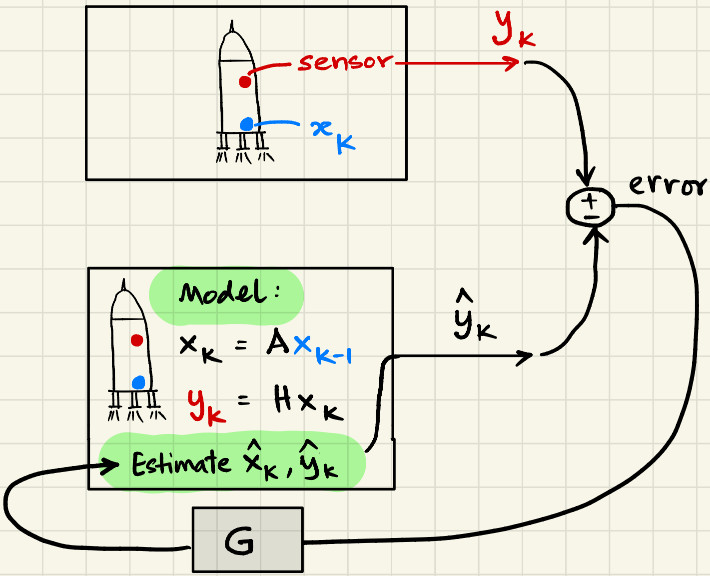
Kalman Filter

Problem Set-up :



Reality :

$$x_k = A x_{k-1} + w_p \quad \text{— ① Process model.}$$

$$y_k = H x_k + n_m \quad \text{— ② Measurement model.}$$

State estimator : Combine process & measurement

$$\hat{x}_k = x_k^p + K_k \left(y_k - \hat{y}_k \right)$$

from process model

from measurement model

If I have x_{k-1}, then I can model both

x_k^p and $\hat{y}_k = H x_k^p$

- What do we need to do this ?
  - ↳ Estimate K_k → time index
    - ↳ If process noise $n_p = 0$,
      K_k should be 0
    - ↳ if measurement noise $n_m = 0$
      K_k should be H^{-1}

      because, $\hat{x}_k = x_k^p + H^{-1}(Hx_k + n_m - Hx_k^p)$

      $= x_k + H^{-1} n_m \cong x_k$

- Main intuition : Seems like possible to use K_k as a knob that combines the process and measurement.

- If I turn K_k the wrong way, the $\hat{x}_k$ prediction should diverge from the true x_k which will also manifest in gap in true and modeled measurement. $(y_k - \hat{y}_k)$

$x_1^p \quad \hat{x}_0$

$$x_K = A x_{K-1} + n_p$$

$$y_K = H x_K + n_m$$

434: Kalman Filters

1. Start with an initial $\hat{x}_0$
2. That gives us a process based estimate x_1^p
3. If this was correct, then I expect the measurement y_1 to match with my modeled measurement Hx_1^p
   If it does not match, then the error is composed of both process error and measurement error.
4. My goal is to modulate the process estimate x_1^p with some linear function of this error
   This gives me an estimate of the state variable as: $\hat{x}_1 = x_1^p + K_1(y_1 - Hx_1^p)$
5. What should K_1 be?
   - Well, design it such that it minimizes the MSE of state estimate error, defined as : $e_1 = x_1 - \hat{x}_1$
   - So we want to minimize $E[e_1^2]$
6. Let's model e_1 … $e_1 = x_1 - \left(x_1^p + K_1(y_1 - Hx_1^p)\right) \quad = x_1 - \left(x_1^p + K_1(Hx_1 + n_m - Hx_1^p)\right)$

   $$= (1 - K_1H)x_1 - (1 - K_1H)x_1^p + K_1 n_m$$

   $$= (1 - K_1H)\underbrace{(x_1 - x_1^p)}_{\text{process error}} + K_1 n_m \quad e_1 = (1 - K_1H)e_1^p + K_1 n_m$$

   Not surprising that this error has both un-modeled components → the process error e_1^p and the measurement error n_m $\quad x_1 - x_1^p = e_1^p$
7. For MSE, we compute $E[e_1 e_1^T]$ … since e_1 can be a vector … or you can stack up $[e_1, e_2, e_3 ...]$ to make a vector

   - This expectation then becomes: $P_1 = E[e_1 e_1^T] = (1 - K_1H)\underbrace{(x_1 - x_1^p)}_{e_1^p}\underbrace{(x_1 - x_1^p)^T}_{e_1^{pT}}(1 - K_1H)^T + K_1 n_m n_m^T K_1^T$

   $(K_1 n_m)(K_1 n_m)^T$
   $= K_1 n_m n_m^T K_1^T$
   $= K_1 R_m K_1^T$

   Cross terms aren't present because e_1^p and n_m are uncorrelated. Why? Because the process and measurement errors are independent
   Thus,

   $$E[e_1 e_1^T] = (1 - K_1H)P_1^p(1 - K_1H)^T + K_1 R_m K_1^T$$

   Not surprising that this error covariance has the process error covariance P_1^p and the measurement error covariance R_m
8. Find K_1 that minimizes this covariance, i.e., $\arg\min_K \; (1 - K_1H)P_1^p(1 - K_1H)^T + K_1 R_m K_1^T$

   - This gives: $$K_1 = \frac{P_1^p H^T}{(HP_1^pH^T + R_m)}$$ [See equations 11.21 to 11.24 in this article for minimization]
9. Perhaps we can assume we know covariance for the measurement error R_m … but we don't have P_1^p

   Since $K_1 = f(P_1^p)$ … we need P_1^p
10. Ok, so let's define $P_1^p = E[e_1^p e_1^{pT}]$ where $e_1^p = x_1 - x_1^p \quad = Ax_0 + n_p - A\hat{x}_0 \quad = Ae_0 + n_p$

    So, $P_1^p = E[(Ae_0 + n_p)(Ae_0 + n_p)^T] \quad = AP_0A^T + R_p$ … again, assuming e_0 and n_p are uncorrelated.

    $E[Ae_0e_0^TA^T + n_pn_p^T + Ae_0n_p^T + n_pe_0^TA^T]$

    So we have $P_1^p = AP_0A^T + R_p$

    $E[Ae_0(Ae_0)^T] = A\,E[e_0e_0^T]A^T$
    $= AP_0A^T$

    Of course, this P_1^p depends on the previous P_0
    Assuming we have resolved the previous states well, we now have everything we need for K_1
11. Now, let's resolve this P_0 from last step … which is the same as resolving P_1 since it would be used in the next step.

    $P_1 = f(K_1, P_1^p)$ … so we have resolved both K_1 and P_1 now

    In fact, plugging K_1 into P_1 …. and then simplifying, we get: $P_1 = (1 - K_1H)P_1^p$

    ▼ Derivation

$$P_1 = (1 - K_1 H) P_1^P (1 - K_1 H)^T + K_1 R_m K_1^T$$

$$= \left(P_1^P - K_1 H P_1^P\right)\left(1 - H^T K_1^T\right)$$

$$= P_1^P - P_1^P H^T K_1^T - K_1 H P_1^P + K_1\left(H P_1^P H^T + R_m\right) K_1^T$$

Substitute $K_1 = \dfrac{P_1^P H^T}{H P_1^P H^T + R_m} = \dfrac{P_1^P H^T}{D}$

$$= P_1^P - P_1^P H^T \left(\frac{P_1^P H^T}{D}\right)^T - \cancel{\left(\frac{P_1^P H^T}{D}\right) H P_1^P} + \cancel{\frac{P_1^P H^T \cdot H P_1^P}{D}}$$

$$= P_1^P - P_1^P H^T \left(H P_1^P H^T + R_m\right)^{-1} H P_1^P$$

$$= P_1^P - K_1 H P_1^P = (1 - K_1 H) P_1^P$$

Thus, $\boxed{P_1 = (1 - K_1 H) P_1^P}$

PREDICTION

$$x_k^p = A\hat{x}_{k-1}$$

$$P_k^p = A P_{k-1} A^T + R_p$$

$\hat{x}_0$ P_0

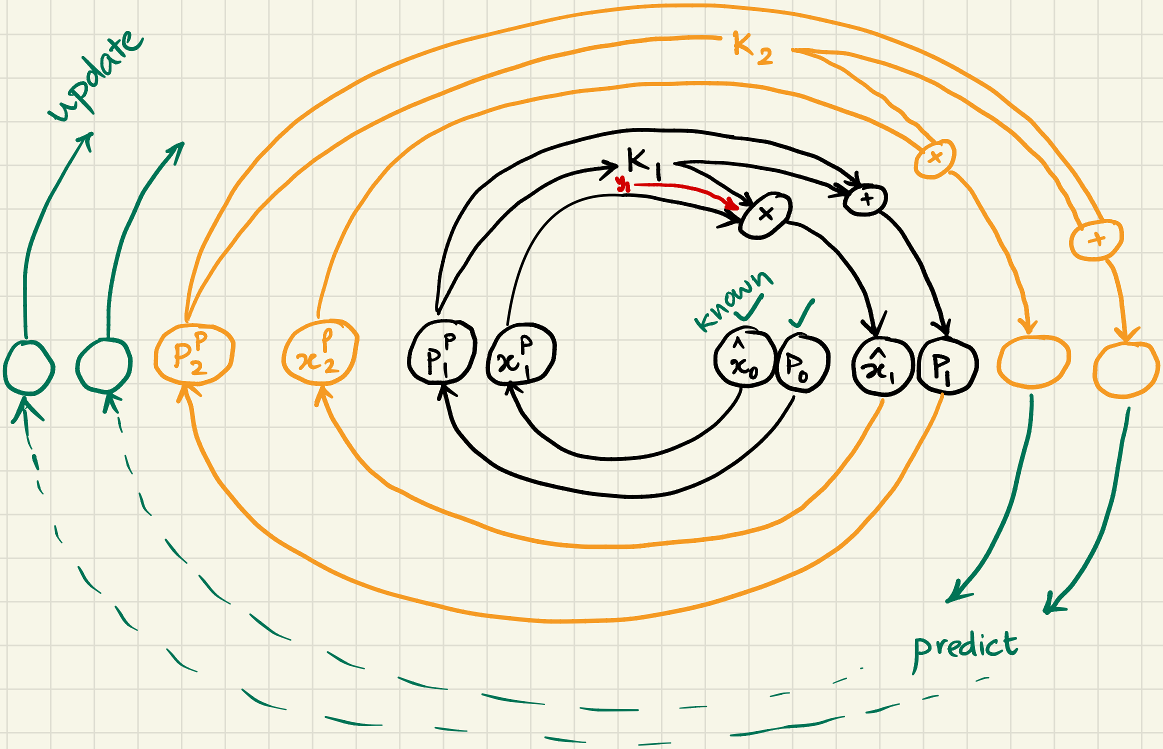
initialization

UPDATE

$$\hat{x}_k = x_k^p + K_k(y_k - H x_k^p)$$

$$K_k = \frac{P_k^p H^T}{H P_k^p H^T + R_m}$$

$$P_k = (1 - K_k H) P_k^p$$



update
K_2
K_1
y_1
known
P_2^p
x_2^p
P_1^p
x_1^p
$\hat{x}_0$
P_0
$\hat{x}_1$
P_1
predict

Questions