What is bad about the Kalman filter (Summary)

(1) Implementation gets more complicated if ODE solver is more complicated.

(2) Implementation gets more complicated if there are several steps between observations.

(3) Implementation becomes tricky when dimension is large.
Suppose state has $\sim 10^6$ components
↳ M is a $10^6 \times 10^6$ matrix with 10^{36} elements.
Difficult to store / compute w.k.

(4) This is limited to linear models, Gaussian noise, linear observation fctns.

Recall: KF

$$x_{k+1} = M x_k \qquad , \; x_k \sim N(\mu_k, P_k)$$
$$y_{k+1} = H x_{k+1} + \eta_{k+1}, \qquad \eta_{k+1} \sim N(0, R)$$

$$\mu_f = M \mu_k$$
$$P_f = M P_k M^T$$
$$K = P_f H^T (H P_f H^T + R)^{-1}$$
$$\mu_{k+1} = \mu_k + K(y_{k+1} - H\mu_k)$$
$$P_{k+1} = (I - KH) P_f.$$

Recall: Monte Carlo

$\{x^j\}$, $j = 1 \ldots N_e$, samples of r.v. w.th pdf p.

$$E[x] \simeq \frac{1}{N_e} \sum_{j=1}^{N_e} x^j = \bar{x}$$

$$\mathrm{Cov}(x) \simeq \frac{1}{N_e - 1} \sum (x^j - \bar{x})(x^j - \bar{x})^T$$

EnKF $\approx$ KF + Monte Carlo.
ensemble

Idea: Suppose you have N_e samples of

$$p(x^k \mid y^{1:k}) \quad , \quad \{X_k^j\} \quad j = 1, 2, 3, \ldots N_e$$

Define: $\quad x_f^j = H x_k^j$, Forecast ensemble

$$\mu_f \approx \frac{1}{N_e} \sum_{j=1}^{N_e} H x_k^j = \bar{\mu}_f = \frac{1}{N_e} \sum_{j=1}^{N} x_f^j$$

$$P_f \approx \frac{1}{N_e - 1} \sum_{j=1}^{N_e} (x_f^j - \bar{\mu}_f)(x_f^j - \bar{\mu}_f) = \bar{P}_f$$

Define $\bar{K} = \bar{P}_f H^T (H \bar{P}_f H^T + R)^{-1}$

Analysis ensemble: $\quad x_a^j = x_f^j + \bar{K}(y_{k+1} - H x_f^j)$

→ Use analysis ensemble for state-estimation!

→ Rather than tracking/updating posterior mean and covariance, we update an ensemble of states

→ Using the ensemble, we can compute mean and covariances.

What is the distribution of the analysis ensemble?

Example (scalar) $x_k \sim N(0,1)$

$x_{k+1} = x_k \qquad (M=1)$

$y_{k+1} = x_{k+1} + \eta, \quad \eta \sim N(0,1)$

KF:

$\mu_f = 0 \qquad H = 1 \qquad M = 1$

$P_f = 1 \qquad R = 1$

$K = 1 \cdot 1 (1+1)^{-1} = 1/2$

$\hookrightarrow \mu_{k+1} = \mu_f + \frac{1}{2}(y - H\mu_f) = \frac{1}{2} y$

$P_{k+1} = (1 - \frac{1}{2}) \cdot 1 = \frac{1}{2}$

EnKF: $\{x_k^j\}$, $j = 1 \dots N_e$ are samples from $N(0,1)$

Approximate! $\bar{\mu_f} \approx 0$ (for large ensembles

$\bar{P_f} \approx 1$ this is valid)

$\bar{K} = 1/2$

$x_a = x_j^j + \frac{1}{2}(y - x_j^j)$

$E[x_a] = 0 + \frac{1}{2}y - 0 = \frac{1}{2}y$ ✓

(in $N_e \to \infty$ limit)

$P_a = \mathrm{Cov}(x_a) = \mathrm{Cov}\left(\cancel{(I - KH)}(1 - \frac{1}{2}) x_j^j + \frac{1}{2} y\right)$

$= \frac{1}{4} 1 + 0 = 1/4 \neq 1/2$:(

→ Posterior covariance is underestimated!

Try to fix it

$$x_a^j = x_f^j + K(\tilde{y}^j - Hx_f^j)$$

$$\underbrace{\tilde{y}^j = y + \eta^j}_{\text{"Perturbed observation"}} \qquad \eta^j \sim N(0,R)$$

Back to example:

$$E[x_a^j] = 0 + \tfrac{1}{2}(E[\tilde{y}] - H\cdot 0) = \tfrac{1}{2}y \checkmark$$

(did not break this nice property)

$$\mathrm{Cov}(x_a^j) = \mathrm{Cov}(\overbrace{(I-KH)}^{1/2}x_f^j + K\tilde{y})$$

$$= \tfrac{1}{4}\cdot 1 + \tfrac{1}{4}\cdot \underbrace{\mathrm{Cov}(y+\eta^j)}_{\mathrm{Cov}(y)+\mathrm{Cov}(\eta^j) = 0+1 = 1}$$

$$= \frac{1}{4} + \frac{1}{4} = \frac{1}{2} \checkmark$$

In general:

$$x_f^j = M x_a \quad \text{(Forecast)}$$
$$\overline{x_f} = \frac{1}{N_e} \sum x_f^j \quad \text{(Forecast mean)}$$
$$\overline{P_f} = \frac{1}{N_e - 1} \sum_{j=1}^{N_e} (x_f^j - \overline{x_f})(x_f^j - \overline{x_f})^T \quad \text{(Forecast covariance)}$$

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

$$x_a^j = x_f^j + \overline{K} (\tilde{y}^j - H x_f^j) \qquad \tilde{y}^j = y + \eta^j , \ \eta^j \sim N(0, R)$$

We know (see review), that

$$\lim_{N_e \to \infty} \overline{x_f} = \mu_f \qquad \text{(we are still using linear model!)}$$

$$\lim_{N_e \to \infty} \overline{P_f} = P_f$$

We use $\overline{P_f} = P_f$ & $\overline{x_f} = \mu_f$ in below calculations.

↳ This means, that our results are only valid for $N_e \to \infty$, or at least for very large N_e.

↳ Small N_e is a practical requirement, we will address small N_e issues later on.

Question:

What is the distribution of $\{ x_a^j \}$.

1) It is Gaussian, because each x_a^j is a (weighted) sum of Gaussians, which we know is Gaussian

↳ What is the mean & covariance of analysis ensemble?

$$E[x_a^j] = \lim_{N_e \to \infty} \frac{1}{N_e} \sum_{j=1}^{N_e} \left(x_f^j + K(\tilde{y}^j - H x_f^j) \right)$$

$$= \lim_{N_e \to \infty} \frac{1}{N_e} \sum_{j=1}^{N_e} (I - KH) x_f^j + K \tilde{y}^j$$

$$= (I - KH) \underbrace{\lim_{N_e \to \infty} \frac{1}{N_e} \sum_{j=1}^{N_e} x_f^j}_{\mu_f} + K \underbrace{\lim_{N_e \to \infty} (y + \eta^j)}_{y}$$

$$= (I - KH)\mu_f + Ky = \mu_f + K(y - H\mu_f)$$

$\rightsquigarrow$ Same as KF!

$$\mathrm{Cov}(x_a^j) = \mathrm{Cov}\left((I - KH) x_f + K\tilde{y} \right)$$

$$= (I - KH) P_f (I - KH)^T + KRK^T$$

$$= (P_f - KHP_f)(I - KH)^T + KRK^T$$

$$= P_f - P_f H^T K^T - KHP_f + KHP_f H^T K^T + KRK^T$$

$$= (I - KH)P_f - P_f H^T K^T + \underbrace{K(HP_f H^T + R)K^T}_{P_f H^T \cancel{(HP_f H^T + R)^{-1}(HP_f H^T + R)} K^T \;=\; P_f H^T K^T}$$

$$= (I - KH)P_f$$

$\rightsquigarrow$ Same as KF.

Result: For linear Gaussian problems, EnKF produces an analysis ensemble whose distribution is Gaussian with mean μ_{k+1} and cov. P_{k+1}, same as KF.

Note: this implementation of EnKF is called "Stochastic EnKF", or perturbed obs. EnKF.

Remaining questions: (i) What about nonlinear models
(ii) What about small ensembles?

→ Wait! First another implementation of EnKF called "square root EnKF."

EnKF in Square root form

Basic idea: $\{x_f^j\}$, $j = 1 \dots N_e$

Forecast ensemble

→ Compute μ_f and P_f and

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

perturbed obs: $x_a^j = x_f^j + K(\tilde{y} - H x_f^j)$

perturbed obs.

How about this: $\mu_a = \mu_f + K(y - H\mu_f)$

→ Compute $P_a = (I - KH)P_f$

and generate an ensemble with mean μ_a and covariance P_a.

How? Easy!

$$x_a^j = \mu_f + P_a^{1/2} \xi \quad , \quad \xi \sim N(0, I)$$

$$P_a^{1/2} P_a^{1/2} = P_a$$

"matrix square root".

More on computation of matrix square root.

$$P_a = P_a^{1/2} P_a^{1/2} = (I - KH) P_f.$$

Define: $X_f = \frac{1}{\sqrt{N_e - 1}} [x_f^j - \mu_f]$

matrix ~~cols~~ whose colums are mean adjusted ensemble perturbations.

$$P_f = X_f X_f^T$$

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

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

drop subscript f.

$$P_a = P_a^{1/2} P_a^{1/2} = (I - KH) P_f = P_f - KHP_f$$

$$= XX^T - XX^T H^T (HXX^T H^T + R)^{-1} HXX^T$$

$$= X \left(I - X^T H^T (HXX^T H^T + R)^{-1} HX \right) X^T$$

$$V = X^T H^T$$

$$= X \underbrace{\left(I - V (V^T V + R)^{-1} V^T \right)}_{ZZ^T} X^T$$

$$= X Z Z^T X^T$$

$$\Rightarrow P_a^{1/2} = XZ$$

Or write in terms of "analysis perturbations".

$$P_a = \tilde{X}_a \tilde{X}_a^T$$

$$\tilde{X}_a = \frac{1}{\sqrt{N_e - 1}} \left[x_a^j - \mu_a \right]$$

$$P_a = \tilde{X}_a \tilde{X}_a^T \Rightarrow \tilde{X}_a = XZ.$$

Algorithm: (Square root EnKF)

$\{x_k^j\}$ ensemble at time k $(\sim p(x_k|y_{1:k}))$

Compute forecast ensemble: $x_f^j = M x_k^j$

forecast mean: $\mu_f = \frac{1}{N_e}\sum_{j=1}^{N} x_f^j$

$$X_f = \frac{1}{\sqrt{N_e-1}}\left[x_f^j - \mu_f\right]$$

(forecast perturbations)

Square root: $V = X^T H^T$

$$ZZ^T = I - V(R + V^T V)^{-1} V^T$$

$$\cancel{P_a^{1/2} = X_f Z}$$

$$P_f = X_f X_f^T$$

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

$$\tilde{X}_a = X_f Z$$

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

$$[x_a^j] = \mu_a + \sqrt{N_e-1}\,\tilde{x}_a^j$$

How to compute the square Root more effectively

$$I - V(V^T V + R)^{-1} V^T = (I + V R^{-1} V^T)^{-1}$$

Woodbury formula: $A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA = (A - BD^{-1}C)^{-1}$

Recall matrix square root:

$A = U \Lambda U^T$ $\qquad \Lambda = \text{diag}(\lambda_j)$ ← evals of A.

$U^T U = I \qquad U = [u_j]$ ← evecs of A

$= U \sqrt{\Lambda} \sqrt{\Lambda} U^T$

$= A^{1/2} (A^{1/2})^T$

Recall: λ is eval of A, $(1+\lambda)$ is eval of $A + I$

λ is eval of A, $1/\lambda$ is eval of A^{-1}.

↳ To compute evals / evecs of

$(I + V R^{-1} V^T)^{-1}$

1) Compute evecs and evals of $V R^{-1} V^T = U \Lambda U^T$

2) $(I + V R^{-1} V^T)^{-1} = (U U^T + U \Lambda U^T)^{-1} = (U(I + \Lambda)U^T)^{-1}$

$= U(I + \Lambda)^{-1} U^T$

3) $\underbrace{((I + V R^{-1} V^T)^{-1})^{1/2}}_{\text{square root}} = \underline{U(I + \Lambda)^{-0.5}}$