Kalman Filter

— Extension of Wiener filter to non-stationary signals.

$$\begin{cases} x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} & \leftarrow \text{state model} \\ z_k = Hx_k + v_k & \leftarrow \text{observation model.} \end{cases}$$

x_k = underlying state.

z_k = observation.

w_k = process noise $\sim N(0, Q)$, $Q = Q(t)$

v_k = measurement noise $\sim N(0, R)$, $R = R(t)$

u_k = control input to the state.

B = relates input to the state — "input matrix"

A = dynamical model / update equation for state — "State Transition Matrix"

$H(t)$ = observation matrix.

Derivation:

$\hat{x}_k^- \in \mathbb{R} \rightarrow$ a priori state estimate at k, given knowledge of the process prior to k.

$\hat{x}_k \in \mathbb{R} \rightarrow$ a posteriori state estimate at k, given measurement at k - z_k.

a priori estimate error: $e_k^- = x_k - \hat{x}_k^-$ $\xrightarrow[\text{of error}]{\text{covariance}}$ $P_k^- = E[e_k^- e_k^{-T}]$

a posteriori estimate error: $e_k = x_k - \hat{x}_k$ $\qquad P_k = E[e_k e_k^T]$ (1)

Goal: Compute a posteriori state estimate $\hat{x}_k$ as a linear combination of a priori estimate $\hat{x}_k^-$, measurement z_k, and measurement prediction $H\hat{x}_k^-$.

$$\hat{x}_k = \hat{x}_k^- + \underset{\uparrow\ \text{gain/blending factor.}}{K}\underbrace{(z_k - H\hat{x}_k^-)}_{\text{innovation or residual}} \qquad (2)$$

(actual measurement − predicted measurement)

$H\hat{x}_k^-$ is the "predicted measurement"

Want to choose K to minimize a posteriori error covariance - P_k. This is equivalent of trying to make the error orthogonal to state.

Do this by subbing (2) into (1), taking expectation and then setting its derivative to 0, solve for K:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} = \frac{P_k^- H^T}{HP_k^- H^T + R} \qquad (3)$$

Consequences:

$\lim_{R_k \to 0} K_k = H^{-1} \rightarrow$ As R, the cov of measurement noise goes to 0, we trust the actual measurement z_k more, so $\hat{x}_k \rightarrow \frac{z_k}{H}$ ... meaning we have a high-fidelity observation model.

$\lim_{P_k^- \to 0} K_k = 0 \rightarrow$ As P_k^-, the cov of a priori estimate error goes to 0, that means we estimate $\hat{x}_k^-$ better, and thus trust the predicted measurement $H\hat{x}_k^-$ more. Therefore, we use the state estimate $\hat{x}_k^-$ to represent $\hat{x}_k$, ~~discard~~ disregarding measurement z_k.

Discrete Kalman Filter Algorithm

Every step, we update two things.

1) Time update – update the ~~curr~~ state estimate based on the <u>state model</u> $\Rightarrow \hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$

<u>a priori</u> } estimate error cov. $P_k^- = AP_{k-1}A^T + Q \leftarrow$ cov of process noise w_{k-1} from state update model.

2) Measurement update – Based on measurement taken, correct ~~the~~ or modify the ~~state estimate~~ <u>a priori state-estimate</u>, to achieve the <u>a posteriori state-estimate</u>.

<u>Kalman gain</u>: $K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$

<u>Incorporate measurement</u>: $\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \leftarrow$ this is essentially a proportional controller.

<u>Update a posteriori error cov</u>: $P_k = (I - K_k H)P_k^-$

Implementation is recursive – online updates the Kalman gain. ~~H fixed~~
H, A, and B are fixed, however.

Filter Parameters and tuning:

1) Measurement noise variance R – measured prior to process w/ off-line data. } Requires tuning.
2) Process noise variance Q – difficult to measure

when Q and R are constant, both P_k and K_k will stabilize quickly and then remain constant. Then these parameters maybe pre-computed offline.

However, Q_k and R_k most often will be a function of time.

Usage in BMI:

$\hat{x}_k$ = a posteriori estimate of kinematics (pos or vel)

A, B = state model; B may be 0.

z_k = neuron firing rates – measurement.

H = tuning model relating kinematics to firing rates.

P_k^- = covariance of kinematics error (can prob be estimated from actual x_k and $\hat{x}_k$)
a priori

R = Tuning model covariance.

The extended Kalman Filter (EKF)

When state model is not linear, linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates.

- Nonlinear state-model/stochastic diff eq:

$$\left.\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}) \\ z_k &= h(x_k, v_k) \end{aligned}\right\} \text{ Both } \underline{f} \text{ and } \underline{h} \text{ are nonlinear.}$$

Do not know v_k and w_k, approximate with Gaussian noise mean of 0:

$$\hat{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0)$$

$$\tilde{z}_k = h(\hat{x}_k, 0)$$

Note that the distributions $\hat{x}_k$ and $\hat{z}_k$ are no longer normal after nonlinear transformation.

EKF approximates the optimality of Baye's rule by linearization.

To approximate linearly, do Taylor series:

$$\begin{cases} x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + W w_{k-1} \\ z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + V v_k \end{cases}$$

x_k, z_k = actual state and measurement.

$\tilde{x}_k, \tilde{z}_k$ = approximate state and measurements

$\hat{x}_k$ = a posteriori estimate of state.

A = Jacobian of $\frac{\partial f}{\partial x} \rightarrow A[i,j] = \frac{\partial f[i]}{\partial x[j]}(\hat{x}_{k-1}, u_{k-1}, 0)$

W = Jacobian of $\frac{\partial f}{\partial w} \rightarrow W[i,j] = \frac{\partial f[i]}{\partial w[j]}(\hat{x}_{k-1}, u_{k-1}, 0)$

H = Jacobian of $\frac{\partial h}{\partial x} \rightarrow H[i,j] = \frac{\partial h[i]}{\partial x[j]}(\tilde{x}_{k-1}, u_{k-1}, 0)$

V = Jacobian of $\frac{\partial h}{\partial v} \rightarrow V[i,j] = \frac{\partial h[i]}{\partial v[j]}(\tilde{x}_k, 0)$.

---

prediction error:

$$\hat{e}_{x_k} = x_k - \tilde{x}_k \Rightarrow \hat{e}_{x_k} = A(x_{k-1} - \hat{x}_{k-1}) + \underbrace{WQW^T}_{\varepsilon_k}$$

measurement residual:

$$\hat{e}_{z_k} = z_k - \tilde{z}_k \Rightarrow \hat{e}_{z_k} = H\hat{e}_{x_k} + \underbrace{VRV^T}_{\eta_k}.$$

This new set can be seen as a second Kalman filter.
Use the 2nd Kalman filter to estimate prediction error $\hat{e}_{x_k} \rightarrow \hat{e}_k$.

Obtain a posterior ~~state~~ state estimate:

$$\begin{aligned} \hat{x}_k &= \tilde{x}_k + \hat{e}_k. \\ &= \tilde{x}_k + K_k \hat{e}_{z_k} \\ &= \tilde{x}_k + K_k(z_k - \tilde{z}_k) \end{aligned}$$

don't need 2nd KF after all to estimate.

Use to do measurement update.

EKF

1) Time update: $\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

$\left\{ \begin{array}{l} A_k \\ W_k \end{array} \right.$ are Jacobians.

2) Measurement update: $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0))$$

$$P_k = (I - K_k H_k) P_k^-$$

$\left\{ \begin{array}{l} H_k \\ V_k \end{array} \right.$ are Jacobians.