Kalman filters: implementation issues

Marco Lovera
Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano
marco.lovera@polimi.it

Outline

- Summary of Kalman filtering;
- Kalman filters: divergence from theoretical performance;
- Ill-conditioned Kalman filtering problems;
- Implementation issues:
  - Joseph form;
  - Scalar updates of the state estimate;
  - Factorisation methods;

Summary of Kalman filtering

The filtering problem

Given a dynamical system

$$x_k = f(x_{k-1}, w_{k-1})$$

and a measurement equation

$$z_k = g(x_k, v_k)$$

where w_k and v_k are *process* and *measurement* noise, respectively.

The filtering problem consists in estimating the state of the system at time k using measurements of z up to time k and the available mathematical model of the system.

The Kalman filtering problem

- Special case:
  - the system is linear (possibly time-varying);
  - process and measurement noise are white noise processes;

- The Kalman filter provides the *optimal* solution to the filtering problem, in the sense that it minimises the state estimation error variance.

System dynamic model

The system is linear, time-varying, in discrete-time:

$$x_k = \Phi_k x_{k-1} + w_{k-1}$$
$$z_k = H_k x_k + v_k$$

Noise assumptions:

$$w_k \simeq WN(0, Q_k), \quad v_k \simeq WN(0, R_k), \quad E[w_k v_j^T] = 0, \quad \forall k, j$$

Initial conditions:

$$E[x_0] = \hat{x}_0$$
$$E[x_0 x_0^T] = P_0$$

Special case: the deterministic problem

The system is linear, time-invariant, in discrete-time:

$$
\begin{aligned}
x_k &= \Phi x_{k-1} \\
z_k &= H x_k
\end{aligned}
$$

Then the state can be recostructed using

$$
\begin{aligned}
\hat{x}_k &= \Phi \hat{x}_{k-1} + K(z_{k-1} - \hat{z}_{k-1}) \\
\hat{z}_k &= H \hat{x}_k
\end{aligned}
$$

provided that K is suitably chosen:

$$
e_k = x_k - \hat{x}_k = \Phi(x_{k-1} - \hat{x}_{k-1}) - K(z_{k-1} - \hat{z}_{k-1}) = [\Phi - KH]\, e_{k-1}
$$

if K: (Φ-KH) asymptotically stable) e_k ! 0, k ! 1.
(always possible if (Φ,H) is observable)

Optimal linear filter

In the stochastic case, the question is: how to choose the gain *optimally* in order to minimise the variance of the state estimation error?

$$x_k = \Phi_k x_{k-1} + w_{k-1}$$
$$z_k = H_k x_k + v_k$$

$$\hat{x}_k = \Phi_k \hat{x}_{k-1} + K_k (z_k - \hat{z}_k)$$
$$\hat{z}_k = H_k \hat{x}_k$$

K_k: $E[(x_k - \hat{x}_k)^T (x_k - \hat{x}_k)]$ is minimised.

Summary of filter equations

State estimate and error covariance extrapolation:

$$\hat{x}_k(-) = \Phi_{k-1}\hat{x}_{k-1}(+)$$

$$P_k(-) = \Phi_{k-1}P_{k-1}(+)\Phi_{k-1}^T + Q_{k-1}$$

State estimate observational update and error covariance update:

$$\hat{x}_k(+) = \hat{x}_k(-) + \bar{K}_k\left(z_k - H_k\hat{x}_k(-)\right)$$

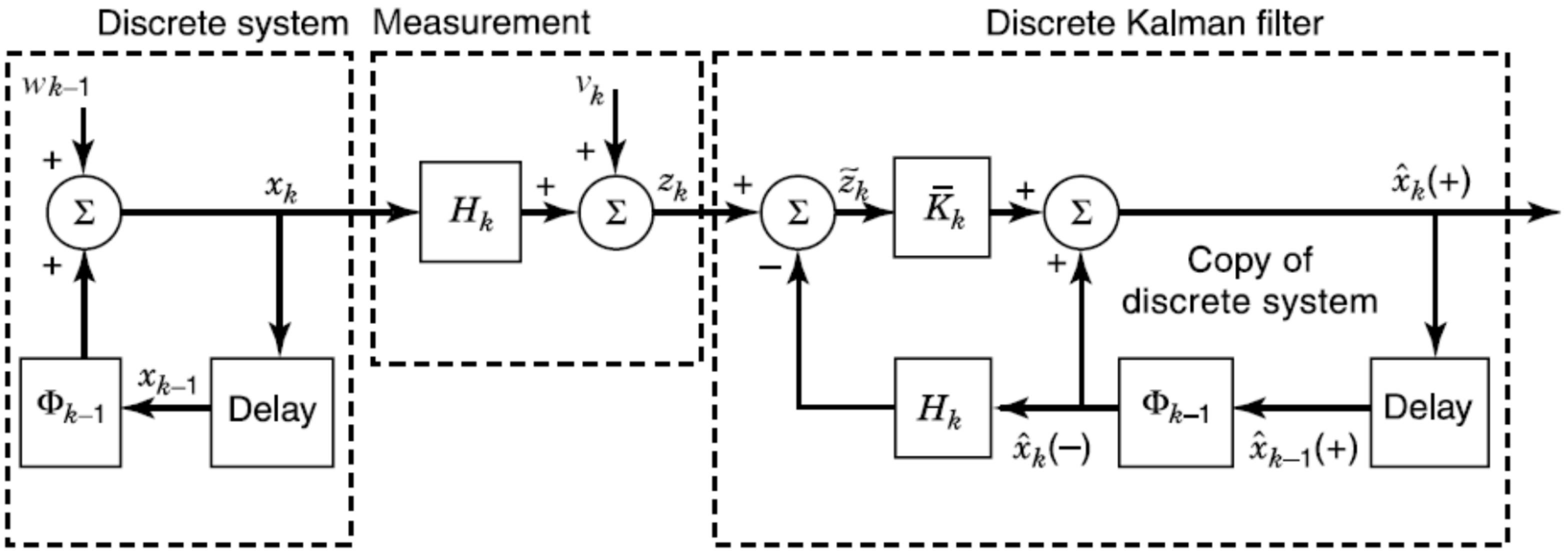
$$P_k(+) = \left[I - \bar{K}_kH_k\right]P_k(-)$$

Gain update:

$$\bar{K}_k = P_k(-)H_k^T\left[H_kP_k(-)H_k^T + R_k\right]^{-1}$$

Block diagram for the filter

The dynamics of the filter can be represented as



An example

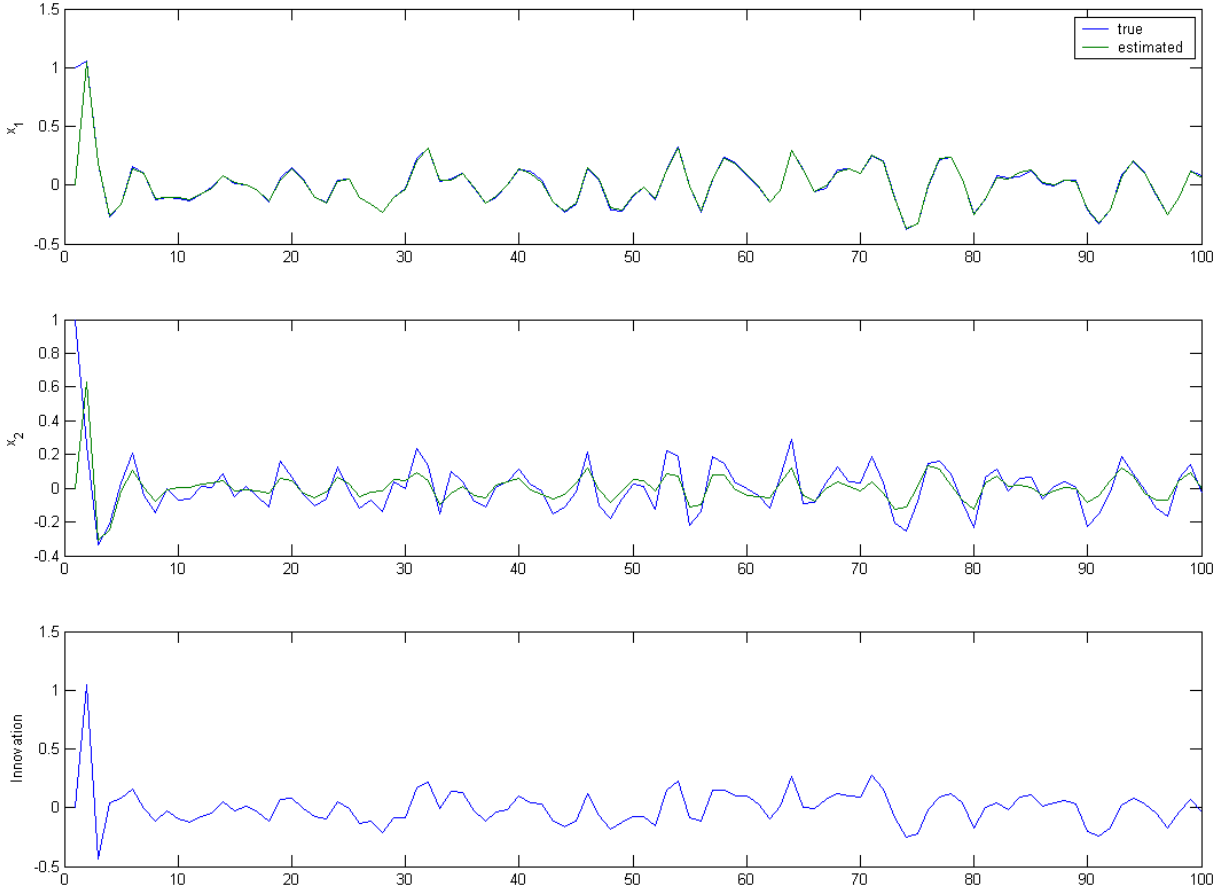
Consider the linear, discrete-time system given by

$$x_k = \Phi x_{k-1} + w_{k-1}$$
$$z_k = H x_k + v_k$$

$$\Phi = \begin{bmatrix} 0 & 1 \\ -0.4 & 0.6 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix},$$
$$Q = \sigma_w^2 I_2, \quad R = \sigma_v^2, \quad \sigma_w = 0.1, \quad \sigma_v = 0.01$$

An example (2)

Simulation results

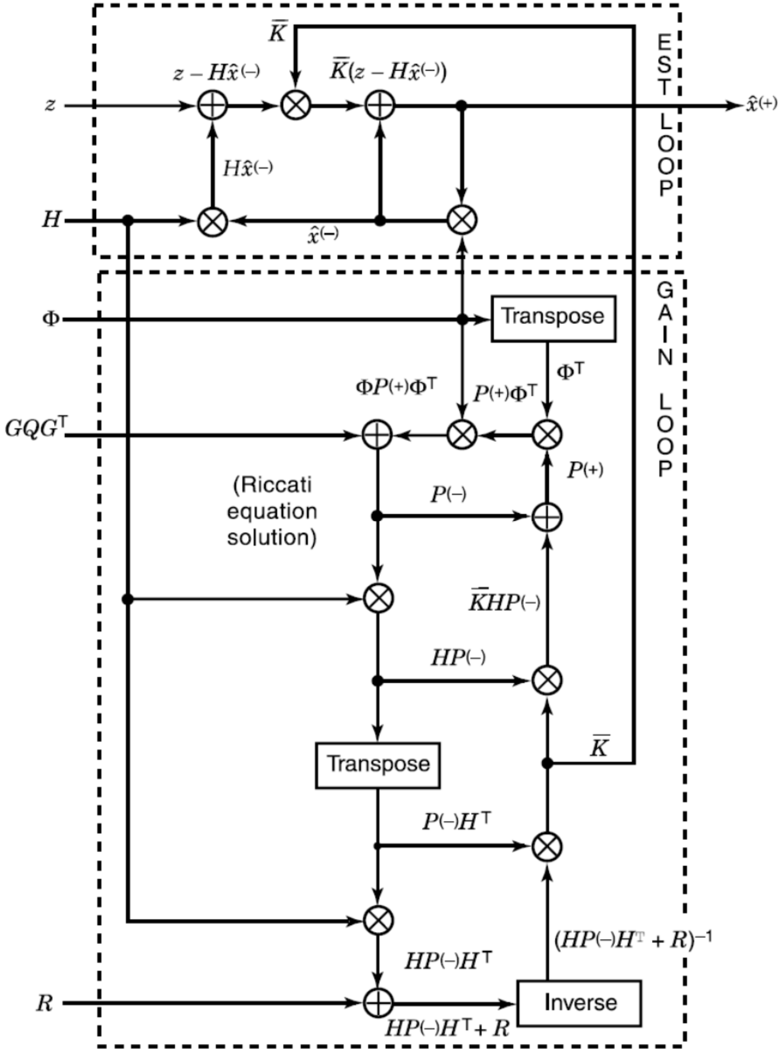


Ill-conditioned Kalman filtering problems

Ill-conditioning in Kalman filtering

- Uncertainty in the values of Φ, Q, H and R;
- Large ranges of values of parameters, state variables or measurements (poor scaling);
- Ill-conditioning of HPH^T+R for inversion;
- Ill-conditioned theoretical solution of the solution of the ARE;
- Large matrix dimensions;
- Poor machine precision.

Propagation of roundoff errors in KFs



Kalman filter data flow

Propagation of roundoff errors in KFs (1)

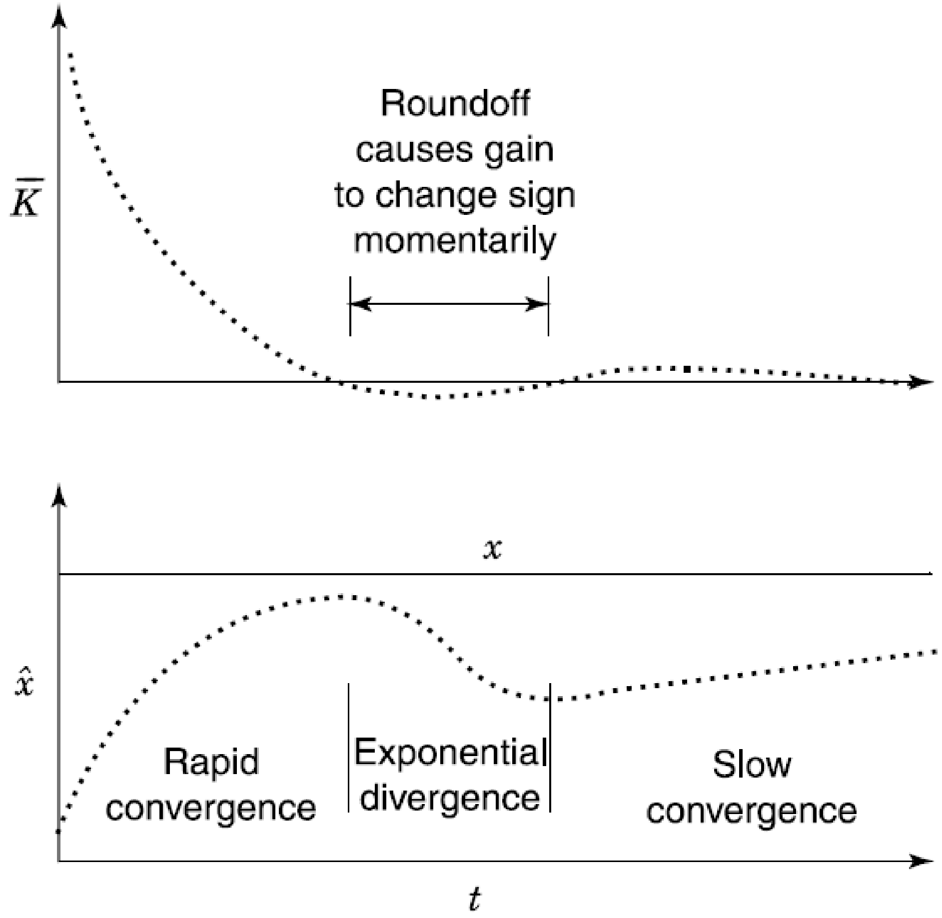
Comments:

- The estimation loop is a feedback loop)
  roundoff errors are corrected by feedback *as long as the gain is correct*.

- The gain loop has no feedback)
  - No way of detecting and correcting the effect of roundoff errors;
  - The gain loop also involved the largest number of roundoff-sensitive operations.

Propagation of roundoff errors in KFs (2)

Example: what happens if the sign of P changes?



Error propagation models (1)

Numerical analysis of error propagation

Roundoff Error in Filter Variable	Error Model (by Filter Type): Conventional Implementation	Error Model (by Filter Type): Square-Root Covariance
$\delta x_{k+1}(-)$	$A_1[\delta x_k(-) + \delta P_k(-) A_2 (z - H x_k(-))] + \Delta x_{k+1}$	
$\delta \overline{K}_k$	$A_1 \delta P_k(-)$	
$\delta P_{k+1}(-)$	$A_1 \delta P_k(-) A_1^T + \Delta P_{k+1}$ $+ \Phi(\delta P_k(-) - \delta P_k^T(-))\Phi^T$ $- \Phi(\delta P_k(-) - \delta P_k^T(-)) A_1^T$	$A_1 \delta P_k(-) A_1^T$ $+ \Delta P_{k+1}$

Notes: $A_1 = \Phi - \overline{K}_k H;\ A_2 = H^T[H P_k H^T + R]^{-1}$.

Error propagation models (2)

Theoretical upper bounds of propagation error

Norm of Roundoff Errors	Upper Bounds (by Filter Type)	
	Conventional Implementation	Square-Root Covariance
$\lvert\Delta x_{k+1}(-)\rvert$	$\varepsilon_1(\lvert A_1\rvert\lvert x_k(-)\rvert + \lvert\overline{K}_k\rvert\lvert z_k\rvert) + \lvert\Delta\overline{K}_k\rvert(\lvert H\rvert\lvert x_k(-)\rvert + \lvert z_k\rvert)$	$\varepsilon_4(\lvert A_1\rvert\lvert x_k(-)\rvert + \lvert\overline{K}_k\rvert\lvert z_k\rvert) + \lvert\Delta\overline{K}_k\rvert(\lvert H\rvert\lvert x_k(-)\rvert + \lvert z_k\rvert)$
$\lvert\Delta\overline{K}_k\rvert$	$\varepsilon_2\kappa^2(R^\star)\lvert\overline{K}_k\rvert$	$\varepsilon_5\kappa(R^\star)[\lambda_m^{-1}(R^\star)\lvert C_{P(\overline{K}+1)}\rvert + \lvert\overline{K}_k C_{R^\star}\rvert + \lvert A_3\rvert/\lambda_1(R^\star)]$
$\lvert\Delta P_{k+1}(-)\rvert$	$\varepsilon_3\kappa^2(R^\star)\lvert P_{k+1}(-)\rvert$	$\dfrac{\varepsilon_6[1+\kappa(R^\star)]\lvert P_{k+1}\rvert\lvert A_3\rvert}{\lvert C_{P(k+1)}\rvert}$

Notes: $\varepsilon_1, \ldots, \varepsilon_6$ are constant multiples of ε, the unit roundoff error; $A_1 = \Phi - \overline{K}_k H$; $A_3 = [(\overline{K}_k C_{R^\star})\lvert C_{P(k+1)}]$; $R^* = HP_k(-)H^T + R$; $R^\star = C_{R^\star} C_{R^\star}^T$ (triangular Cholesky decomposition); $P_{k+1}(-) = C_{P(k+1)} C_{P(k+1)}^T$ (triangular Cholesky decomposition); $\lambda_1(R^*) \geq \lambda_2(R^*) \geq \cdots \geq \lambda_m(R^*) \geq 0$ are the characteristic values of R^*; $\kappa(R^*) = \lambda_1(R^\star)/\lambda_m(R^\star)$ is the condition number of R^*.

Examples of filter divergence (1)

Consider the estimation problem with Φ=I, H=1, Q=0, l=n=1, in which $P_0 >> R$, in the sense that $R < \varepsilon P_0$. Then the iteration of the KF gives

Expression	Exact	Rounded
$P_0 H^T$	P_0	P_0
$H P_0 H^T$	P_0	P_0
$H P_0 H^T + R$	$P_0 + R$	P_0
$\bar{K}_1 = P_0 H^T (H P_0 H^T + R)^{-1}$	$P_0 (P_0 + R)^{-1}$	1
$P_1 = P_0 - \bar{K}_1 H P_0$	$P_0 - P_0 (P_0 + R)^{-1}$	0

Examples of filter divergence (2)

Consider the filtering problem with

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1+\delta \end{bmatrix}$$

$$P_0 = I_3, \quad R = \delta^2 I_2$$

where $\delta^2 < \varepsilon$ but $\delta > \varepsilon$.

The we get

$$HP_0H^T = \begin{bmatrix} 3 & 3+\delta \\ 3+\delta & 3+2\delta+\delta^2 \end{bmatrix}$$

which is singular to machine precision.

Implementation methods for Kalman filtering

Implementation issues

- Symmetry of P:Joseph formula;
- Scalar updates of the state estimate;
- Symmetry, computational cost and roundoff error: factorisation methods;

Symmetry of P: the Joseph form

The covariance propagation equations are given by

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q_{k-1}$$

$$P_k(+) = \left[I - \bar{K}_k H_k\right] P_k(-)$$

The first equation already guarantees symmetry.
The second can be equivalently written as

$$P_k(+) = \left[I - \bar{K}_k H_k\right] P_k(-) \left[I - \bar{K}_k H_k\right]^T + \bar{K}_k R_k \bar{K}_k^T$$

which again guarantees symmetry.

NOTE: this is the least one can do in implementing a KF!

Scalar updates of the state estimate (1)

Consider the measurement equation

$$z_k = H_k x_k + v_k, \quad v_k \simeq WN(0, R_k)$$

and assume that R_k is diagonal, i.e., the measurements are *statistically independent*.

Then the computation of the gain and the update of the estimate can be carried out considering each measurement individually.

Scalar updates of the state estimate (2)

Advantages:

- Reduced computational cost:
  - Vector implementation: cost grows as l^3;
  - Scalar implementation: cost grows as l;

- Improved numerical accuracy: consider the computation of

$$\bar{K}_k = P_k(-)H_k^T \left[H_k P_k(-) H_k^T + R_k\right]^{-1}$$

If z_k is scalar then we avoid matrix inversion!

Scalar updates formulas

$$P_k^{[0]} = P_k(-), \quad \hat{x}_k^{[0]} = \hat{x}_k(-)$$

for i=1, .., l

$$\bar{K}_k^{[i]} = \frac{1}{H_k^{[i]} P_k^{[i-1]} H_k^{[i]T} + R_k^{[i]}} P_k^{[i-1]} H_k^{[i]T}$$

$$P_k^{[i]} = P_k^{[i-1]} - \bar{K}_k^{[i]} P_k^{[i-1]} H_k^{[i]}$$

$$\hat{x}_k^{[i]} = \hat{x}_k^{[i-1]} + \bar{K}_k^{[i]} \left[\{z_k\}_i - H_k^{[i]} \hat{x}_k^{[i-1]} \right]$$

end for;

$$P_k^{[l]} = P_k(+), \quad \hat{x}_k^{[l]} = \hat{x}_k(+)$$

Application: handling sensor faults

- In many applications the operation of the KF must be guaranteed in the presence of sensor faults;
- The scalar update allows to "switch off" a faulty sensor without affecting the operation of the filter (provided that the system remains observable).
- Sensor faults can be detected by monitoring the innovation for each measured output:

$$\{z_k\}_i - H_k^{[i]}\widehat{x}_k(+)$$

An example

Consider the linear, discrete-time system given by

$$x_k = \Phi x_{k-1} + w_{k-1}$$
$$z_k = H x_k + v_k$$

$$\Phi = \begin{bmatrix} 0 & 1 \\ -0.4 & 0.6 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},$$
$$Q = \sigma_w^2 I_2, \quad R = \sigma_v^2, \quad \sigma_w = 0.1, \quad \sigma_v = 0.01$$

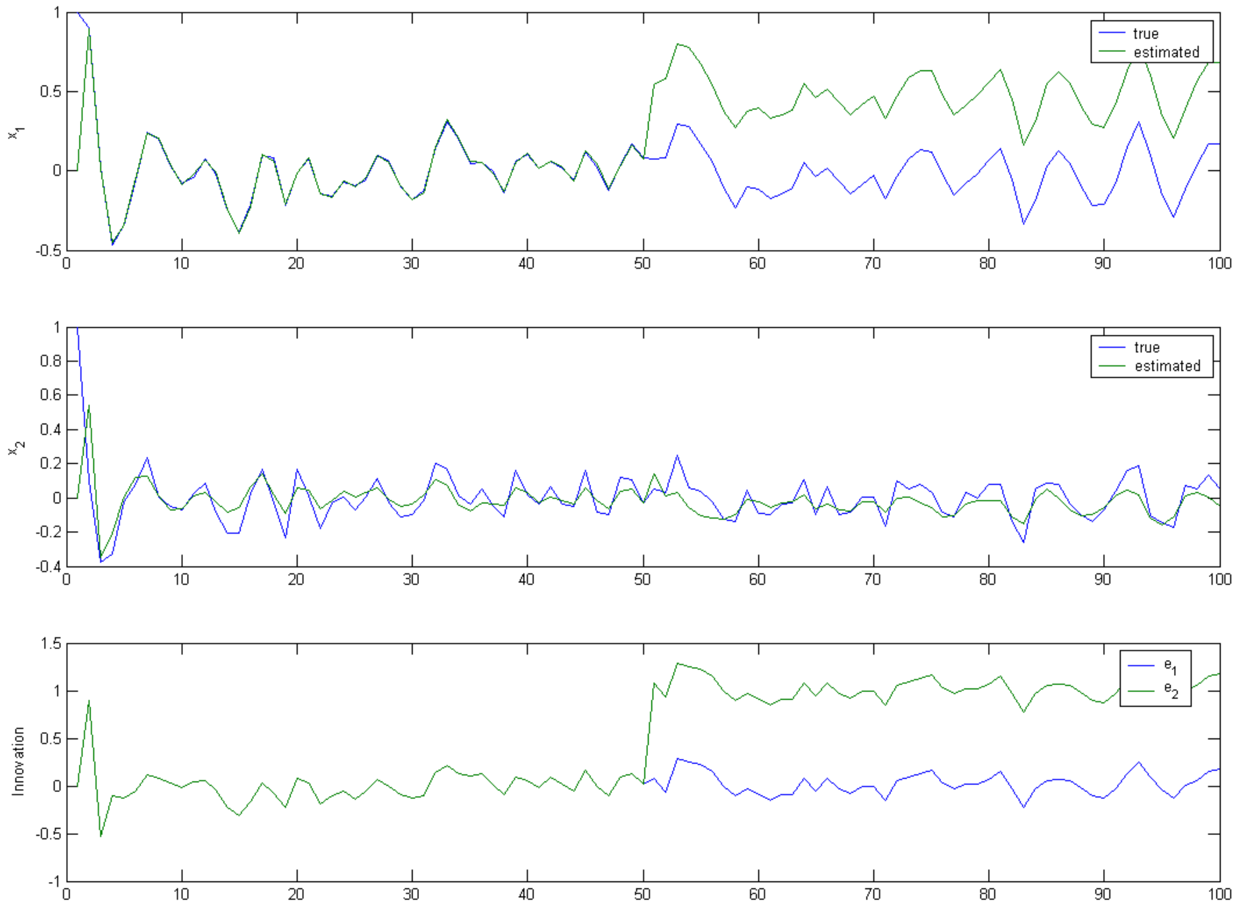
Now we have two sensors measuring x_1.

At time k=50, the second sensor becomes biased:

$$\{z_k\}_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k + 1, \quad k > 50$$

An example (2)

Simulation results



An example (3)

- The problem with sensor 2 can be detected by monitoring $\{e_k\}_2$;

- The faulty sensor can then be switched off;

- If needed, a warning can be sent to a supervision system.

Overview of factorisation methods

Symmetry can be ensured and numerical stability can be improved by using one or more of the following ideas:

- Factoring P into Cholesky (or UDU) factors;
- Factoring R (to simplify observational update) and/or Q (to simplify temporal update);
- Taking square roots of elementary matrices;
- Using QR factorisations of general matrices;

The Potter square root filter

Main idea: factor P(-) and P(+) according to:

$$P(-) = C(-)C^T(-), \quad P(+) = C(+)C^T(+)$$

so that the observational update

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

becomes

$$\begin{aligned}
&C(+)C^T(+) = \\
&= C(-)C^T(-) - C(-)\boxed{C^T(-)H^T}\left(HC(-)C^T(-)H^T + R\right)^{-1} HC(-)C^T(-) = \\
&= C(-)C^T(-) - C(-)\boxed{V}\left(V^T V + R\right)^{-1} V^T C^T(-) = \\
&= C(-)\left[I_n - V\left(V^T V + R\right)^{-1} V^T\right] C^T(-)
\end{aligned}$$

The Potter square root filter (2)

Matrix

$$\left[I_n - V\left(V^TV + R\right)^{-1}V^T\right]$$

is symmetric, so if we can factor it as WW^T we can obtain the complete "square root" update:

$$C(+)C^T(+) = C(-)WW^TC^T(-) \quad \Rightarrow \quad C(+) = C(-)W$$

Consider the special case of a *scalar* measurement z. Then $V=v=C^T(-)H^T$ is a column vector and WW^T reduces to

$$WW^T = \left[I_n - \frac{vv^T}{R + \|v\|^2}\right]$$

The Potter square root filter (3)

Computing the square root of $\left[I_n - svv^T\right]$

We have that

$$\left[I_n - svv^T\right]^{1/2} = \left[I_n - \sigma vv^T\right]$$

where

$$\sigma = \frac{1 + \sqrt{1 - s\|v\|^2}}{\|v\|^2}$$

provided that $1 - s\|v\|^2 \geq 0$

The Potter square root filter (4)

Therefore in our case we have that

$$WW^T = \left[I_n - \frac{vv^T}{R + \|v\|^2} \right] \Rightarrow s = \frac{1}{R + \|v\|^2}$$

) $$\sigma = \frac{1 + \sqrt{1 - s\|v\|^2}}{\|v\|^2} = \frac{1 + \sqrt{R/(R + \|v\|^2)}}{\|v\|^2}$$

So the update is

$$C(+) = C(-) \left[I_n - \sigma v v^T \right]$$

The Potter square root filter (5)

Main advantages:

- Reduced computational cost (only "half" of the covariance is updated);
- Inherent symmetry of the covariance matrix.