction.

Once the measurement is received, the Kalman Filter updates (or corrects) the prediction and the uncertainty of the current state. As well the Kalman Filter predicts the following states, and so on. The following diagram provides a complete picture of the Kalman Filter operation.