Last time: Kalman filter

$$x_{k+1} = Mx_k \quad , \quad k = 0, 1, 2, \ldots \quad , \quad x_0 \sim N(\mu_0, P_0)$$

$$y_{k+1} = Hx_{k+1} + \eta_k \quad , \quad \eta_k \sim N(0, R), \text{ iid}$$

1st step:

$$x_0 \sim N(\mu_0, P_0)$$

Forecast:

$$x_f = Mx_0 \sim N(\underbrace{M\mu_0}_{\mu_f}, \underbrace{MP_0M^T}_{P_f})$$

Analysis:

$$K = P_f H^T (HP_f H^T + R)^{-1}$$

$$x_a = \mu_f + K(y - H\mu_f)$$

$$P_a = (I - KH)P_f$$

Result:

$$x_1 | y_1 \sim N(x_a, P_a)$$

2nd step:

$$x_1 | y_1 \sim N(x_a, P_a)$$

Forecast

$$\underbrace{x_2 = Mx_1}_{x_2 | y_1} \sim N(\underbrace{Mx_a}_{\mu_f}, \underbrace{MP_a M^T}_{P_f})$$

Analysis:

$$K = P_f H^T (HP_f H^T + R)^{-1}$$

$$x_a = \mu_f + K(y_2 - H\mu_f)$$

$$P_a = (I - KH)P_f$$

Result:

$$x_2 | y_1, y_2$$

In general:

Given : $\underbrace{x_k \mid y_1, y_2 \dots y_k}_{y_{1:k}} \sim N(\mu_a, P_a)$

Forecast $\mu_f = M\mu_a \qquad P_f = M P_a M^T$

Analysis: $K = P_f H^T (H P_f H^T + R)^{-1}$

$\mu_a = \mu_f + K(y_{k+1} - H\mu_f)$

$P_a = (I - KH) P_f$.

Result : $x_{k+1} \mid y_{1:k+1} \sim N(\mu_a, P_a)$

$\rightsquigarrow$ Sequential updates to construct

$x_k \mid y_{1:k}$, for increasing k.

A closer look at the posterior covariance.

$P_{k+1} = (I - KH) P_f$

$P_f = M P_k M^T$

$K = P_f H^T (H P_f H^T + R)^{-1}$

Look for steady state $P_{k+1} = P_k = P$.

Call: $P_f = M P_k M^T = M P M^T = X$

$$P = (I - KH)X$$

$$P = X - XH^T(HXH^T+R)^{-1}HX$$

$$\underbrace{MPM^T}_{X} = MXM^T - MXH^T(HXH^T+R)^{-1}HXM^T$$

Algebraic Riccati eqn. (⇝ see Refs, book by Lancaster & Rodman)

Under some technical conditions (and const M, H, R)
Posterior covariance converges.

⇒ Kalman gain also converges.

• You can compute this offline.

<u>Some practical issues:</u>

<u>(1) Synthetic data / twin experiments</u>

· Right now, we don't have "real" data.

· Dealing with real data is <u>not</u> a good idea if you want to learn about an algorithm.

· We deal with data like and for now focus on data that are compatible with our model, because they are generated by the model.

Create synthetic data.

Model: $\quad x_{k+1} = M x_k \qquad$ + pick an IC.

Obs $\quad y_{k+1} = H x_k + \eta_k$

Iterate to obtain $\underbrace{(y_1, y_2, \dots, y_n)}_{\text{use these data for KF}}$

initialize KF: $\quad x_0 \sim \mathcal{N}(\mu_0, P_0)$

Iterate : for k in range ...

$$
\begin{aligned}
x_f &= M x \\
P_f &= M P M^T \\
&\vdots \\
\mu_a &= \mu_f + K(y[k] - H\mu_f) \\
P_a &= (I - KH) P_f
\end{aligned}
$$

Synthetic data or twin experiment.

In HW: Deliberately violate some assumptions and see what happens.

Examples:
- use different model params for generating synthetic data than what KF uses
- add noise that is not Gaussian, or has different variance.
- add unobserved variables!

(2) Discretization of Diff Eqns

$$\frac{dx}{dt} = f(x)$$

requires small time step.

Euler scheme: $x_{k+1} = x_k + \Delta t\, f(x_k)$

$\cancel{(I \pm \Delta t M)}\, x_{k+1} = \cancel{I}(I + M\Delta t)x_k$

implicit Euler scheme: $x_{k+1} = x_k + \Delta t\, f(x_{k+1})$

$(I - \Delta t M)\, x_{k+1} = x_k$

Runge Kutta 4: ↗ fourth order scheme, allows for bigger time steps.

$$K_1 = f(x_k) = M x_k$$

$$K_2 = f(x_k + \tfrac{\Delta t}{2} K_1)$$

$$K_3 = f(x_k + \tfrac{\Delta t}{2} K_2)$$

$$K_3 = f(x_k + \Delta t K_3)$$

$$x_{k+1} = x_k + \frac{\Delta t}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

how to write this as $M x_k$?

$$K_1 = f(x_k) = M x_k$$

$$K_2 = f(x_k + \tfrac{\Delta t}{2}K_1) = M(x_k + \tfrac{\Delta t}{2} M x_k) = M(I + \tfrac{\Delta t}{2}M)x_k$$

$$K_3 = f(x_k + \tfrac{\Delta t}{2}K_2) = M(x_k + \tfrac{\Delta t}{2} M(I + \tfrac{\Delta t}{2}M)x_k)$$

$$= M(I + \tfrac{\Delta t}{2}M(I + \tfrac{\Delta t}{2}M))x_k$$

$$K_4 = f(x_k + \tfrac{\Delta t}{2}K_3) = M(x_k + \tfrac{\Delta t}{2}M(I + \tfrac{\Delta t}{2}M(I + \tfrac{\Delta t}{2}M))x_k)$$

⇝ it gets more complicated but it's doable!

(3) What if observations are less frequent.

↳ Suppose you have ℓ model steps between observations.

$\underline{\ell=3}$

$$x_1 = Mx_0$$
$$x_2 = Mx_1 = MMx_0$$
$$\left.\begin{array}{l} x_3 = Mx_2 = M^3 x_0 \\ y_1 = Hx_3 + \eta. \end{array}\right\}$$

you can use KF, but the M you use changes and becomes M^ℓ where ℓ is the # of steps between obs.

(4) More sophisticated ODE solvers + multiple steps between obs.

→ you can construct the corresponding M, but it requires patience.

What is "bad" about KF?

- Computing M can be cumbersome.
- Restrictive setting! linear model: $x_{k+1} = Mx_k$, $y_{k+1} = Hx_k + \eta_k$, η_k Gaussian!
- What if matrices are big? ↳ Computations impossible!

→ KF is of limited use in practice.

How can I check if my KF "works"?

"works" = errors are small.

$$RMSE = \left(\frac{1}{n} \sum_{i=1}^{n} \left([x^t_k]_i - [x^a_k]_i \right)^2 \right)^{1/2}$$

advantage of twin experiment!

↳ Compute this error as fcn of time, it should be "small".

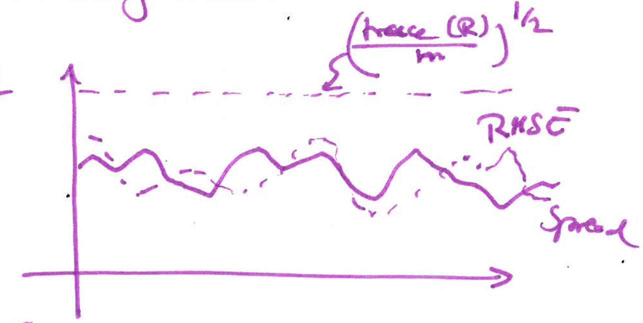
What is small?

$$\text{trace}\, P^a_k = \sum_{i=1}^{n} [P^a_k]_{ii} \rightarrow \text{compute at every time.}$$

"Spread"

Compare RMSE and $\left(\frac{\text{trace}\, P}{n} \right)^{1/2}$

These should be roughly equal



If you see this, then KF is "working".

We will use this criterion for other DA methods as well!

It states: average error ≈ predicted error