KALMAN FILTER OVERVIEW:

Consists of two phases

- Time Update (“predict”)
- Measurement Update (“correct”)

1) Time Update ("predict"):

$$x_{k+1} = Ax + Bu$$

$$P_{k+1} = AP_kA^T + Q$$

$$z = Hx$$

x : a vector (n by 1 matrix); each component represents a ***variable*** in the system.
Eg., x = [position, velocity, acceleration]

A: an [n by n] matrix, the state matrix (AKA the update matrix)
(where n is the number of ***variables*** in our system.)

B and u: control input.
Represent the change to x that we cause. Most of the time, it can be assumed to be 0

P: The state covariance matrix, represents the relationship between each ***variable***
Eg: Suppose x [temperature, change_in_temp] => P = {[a, 0], [0, b]}
Where a and b are the variance of temperature and change_in_temp, respectively
P[0][[1] = 0 and P[1][1] = 0 because there is no relationship between "temperature" and "change_in_temp"

Q: process noise covariance matrix (the filter's noise), describes the error in P
H: the observation matrix

2) Measurement Update("correct"):

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

$$x_{k+1} = x_k + K(z_{k+1} - Hx_k)$$

$$P_{k+1} = (I - K_k H) P_k$$

K: the Kalman Gain

R: Measurement noise covariance matrix(relates to the sensors' noise)
==> Represents the relationship betweem multiple sensors' noise
==> If there is only one sensor, ==> its just R(k) = [v]. (v is variance of the one sensor)

Formulating the Filter

- Vector **x** = $\{x, y, v_x, v_y, a_x, a_y\}$

$x = x_0 + v_x\,dt$

$y = y_0 + v_y\,dt$

$v_x = v_{x0} + a_x\,dt$

$v_y = v_{y0} + a_y\,dt$

$a_x = \alpha$ (const)

$a_y = \beta$ (const)

- B and u can be 0 (no control input)

- The state matrix **A:**

(the coefficients of each variable in vector x)

$$A = \begin{vmatrix} 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta \end{vmatrix}$$

- The state covariance matrix P

We can assume P_0 is an [n x n] matrix of all zeroes

$$P = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

- The observation matrix H

This tells us what sensor readings we'd get if x were the true state (if the sensor were perfect).

Here, the measurements/inputs only come from one source ('one sensor').

⇒ H is a [1 x n] matrix

⇒ (Suppose x" is the true value,

⇒ x_k(the input) = x"

$$H = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Notes: "In STAT, the hat matrix H (aka projections matrix) maps the vector of observed values to the vector of fitted values"

Measurement noise Matrix R

- R = [n by n] matrix, where n is number of sensors in use.

- R is similar to the identity matrix in that most entries are 0, except that the diagonal contains the noise variance of each sensor

- As stated, inputs come from only one source (one ‘sensor’)

  => R =[variance]

Notes: " Kalman filter can 'fuse' multiple sensor readings together, taking advantages of their individual strength, while gives readings with a balance of noise cancellaion and adaptability."

Program’s skeleton

- Input: a list of tuples (x, y, t)
- Output: the corrected tuples

- Code hosted at:
- https://launchpad.net/kalmanfilterimpl

- TODO:
  - Calculate the acceleration
  - Determine Q matrix (process noise covariance matrix)

Reference

- http://wiki.udacity.com/CS373%20Kalman%20Filter%20Matrices#Part_I_-_Who_is_Who_in_the_Land_of_Kalman_Filters
- http://biosport.ucdavis.edu/lab-meetings/KalmanFilterPresentation
- http://www.mathworks.com/help/toolbox/simulink/ug/bszo62g.html#bspqkec-3
- http://en.wikipedia.org/wiki/Kalman_filter
- http://en.wikipedia.org/wiki/Hat_matrix

Q Process noise covariance matrix

- $Q = F R F^T$
- http://biosport.ucdavis.edu/lab-meetings/KalmanFilterPresentation