A family of hybrid Kalman/H∞ filtering algorithms

Paul A. Beltyukov <beltyukov.p.a@gmail.com>

Annotation

A family of hybrid filtering algorithms derived. H∞ filtering is used to correct Kalman filter divergence. A numerical experiment on a toy problem with python prototypes was taken to choose algorithms for implementation in software libraries for embedded systems.

Introduction

The Kalman filter is known the since 60s of the 20th century [1]. During the past six decades it has been used in a wide range of applications from navigation systems to portfolio management. During the Apolo mission [3] the Extended Kalman Filter (EKF) [2] was developed by NASA engineers for nonlinear applications.

When the EKF was used in industrial applications it was found to diverge in some cases [4]. The reasons of the EKF divergence were inaccuracies of the mathematical models of observed processes. To mitigate the problem of the EKF divergence several approaches such as [6] were developed.

The industry needed some filtration algorithms which can guarantee filter convergence in cases where accurate process models were unavailable or too expensive to develop. The H∞ filter was developed in 80s to meet the industry needs in robust filtering algorithms. In [5] the game theory approach was used to derive H∞ filter.

We will compare the Kalman and the H∞ filters. Then we will consider the usage later to guarantee the Kalman filter convergence. Finally we will derive hybrid Kalman/H∞ filters and compare them with Kalman filter on a toy problem.

The filtering problem

Let's consider a system with the folowing state space model:

$$x_k = F(x_{k-1}, u_{k-1}, w_{k-1}) \quad ;$$

$$z_k = H(x_k) + v_k \quad ;$$

where x_k is the systems state vector, u_{k-1} is the system perturbation vector, z_k is the observation vector, w_{k-1} is the Gaussian systems noise with covariance Q_k and with zero average,

v_k is the Gaussian obesvation noise with R_k covariance and with zero average.

The filtering problem is to estimate the system state vector using series of noisy observations with respect to minimizing some cost function.

The Kalman filter

The Kalman filter is described in [1], during its work it minimizes the cost function:

$$J = E\left(\|z_k - \hat{z}_k\|_2^2\right) \quad ,$$

where $\hat{z}_k$ is the measurement estimate, which is taken from the filter output.

The Extended Kalman filter algorithm is:

Prediction:

$$x_{k|k-1} = F\left(x_{k-1|k-1}, u_{k-1}, 0\right) \quad ;$$

$$F_k = \left.\frac{\partial F}{\partial x}\right|_{x_{k-1|k-1}} \quad ; \quad B_k = \left.\frac{\partial F}{\partial w}\right|_{x_{k-1|k-1}} \quad ; \quad H_k = \left.\frac{\partial H}{\partial x}\right|_{x_{k|k-1}}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + B_k Q_k B_k^T \quad ;$$

$$v_k = z_k - H\left(x_{k|k-1}\right) \quad ;$$

Correction:

$$P_{k|k} = \left(P_{k|k-1}^{-1} + H_k^T R_k^{-1} H_k\right)^{-1} \quad ;$$

$$x_{k|k} = x_{k|k-1} + K_k\, v_k \quad ;$$

$$K_k = P_{k|k} H_k R_k^{-1} \quad ;$$

There is also more convenient form which is:

Prediction:

$$x_{k|k-1} = F\left(x_{k-1|k-1}, u_{k-1}, 0\right) \quad ;$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + B_k Q_k B_k^T \quad ;$$

$$F_k = \left.\frac{\partial F}{\partial x}\right|_{x_{k-1|k-1}} \quad ; \quad B_k = \left.\frac{\partial F}{\partial w}\right|_{x_{k-1|k-1}} \quad ; \quad H_k = \left.\frac{\partial H}{\partial x}\right|_{x_{k|k-1}} \quad ;$$

The H_k is asumed to have full rank;

$$v_k = z_k - H\left(x_{k|k-1}\right) \quad ;$$

Correction:

$$K_k = P_{k|k-1} H_k^T (R_k + H_k P_{k|k-1} H_k^T)^{-1} \quad ;$$

$$x_{k|k} = x_{k|k-1} + K_k \nu_k \quad ;$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T = (I - K_k H_k) P_{k|k-1} \quad ;$$

In cases when the Kalman fiter convergence can not be guaranteed one can use H∞ filter or some mixed/hybrid filtration algorithms.

The discrete time H∞ filter

The H∞ filter [5] estimates some linear combination of the state vector but not the state vector itself as in the Kalman filter:

$$y_k = L_k x_k$$

where L_k is some full rank matrix which is given by the filter user.

The H∞ filter is minimizing the folowing cost function:

$$J = \frac{\sum_{k=0}^{N-1} \|y_k - \hat{y}_k\|^2_{S_k}}{\|x_0 - \hat{x}_0\|^2_{P_0^{-1}} + \sum_{k=0}^{N-1} \left(\|w_k\|^2_{Q_k^{-1}} + \|v_k\|^2_{R_k^{-1}} \right)} < \frac{1}{\theta} \quad ;$$

where $\hat{y}_k$ is y_k estimate which minimizes J, θ -is user defined threshold.

Let's denote the H∞ filter algorithm as follows:

Prediction:

$$x_{k|k-1} = F(x_{k-1|k-1}, u_{k-1}, 0) \quad ;$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + B_k Q_k B_k^T \quad ;$$

$$F_k = \left. \frac{\partial F}{\partial x} \right|_{x_{k-1|k-1}} \quad ; \quad B_k = \left. \frac{\partial F}{\partial w} \right|_{x_{k-1|k-1}} \quad ; \quad H_k = \left. \frac{\partial H}{\partial x} \right|_{x_{k|k-1}} \quad ;$$

$$\nu_k = z_k - H(x_{k|k-1}) \quad ;$$

Correction:

$$P_{k|k} = P_{k|k-1}^{-1} (I - \theta \tilde{S}_k P_{k|k-1} + H_k^T R_k^{-1} H_k P_{k|k-1})^{-1} = (P_{k|k-1}^{-1} - \theta \tilde{S}_k + H_k^T R_k^{-1} H_k)^{-1} \quad ;$$

$\tilde{S}_k = L_k^T \bar{S}_k L_k$, where $\bar{S}_k > 0$ is user defined matrix;

$$x_{k|k} = x_{k|k-1} + K_k \nu_k \quad ;$$

$$K_k = P_{k|k} H_k R_k^{-1} \quad ;$$

The filter performance criterion is:

$$P_{k|k}=\left(P_{k|k-1}^{-1}-\theta\widetilde{S}_k+H_k^T R_k^{-1} H_k\right)^{-1}>0 \quad ;$$

The $P_{k|k}$ matrix can be denoted as two consequent operations as follows:

$$P_b=\left(P_{b-1}^{-1}-\theta\widetilde{S}_b\right)^{-1}=P_{b-1}+\Delta Q_b \quad (1);$$

$$P_d=\left(P_{d-1}^{-1}+H_d^T R_d^{-1} H_d\right)^{-1} \quad (2);$$

The (2) operation is state estimate covariation update which is also used in the Kalman filter. This operation usually preserves the positive definiteness of the state estimate covariation matrix;

For rhe (1) operation to preserve the positive definiteness of the state estimate covariation matrix it is necessary and sufficient that the following criterion is fulfilled:

$$\Delta Q_b>0 \quad (3);$$

The order of operations is determined by the particular filtering algorithm implementation.

Hybrid filters

The mixed filtering algorithm whict is in fact the H_∞ filter with the specific L_k, S_k, θ parameters choise is given in [7].

There also hybrid filtering algorithms which use Kalman filter algorithm by default and which use H_∞ technolohy [8] based adaptive correction when the Kalman filter diverges. These algorithms however minimize the cost function which is significantly different from H_∞ filter [5] cost function.

We will derive two classes of hybrid filtering algorithms which use Kalman filter by default and if the filter diverges switch to H_∞ filtering [5] in order to restore the Kalman filter integrity.

We also will derive sequential UD-factrorized versions of these algorithms.

Keeping the Kalman filter convergence: an approach with no default limitations on H_∞ filter parameters

To restore the Kalman filter conregence we are going to use the operation (1).

The chi-squareв test is used to detect the Kalman filter divergence. According to [9] the quiadratic form

$$\beta_b=\delta_b^T \hat{S}_b^{-1} \delta_b$$

has the chi-squared distrubution;

The parameters of that quadratic form at the moment of computation are defined as follows: $\hat{S}_b = R_b + H_b P_b H_b^T$ is innovation covariance, R_b is observation noise covariance, $H_b = \left.\frac{\partial H}{\partial x}\right|_{x_b}$, P_b is the state estimate covariance, $\delta_b = z_b - H(x_b) \neq 0$ is innovation .

The filter convergence criterion is:

$$\beta_b \leqslant \beta_n = \chi^2_{\alpha,n} \quad (4),$$

where n is the state dimensionality and α is significance level.

In order to maintain the filter convregrnce we need P_b after operation (1) to satisfy the equation:

$$\delta_b^T (R_b + H_b P_b H_b^T)^{-1} \delta_b = \beta_n \quad ;$$

$$\delta_b^T (R_b + H_b (P_{b-1} + \Delta Q_b) H_b^T) \delta_b = \beta_n \quad ;$$

$$\delta_b^T (R_b + H_b P_{b-1} H_b^T + H_b \Delta Q_b H_b^T)^{-1} \delta_b = \beta_n \quad ;$$

$$\delta_b^T (R_b + H_b P_{k-1} H_b^T + H_b \Delta Q_b H_b^T)^{-1} \delta_b = \beta_n \quad ;$$

$$\delta_b^T (S_b + H_b \Delta Q_b H_b^T)^{-1} \delta_b = \beta_n \quad ,$$

where $S_b = R_b + H_b P_{b-1} H_b^T$ (5) is innovation covariance before (1) operation;

$$\delta_b \delta_b^T (S_b + H_b \Delta Q_b H_b^T)^{-1} \delta_b \delta_b^T = \beta_n \delta_b \delta_b^T \quad ;$$

If we compare the sides of the equiation we can see that:

$$\delta_b \delta_b^T (S_b + H_b \Delta Q_b H_b^T)^{-1} = \beta_n I \quad ;$$

$$\delta_b \delta_b^T = \beta_n (S_b + H_b \Delta Q_b H_b^T) \quad ;$$

$$\frac{\delta_b \delta_b^T}{\beta_n} = S_b + H_b \Delta Q_b H_b^T \quad ;$$

$$A_b = H_b \Delta Q_b H_b^T = \frac{\delta_b \delta_b^T}{\beta_n} - S_b \quad (6);$$

On the other hand from (1) we can show the folowing:

$$P_b = (P_{b-1}^{-1} - \theta \widetilde{S}_b)^{-1} \quad \text{, where} \quad \widetilde{S}_b = L_b^T \bar{S}_b L_b \quad ;$$

$$P_b = (P_{b-1}^{-1} - \theta L_b^T S_b L_b)^{-1} = (P_{b-1}^{-1} - L_b^T M_b^{-1} L_b)^{-1} \quad ,$$

where $M_b^{-1} = \theta S_b$;

$$P_b = (P_{b-1}^{-1} - L_b^T M_b^{-1} L_b)^{-1} = P_{b-1} + P_{b-1} L_b^T (M_b + L_b P_{b-1} L_b^T) L_b P_{b-1} = P_{b-1} + \Delta Q_b \quad ;$$

$$\Delta Q_b = P_{b-1} L_b^T (M_b + L_b P_{b-1} L_b^T) L_b P_{b-1} \quad (7);$$

$$A_b = H_b \Delta Q_b H_b^T = H_b P_{b-1} L_b^T (M_b + L_b P_{b-1} L_b^T) L_b P_{b-1} H_b^T \quad ;$$

$$A_b = H_b P_{b-1} L_b^T (M_b + L_b P_{b-1} L_b^T) L_b P_{b-1} H_b^T \quad ;$$

$A_b = C_b (M_b + B_b) C_b^T$, where:

$$C_b = H_b P_{b-1} L_b^T \quad (8),$$

$$B_b = L_b P_{b-1} L_b^T \quad ;$$

$$C_b^T A_b C_b = C_b^T C_b (M_b + B_b) C_b^T C_b \quad (9);$$

Let us consider the left side multipliers of the equation sides (9):

$$C_b^T A_b C_b = (C_b^T C_b + \varepsilon I - \varepsilon I)(M_b + B_b) C_b^T C_b = (C_b^T C_b + \varepsilon I)(M_b + B_b) C_b^T C_b - \varepsilon I (M_b + B_b) C_b^T C_b \quad ;$$

$$(C_b^T C_b + \varepsilon I)^{-1} C_b^T A_b C_b = (M_b + B_b) C_b^T C_b - \varepsilon (C_b^T C_b + \varepsilon I)^{-1} (M_b + B_b) C_b^T C_b \quad ;$$

As the matrices in the equation are symmetric:

$$(C_b^T C_b + \varepsilon I)^{-1} C_b^T A_b C_b = (M_b + B_b) C_b^T C_b - \varepsilon (C_b^T C_b + \varepsilon I)^{-1} C_b^T C_b (M_b + B_b) \quad ;$$

If we consider the limits of the sides of the equation with $\varepsilon \rightarrow +0$ then we will get:

$$C_b^+ A_b C_b = (M_b + B_b) C_b^T C_b \quad ;$$

By doing the same things with rigth side multipliers of the sides of the equation we get:

$$C_b^+ A_b C_b^{+T} = M_b + B_b \quad ;$$

$$M_b = C_b^+ A_b C_b^{+T} - B_b = C_b^+ A_b C_b^{+T} - L_b P_{b-1} L_b^T \quad ;$$

By substitution of M_b to equation (7) we get:

$$\Delta Q_b = P_{b-1} L_b^T (C_b^+ A_b C_b^{+T} - L_b P_{b-1} L_b^T + L_b P_{b-1} L_b^T) L_b P_{b-1} \quad ;$$

$\Delta Q_b = P_{b-1} L_b^T C_b^+ A_b C_b^{+T} L_b P_{b-1}$ (10), where:

$A_b = \dfrac{\delta_b \delta_b^T}{\beta_n} - S_b$, $S_b = R_b + H_b P_{b-1} H_b^T$, $C_b = H_b P_{b-1} L_b^T$ see equations (6), (5) and (8) respectively.

To fulfill the criterion (3) it is necessary and sufficient to meet the following conditions [12]:

- The matrix $Z_b = P_{b-1} L_b^T C_b^+$ has to have full rank;
- $A_b > 0$ (11);

Let us consider $Z_b = P_{b-1} L_b^T C_b^+$:

- as $P_b > 0$, and L_b have full rank, then the product $P_{b-1} L_b$ also has full rank;
- as H_b and L_b have full rank, a $P_b > 0$, then C_b also has full rank, consequently: the product $C_b^T C_b > 0$ [12], and so, $C_b^+ = (C_b^T C_b)^{-1} C_b^T$ also have full rank;
- as both product $P_{b-1} L_b$ and the matrix C_b^+ have full renk, then Z_b also has full rank.

Let us consider the matrix A_b :

- $S_b \geq H_b P_{b-1} H_b^T > 0$;
- now we must consider the product $x^T(\delta_b \delta_b^T)x = (x^T \delta_b)(\delta_b^T x)$ for any $x \neq 0$:
  - if the dimensionality of x is greater than one, then $x^T(\delta_b \delta_b^T)x = x^T(\delta_b \delta_b^T)x \geq 0$, consequently, regarding to positive definiteness of S_b , we **can not** guarante the existense of hybrid filter;
  - if the dimensionality of x is one (when we implement sequential filter), then $x^T(\delta_b \delta_b^T)x = \delta_b^2 x^2 > 0$,
    - in such case the equation (5) reduces to $a_b = \frac{\delta_b^2}{\beta_1} - s_b$,
    - the criterion (4) is violated when $\frac{\delta_b^2}{s_b} > \beta_1$ or $\frac{\delta_b^2}{\beta_1} - s_b > 0$,
    - so when thecriterion (4) is violated we have $A_b = a_b > 0$,

thus trere is at least one sequential implementation.

So we can select L_b, S_b, θ parameters of H_∞ filter so that we can use operation (1) to correct the Kalman filter divergence. Our selection is reduced to definition of L_b and δ_b to be used in certain filtering algorithm.

The hybrid sequential UD-factorized filter with prior residuals used for correction (UDEKHF_A)

In this filtering algorithm we use $L_k = H_k^T$ and innovation as H∞ filter parameters.

Initialization:

$$x_{0|0} = x_0 \quad ; \quad U_{0|0}, D_{0|0} = udu(P_0) \quad ; \quad U_q, D_q = udu(Q) \quad ; \quad U_r^{-1}, D_r = udu(R) \quad ,$$

where $D_r = [r_1 \ldots r_j \ldots r_n]$;

Work:

1 Prediction:

$$m_0 = x_{k|k-1} = F(x_{k-1|k-1}, u_{k-1}, 0) \quad ;$$

$$F_k = \left.\frac{\partial F}{\partial x}\right|_{x_{k-1|k-1}} \quad ; \quad B_k = \left.\frac{\partial F}{\partial w}\right|_{x_{k-1|k-1}} \quad ;$$

$$U_{0,} D_0 = MWGS([F_k U_{k-1|k-1} \vdots B_k U_q], [D_{k-1|k-1} \vdots D_q]) \quad ;$$

2 Correction:

$$\hat{z} = U_r z_k \quad ;$$

2.1 for j = 1,n do:

$$v_j = \hat{z}_j - H_j(m_{j-1}) \quad ;$$

$$h_j = \left(U_r \left.\frac{\partial H}{\partial x}\right|_{m_{j-1}} \right)_j \quad ;$$

2.1.1 Divergency detection:

$$f_j = h_j U_{j-1} \quad ; \quad v_j = D_{j-1} f_j^T \quad ; \quad c_j = f_j v_j \quad ; \quad s_j = c_j + r_j \quad ;$$

$$d_j = \frac{v_j^2}{\beta_1} - s_j \quad ;$$

If $d_j \geqslant 0$ then adaptive correction needed:

$$q_j = \frac{d_j}{c_j^2} \quad ;$$

$$\tilde{U}_{j-1}, \tilde{D}_{j-1} = MWGS([U_{j-1} \vdots U_{j-1} v_j], [D_{j-1} \vdots q_j]) \quad ;$$

$$\tilde{f}_j = h_j \tilde{U}_{j-1} \quad ; \quad \tilde{v}_j = \tilde{D}_{j-1} \tilde{f}_j^T \quad ; \quad \tilde{\alpha}_j = \tilde{f}_j \tilde{v}_j + r_j \quad ;$$

else:

$$\widetilde{U}_{j-1},\widetilde{D}_{j-1},\widetilde{f}_j,\widetilde{v}_j,\widetilde{\alpha}_j=U_{j-1},D_{j-1},f_j,v_j,e_j \quad ;$$

2.1.2 The filter estimate covariance and state update:

$$K_j=\widetilde{U}_{j-1}\widetilde{v}_j\widetilde{\alpha}_j^{-1} \quad ;$$

$$m_j=m_{j-1}+K_j v_j \quad ;$$

$$U_j,D_j=MWGS([K_j\widetilde{f}_j-\widetilde{U}_{j-1}\vdots K_j],[\widetilde{D}_{j-1}\vdots r_j]) \quad ;$$

2.2 Finalize the correction :

$$U_{k|k},D_{k|k}=U_n,D_n \quad ;$$

$$x_{k|k}=m_n \quad .$$

The hybrid sequential UD-factorized filter with posterior residuals used for correction (UDEKHF_B)

The folowing L_k value was chosen for adaptive correction in [8] along with posterior residuals :

$$L_k=\widetilde{H}_k^T \text{, where } \widetilde{H}_k=\left.\frac{\partial H(x+\widetilde{K}_k(z_k-H(x)))}{\partial x}\right|_{x_{k|k-1}}=\left.\frac{\partial H}{\partial x}\right|_{\widetilde{x}_{k|k}}\left(\left.\frac{\partial x}{\partial x}\right|_{x_{k|k-1}}-\widetilde{K}_k\left.\frac{\partial H}{\partial x}\right|_{x_{k|k-1}}\right)=\widehat{H}_k(I-\widetilde{K}_k H_k) \;;$$

Here is the filtering algorithm for such parameter selection:

Initializatio:

$$x_{0|0}=x_0 \quad ; \quad U_{0|0},D_{0|0}=udu(P_0) \quad ; \quad U_q,D_q=udu(Q) \quad ; \quad U_r^{-1},D_r=udu(R) \text{ , где } D_r=[r_1\ldots r_j\ldots r_n] \quad ;$$

Work:

1 Predictiov:

$$m_0=x_{k|k-1}=F(x_{k-1|k-1},u_{k-1},0) \quad ;$$

$$F_k=\left.\frac{\partial F}{\partial x}\right|_{x_{k-1|k-1}} \quad ; \quad B_k=\left.\frac{\partial F}{\partial w}\right|_{x_{k-1|k-1}} \quad ;$$

$$U_{0,}D_0=MWGS([F_k U_{k-1|k-1}\vdots B_k U_q],[D_{k-1|k-1}\vdots D_q]) \quad ;$$

2 Correction:

$$\hat{z}=U_r z_k \quad ;$$

2.1 for j = 1,n do:

$$v_j=\hat{z}_j-H_j(m_{j-1}) \quad ;$$

$$h_j = \left(U_r \frac{\partial H}{\partial x} \bigg|_{m_{j-1}} \right)_j \quad ;$$

Update the filter estimate covariance and state:

$$f_j = h_j U_{j-1} \quad ; \quad v_j = D_{j-1} f_j^T \quad ; \quad \alpha_j = f_j v_j + r_j \quad ;$$

$$K_j = U_{j-1} v_j \alpha_j^{-1} \quad ;$$

$$\tilde{m}_j = m_{j-1} + K_j v_j \quad ;$$

$$\tilde{U}_j, \tilde{D}_j = MWGS([K_j f_j - U_{j-1} \vdots K_j], [D_{j-1} \vdots r_j]) \quad ;$$

2.1.1 filter divegrence detection:

$$\eta_j = \hat{z}_j - H_j(\tilde{m}_j) \quad ;$$

$$\hat{h}_j = \left(U_r \frac{\partial H}{\partial x} \bigg|_{\tilde{m}_j} \right)_j \quad ;$$

$$\hat{f}_j = \hat{h}_j \tilde{U}_j \quad ; \quad \hat{v}_j = \tilde{D}_j \hat{f}_j^T \quad ; \quad s_j = \hat{f}_j \hat{v}_j + r_j \quad ;$$

$$d_j = \frac{\eta_j^2}{\beta_1} - s_j \quad ;$$

if $d_j \geqslant 0$ then do adaptive correction:

$$\tilde{h}_j = \hat{h}_j - \hat{h}_j K_j h_j \quad ;$$

$$\tilde{f}_j = \tilde{h}_j \tilde{U}_j \quad ; \quad \tilde{v}_j = \tilde{D}_j \tilde{f}_j^T \quad ; \quad c_j = \hat{f}_j \tilde{v}_j \quad ;$$

$$q_j = \frac{d_j}{c_j^2} \quad ;$$

$$U_j, D_j = MWGS([\tilde{U}_j \vdots \tilde{U}_j \tilde{v}_j], [\tilde{D}_j \vdots q_j]) \quad ;$$

$$m_j = m_{j-1} + U_j D_j U_j^T h_j^T \frac{v_j}{r_j} \quad ;$$

else:

$$U_j, D_j, m_j = \tilde{U}_j, \tilde{D}_j, \tilde{m}_j \quad ;$$

2.2 Finalize the correction:

$$U_{k|k}, D_{k|k} = U_n, D_n \quad ;$$

$$x_{k|k} = m_n \quad .$$

Keeping the Kalman filter convergence: an approach with limitations on H∞ filter parameters by default.

In this aproach we will use the following selection of H∞ filter parameters:

$L_k = I$;

$\bar{S}_b = a_b^{-1} P_b^{-1}$;

$\theta = 1$.

This selection leads to the following operation (1) reduction:

$P_a = (P_b^{-1} - a_b^{-1} P_b^{-1})^{-1} = P_b + a_b P_b$ (11).

To make operation (11) to preserve positive definiteness of estimate covariance it is necessary and sufficient to have:

$0 < a_b^{-1} < 1$.

In such case:

$\Delta Q_b = a_b P_b$ (13), where $a_b > 0$ (14).

By substitution of the equation (13) to the equation (6) we get:

$$a_b H_b P_b H_b^T = \frac{\delta_b \delta_b^T}{\beta_n} - S_b \quad (15);$$

Let us multiply the equation (15) with δ_b rigth side and δ_b^T left side respectively:

$$a_b \delta_b^T H_b P_b H_b^T \delta_b = \frac{(\delta_b^T \delta_b)(\delta_b^T \delta_b)}{\beta_n} - \delta_b^T S_b \delta_b = \frac{(\delta_b^T \delta_b)^2}{\beta_n} - \delta_b^T S_b \delta_b \quad ;$$

then we get:

$$a_b = \frac{\frac{(\delta_b^T \delta_b)^2}{\beta_n} - \delta_b^T S_b \delta_b}{\delta_b^T H_b P_b H_b^T \delta_b} = \frac{\delta_b^T A_b \delta_b}{\delta_b^T H_b P_b H_b^T \delta_b} \quad (16);$$

In order for a hybrid filter to exist it is sufficient to meet condition (14) in when criterion (4) gets violated.

The denominator of the equation (16) is positive for nonzero δ_b , so we only have to consider the sign of the numerator of this equation.

Let us consider the flowing expression:

$$\delta_b^T S_b^{-1} \delta_b > \beta_n \Leftrightarrow \delta_b \delta_b^T > \beta_n S_b \Leftrightarrow \delta_b^T (\delta_b \delta_b^T) \delta_b - \beta_n \delta_b^T S_b \delta_b > 0 \Leftrightarrow \frac{(\delta_b^T \delta_b)^2}{\beta_n} - \delta_b^T S_b \delta_b > 0 \quad ;$$

So the numerator of the equation (16) is posotive for any nonzero δ_b when the criterion (4) gets violated, so the hybrid filter existence is proven.

The hybrid filter with prior residuals used for correction (EKHF_C)

In this algorothm we use innovation to correct kalman filter divergence. The algorithm is:

Prediction:

$$x_{k|k-1} = F(x_{k-1|k-1}, u_{k-1}, 0) \quad ;$$

$$F_k = \left.\frac{\partial F}{\partial x}\right|_{x_{k-1|k-1}} \quad ; \quad B_k = \left.\frac{\partial F}{\partial w}\right|_{x_{k-1|k-1}} \quad ; \quad H_k = \left.\frac{\partial H}{\partial x}\right|_{x_{k|k-1}} \quad ;$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + B_k Q_k B_k^T \quad ;$$

$$\nu_k = z_k - H(x_{k|k-1}) \quad ;$$

The filter divergence detecction:

$$S_k = R_k + H_k P_{k|k-1} H_k^T \quad ;$$

If $\nu_k^T S_k^{-1} \nu_k > \beta_n$ then adaptive correction needed::

$$A_k = \frac{\nu_k \nu_k^T}{\beta_n} - S_k \quad ;$$

$$a_k = \frac{\nu_k^T A_k \nu_b}{\nu_k^T H_k P_{k|k-1} H_k^T \nu_k} \quad ;$$

$$\widetilde{P}_{k|k-1} = (1 + a_k) P_{k|k-1} \quad ;$$

else:

$$\widetilde{P}_{k|k-1} = P_{k|k-1} \quad ;$$

Correction:

$$K_k = \widetilde{P}_{k|k-1} H_k^T (R_k + H_k \widetilde{P}_{k|k-1} H_k^T)^{-1} \quad ;$$

$$x_{k|k} = x_{k|k-1} + K_k \nu_k \quad ;$$

$$P_{k|k} = (I - K_k H_k) \widetilde{P}_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T = (I - K_k H_k) \widetilde{P}_{k|k-1} \quad .$$

The hybrid sequential UD-factorized filter with prior residuals used for correction (UDEKHF_C)

The sequential UD-factorized variant of the above algorithm is:

Initialization:

$$x_{0|0}=x_0 \quad ; \quad U_{0|0},D_{0|0}=udu\left(P_0\right) \quad ; \quad U_q,D_q=udu\left(Q\right) \quad ; \quad U_r^{-1},D_r=udu\left(R\right) \quad ,$$

where $D_r=[r_1 \ldots r_j \ldots r_n]$;

Work:

1 Prediction:

$$m_0=x_{k|k-1}=F\left(x_{k-1|k-1},u_{k-1},0\right) \quad ;$$

$$F_k=\left.\frac{\partial F}{\partial x}\right|_{x_{k-1|k-1}} \quad ; \quad B_k=\left.\frac{\partial F}{\partial w}\right|_{x_{k-1|k-1}} \quad ;$$

$$U_{0,}D_0=MWGS\left(\left[F_k U_{k-1|k-1} \vdots B_k U_q\right],\left[D_{k-1|k-1} \vdots D_q\right]\right) \quad ;$$

2 Correction:

$$\hat{z}=U_r z_k \quad ;$$

2.1 For j = 1,n do:

$$\nu_j=\hat{z}_j-H_j\left(m_{j-1}\right) \quad ;$$

$$h_j=\left(U_r \left.\frac{\partial H}{\partial x}\right|_{m_{j-1}}\right)_j \quad ;$$

2.1.1 The filter divergence detection:

$$f_j=h_j U_{j-1} \quad ; \quad v_j=D_{j-1} f_j^T \quad ; \quad c_j=f_j v_j \quad ; \quad e_j=c_j+r_j \quad ;$$

$$d_j=\frac{\nu_j^2}{\beta_1}-e_j \quad ;$$

if $d_j > 0$ then do adaptive correction:

$$a_j = \frac{d_j}{c_j} + 1 \quad ;$$

$$\tilde{U}_{j-1} = U_{j-1} \quad ; \quad \tilde{D}_{j-1} = a_j D_{j-1} \quad ; \quad \tilde{f}_j = f_j \quad ; \quad \tilde{v}_j = a_j v_j \quad ; \quad \tilde{\alpha}_j = a_k c_j + r_j \quad ;$$

else:

$$\tilde{U}_{j-1}, \tilde{D}_{j-1}, \tilde{f}_j, \tilde{v}_j, \tilde{\alpha}_j = U_{j-1}, D_{j-1}, f_j, v_j, e_j \quad ;$$

2.1.2 Update the filter estimate covariance and state:

$$K_j = \tilde{U}_{j-1} \tilde{v}_j \tilde{\alpha}_j^{-1} \quad ;$$

$$m_j = m_{j-1} + K_j v_j \quad ;$$

$$U_j, D_j = MWGS([K_j \tilde{f}_j - \tilde{U}_{j-1} \vdots K_j], [\tilde{D}_{j-1} \vdots r_j]) \quad ;$$

2.2 Finalize the correction:

$$U_{k|k}, D_{k|k} = U_n, D_n \quad ;$$

$$x_{k|k} = m_n \quad .$$

The hybrid filter with posterior residuals used for correction (EKHF_D)

The algorithm of a hybrid filter which uses posterior error for divergence correction is:

Prediction:

$$x_{k|k-1} = F(x_{k-1|k-1}, u_{k-1}, 0) \quad ;$$

$$F_k = \left.\frac{\partial F}{\partial x}\right|_{x_{k-1|k-1}} \quad ; \quad B_k = \left.\frac{\partial F}{\partial w}\right|_{x_{k-1|k-1}} \quad ; \quad H_k = \left.\frac{\partial H}{\partial x}\right|_{x_{k|k-1}} \quad ;$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + B_k Q_k B_k^T \quad ;$$

$$v_k = z_k - H(x_{k|k-1}) \quad ;$$

$$\tilde{K}_k = P_{k|k-1} H_k^T (R_k + H_k P_{k|k-1} H_k^T)^{-1} \quad ;$$

$$\tilde{P}_{k|k} = (I - \tilde{K}_k H_k) P_{k|k-1} (I - \tilde{K}_k H_k)^T + \tilde{K}_k R_k \tilde{K}_k^T = (I - \tilde{K}_k H_k) P_{k|k-1} \quad ;$$

$$\tilde{x}_{k|k} = x_{k|k-1} + \tilde{K}_k v_k \quad ;$$

The filter divergence detection:

$$\eta_k = z_k - H\left(\tilde{x}_{k|k}\right) \quad ;$$

$$\hat{H}_k = \left.\frac{\partial H}{\partial x}\right|_{\tilde{x}_{k|k}} \quad ;$$

$$S_k = R_k + \hat{H}_k \tilde{P}_{k|k} \hat{H}_k^T \quad ;$$

if $\nu_k^T S_k^{-1} \nu_k > \beta_n$ then do adaptive corrrection:

$$A_k = \frac{\eta_k \eta_k^T}{\beta_n} - S_k \quad ;$$

$$a_k = \frac{\eta_k^T A_k \eta_b}{\eta_k^T H_k P_{k|k-1} H_k^T \eta_k} \quad ;$$

$$P_{k|k-1} = \left(1 + a_k\right) \tilde{P}_{k|k-1} \quad ;$$

$$K_k = P_{k|k} H_k^T R_k^{-1} \quad ;$$

$$x_{k|k} = x_{k|k-1} + K_k \nu_k \quad ;$$

else:

$$P_{k|k} = \tilde{P}_{k|k} \quad ;$$

$$x_{k|k} = \tilde{x}_{k|k} \quad .$$

The hybrid sequential UD-factorized filter with posterior residuals used for correction (UDEKHF_D)

The sequential UD-factorized variant of the former algorithm is:

Initialization:

$$x_{0|0}=x_0 \quad ; \quad U_{0|0}, D_{0|0}=udu(P_0) \quad ; \quad U_q, D_q=udu(Q) \quad ; \quad U_r^{-1}, D_r=udu(R) \quad \text{, где} \quad D_r=[r_1 \ldots r_j \ldots r_n] \quad ;$$

Work:

1 Prediction:

$$m_0=x_{k|k-1}=F(x_{k-1|k-1}, u_{k-1}, 0) \quad ;$$

$$F_k=\left.\frac{\partial F}{\partial x}\right|_{x_{k-1|k-1}} \quad ; \quad B_k=\left.\frac{\partial F}{\partial w}\right|_{x_{k-1|k-1}} \quad ;$$

$$U_{0,} D_0=MWGS([F_k U_{k-1|k-1} \vdots B_k U_q],[D_{k-1|k-1} \vdots D_q]) \quad ;$$

2 Correction:

$$\hat{z}=U_r z_k \quad ;$$

2.1 For j = 1,n do:

$$\nu_j=\hat{z}_j-H_j(m_{j-1}) \quad ;$$

$$h_j=\left(U_r \left.\frac{\partial H}{\partial x}\right|_{m_{j-1}}\right)_j \quad ;$$

Update the filter estimate covariance and state:

$$f_j=h_j U_{j-1} \quad ; \quad v_j=D_{j-1} f_j^T \quad ; \quad \alpha_j=f_j v_j+r_j \quad ;$$

$$K_j=U_{j-1} v_j \alpha_j^{-1} \quad ;$$

$$\tilde{m}_j=m_{j-1}+K_j \nu_j \quad ;$$

$$\tilde{U}_j, \tilde{D}_j=MWGS([K_j f_j-U_{j-1} \vdots K_j],[D_{j-1} \vdots r_j]) \quad ;$$

2.1.1 The filter divergence detection:

$$\eta_j=\hat{z}_j-H_j(\tilde{m}_j) \quad ;$$

$$\hat{h}_j=\left(U_r \left.\frac{\partial H}{\partial x}\right|_{\tilde{m}_j}\right)_j \quad ;$$

$$\hat{f}_j=\hat{h}_j \tilde{U}_j \quad ; \quad \hat{v}_j=\tilde{D}_j \hat{f}_j^T \quad ; \quad e_j=\hat{f}_j \hat{v}_j+r_j \quad ; \quad d_j=\frac{\eta_j^2}{\beta_1}-e_j \quad ;$$

if $d_j > 0$ then do adaptive correction:

$$a_j = \frac{d_j}{c_j} + 1 \quad ;$$

$$U_j = \tilde{U}_j \quad ; \quad D_j = a_j \tilde{D}_j \quad ; \quad m_j = m_{j-1} + U_j D_j U_j^T h_j^T \frac{\nu_j}{r_j} \quad ;$$

else:

$$U_j, D_j, m_j = \tilde{U}_j, \tilde{D}_j, \tilde{m}_j \quad ;$$

2.2 Finalize the correction:

$$U_{k|k}, D_{k|k} = U_n, D_n \quad ;$$

$$x_{k|k} = m_n \quad .$$

Summary for hybrid filtering algorithms

Based on the formulation of hybrid filtration algorithms we the following preliminary conclusions:

1. Algorithms which use the filter innovation for the adaptive correction need less computations than ones using the filter posterior residuals.
2. In case of no filter divergence sequantial UD-factorized filters need almost no additional computations and memory if compared with sequential UD-factorized EKF.
3. The most simple algorithm is UDEKHF_C.

The numerical experiment

Let us compare the behaviour of hybrid filtering algorithms with the behaviour of the Kalman filter with an inaccurate process model.

As a preliminary assessment of hybrid filtering algorithms for algorithms selection

We will use a toy porblem “maneuvering object tracking” for preliminary assessment amd selection of hybrid filtering algorithms. Selected algorithms will be implemented in softaware libraries for embedded systems.

Maneuvering object tracking

Let us consider a tracking of object which commits a manuever [11]. An example track of such object and its measured locations are shown in fig. 1.

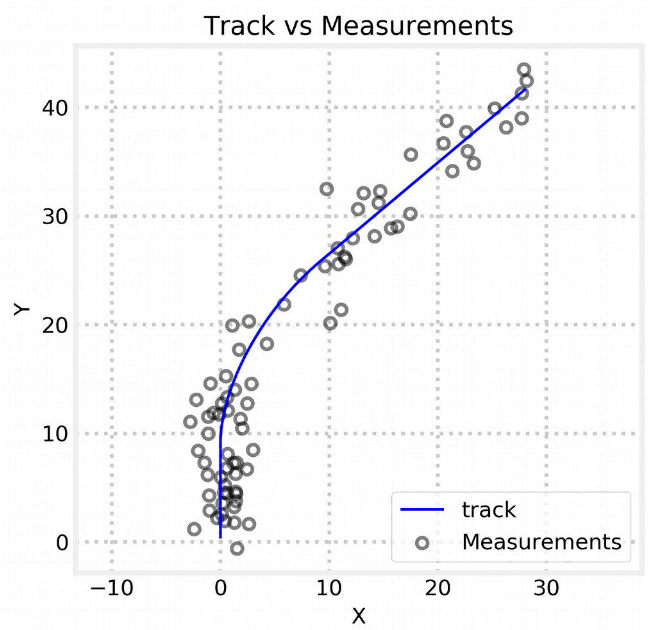


Figure 1: An example track and measured locations on a maneuvering object.

The object is moving with constant velocity and then it commtis a rigth turn with speedup. We will consider two cases of oservation of such object:

- one dimensional, when we track only one of objects coordinates,
- Two dimensional, when we track both objects coordinates.

In both cases we will use constant velocity model in object tracking.

One-dimensional case

We measure only X coordinate of the onject in this case.

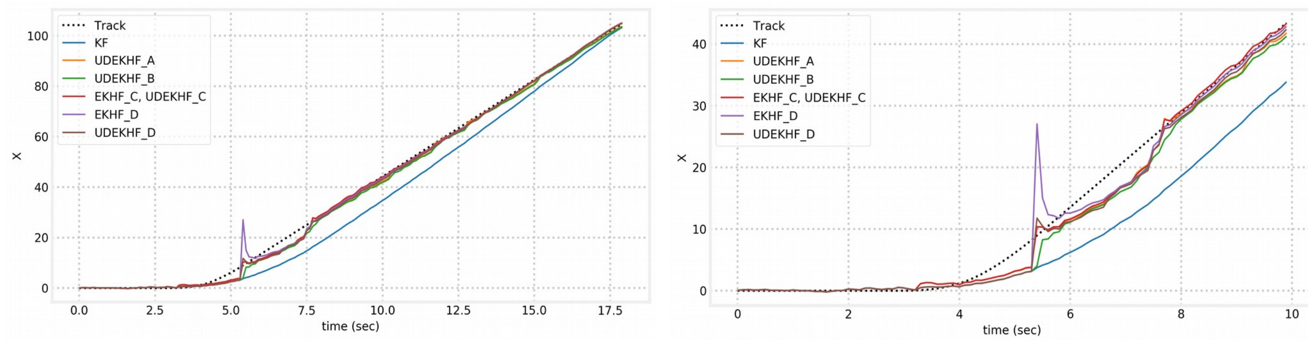
The state transition matrix is:

$$F_k = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \quad ;$$

The observation matrix is:

$$H_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad ;$$

The results of modeling in two time scales are shown in fig. 2:



a b

Figure 2: The objects X-coordinate and filtering results.

It is clear that the Kalman filter diverges when the objects commits the maneuver.

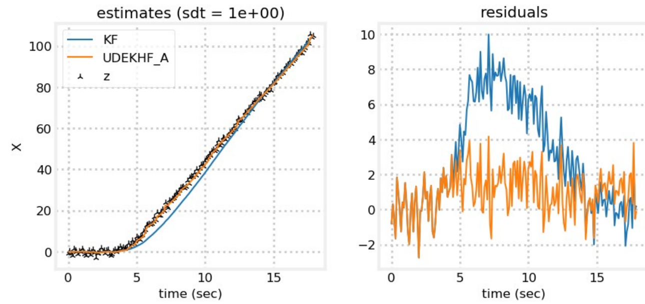
Hybrid filtering algorithms give more accurate estimetes than the Kalman filter with the exception of the spike given by EKHF_D.

The estimates given by EKHF_C and UDEKHF_C are identical and at sthe same time the estimates given by EKHF_D and UDEKHF_D are different.

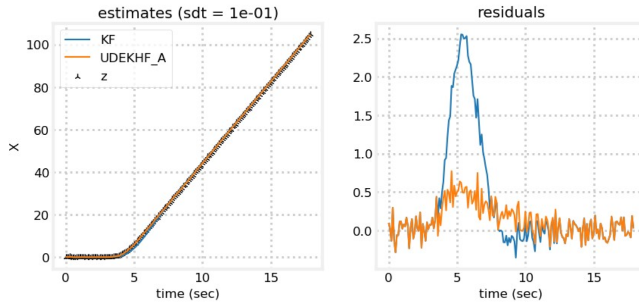
The spike in EKHF_D estimate may be associated with a probable lack of stability of the algorithm.

Let us compare the residuals of different hybrid filters with the Kalman filter residuals at different values of sensor noise stdendart deviation.

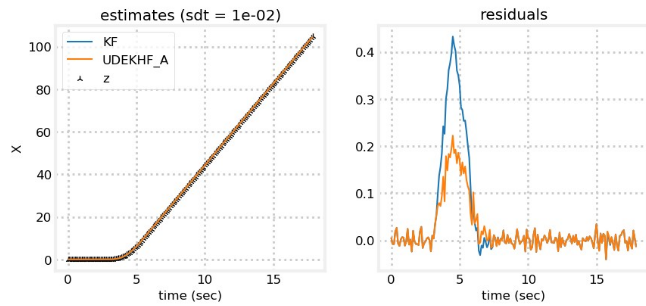
UDERHF_A



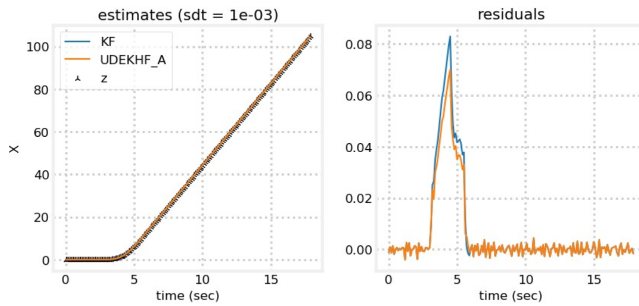
a



b



c



d

Figure 3: The UDEKHF_A and the Kalman filter operation at various values of the observation noise variance.

The results of the comparison of the algorithms are presented in Fig. 3. The estimate given by the UDEKHF_A is more accurate than those given by the Kalman filter under the same conditions (the maximum residual of the hybrid filter is less).

UDEKHF_B

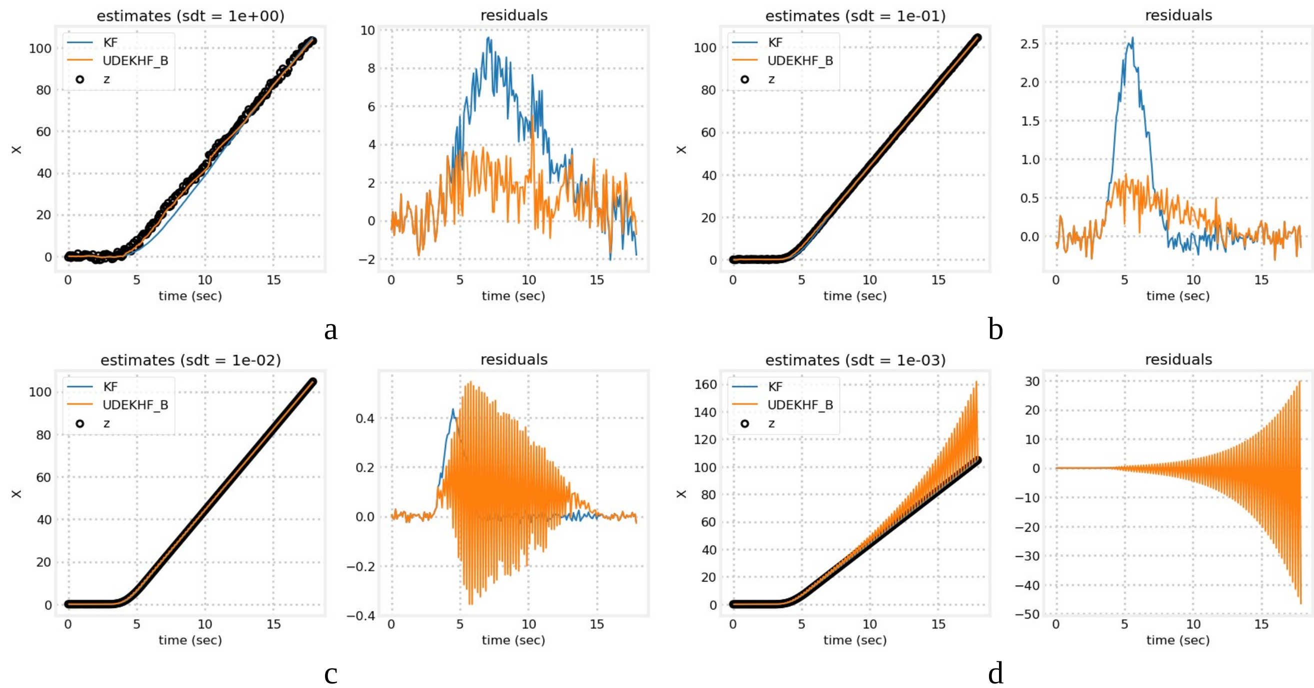


Figure 5: The UDEKHF_B and the Kalman filter operation at various values of the observation noise variance.

The results of the UDEKHF_B are presented in Fig. 5, this algorithm is not stable at low values of the observation noise variance. At sufficiently high values of the observation noise variance, the algorithm is stable and gives more accurate estimates than the Kalman filter.

EKHF_C and UDEKHF_C

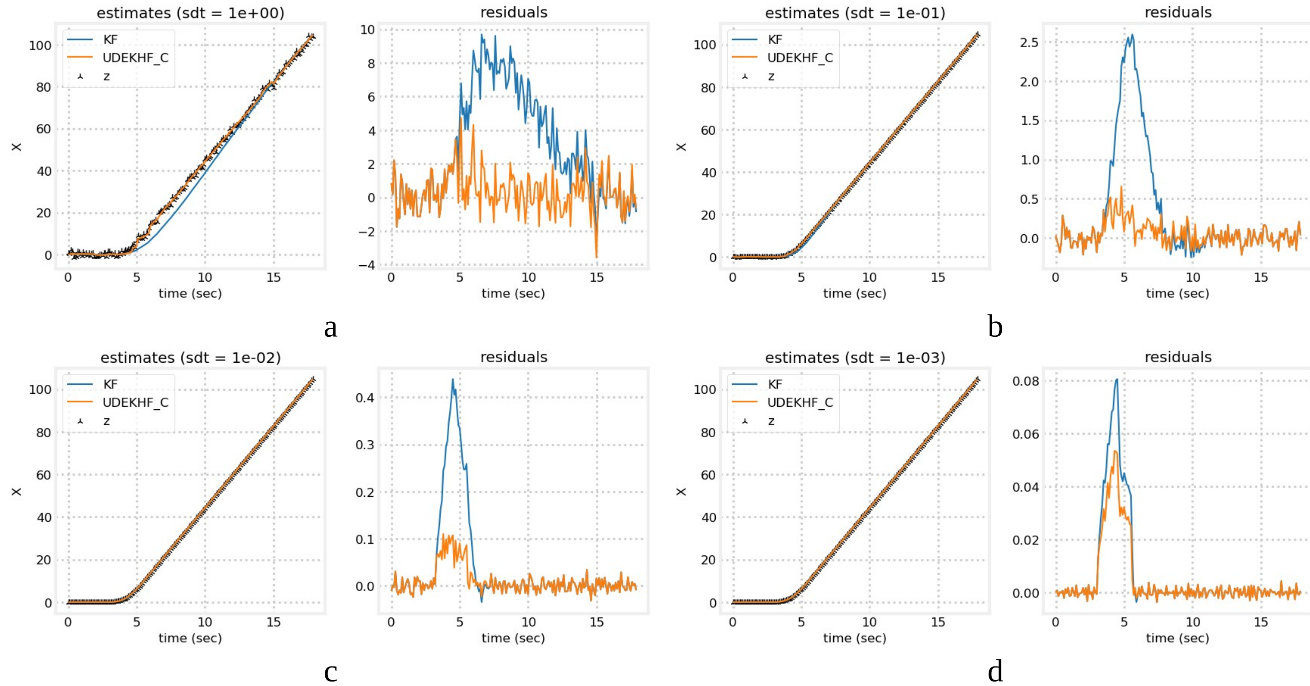


Figure 6: The operation of the EKHF_C and the UDEKHF_C compared to the Kalman filter at different values of the variance of observation noise.

The results of the EKHF_C and the UDEKHF_C are presented in Fig. 6, these filtering algorithms during maneuvers give more accurate estimates than the Kalman filter. The maximum residuals given by the EKHF_C and the UDEKHF_C are lower than that of the Kalman filter.

EKHF_D and UDEKHF_D

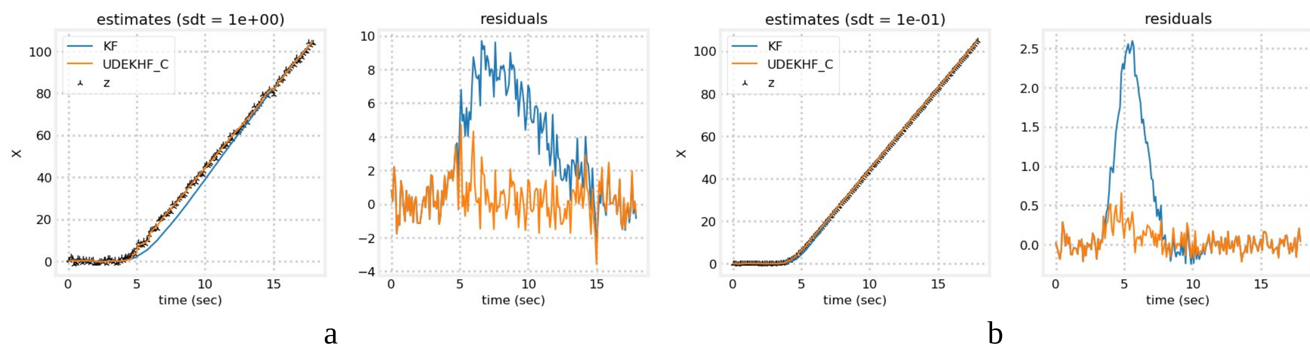


Figure 5: The UDEKHF_D and the Kalman filter operation at various values of the observation noise variance.

Like UDEKHF_B, the EKHF_D and the UDEKHF_D algorithms are not stable at low values of the observation noise variance. The results of EKHF_D are presented in Fig. 7, the UDEKHF_D algorithm exhibits similar to the EKHF_D behavior.

Two-dimensional case

In two dimensional case we measure both coordinates of the ojbect.

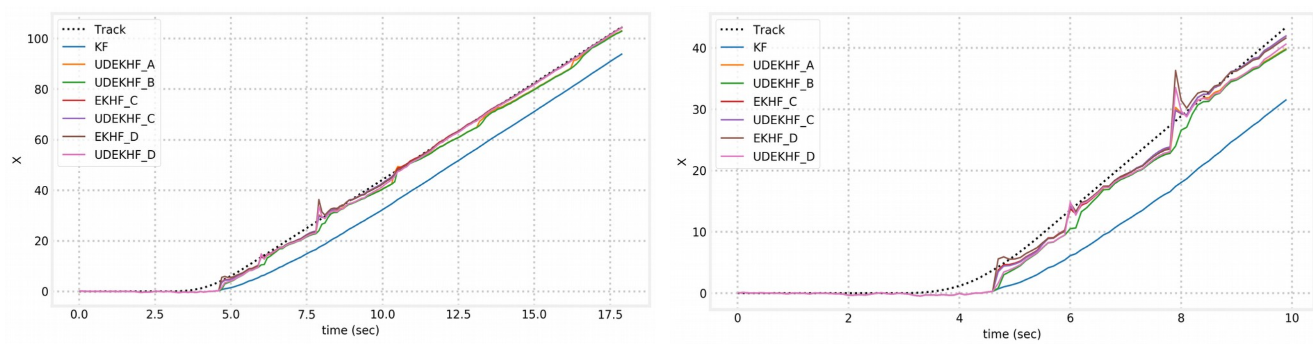
The state transition matrix is:

$$F_k = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ;$$

The observation matrix is:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad ;$$

The results of modeling in two time scales are shown in fig. 8:



a b

Figure 8: The objects X-coordinate and filtering results.

Here and below we will use an X coordinate to to make the results comparable with one-dimensional case.

It is obvious that the Kalman filter diverges, during and after the maneuver of the observed object.

Hybrid filters provide more accurate estimates, with the exception of spikes from EKHF_D and UDEKHF_D. The UD factorized versions of hybrid filters give similar estimates.

One can see when the filters make adaptive corrections upon divergence detection events.

The most accurate estimates are given by the EKHF_C and UDEKHF_C algorithms, in addition, they needed fewer corrections to go to the steady state.

As in the one-dimensional case, we consider the residuals of different versions of hybrid filters with the residuals of the Kalman filter at different values of the variance of the observation noise.

UDEKHF_A

The results of the algorithm for a two-dimensional case are presented in Fig.9.

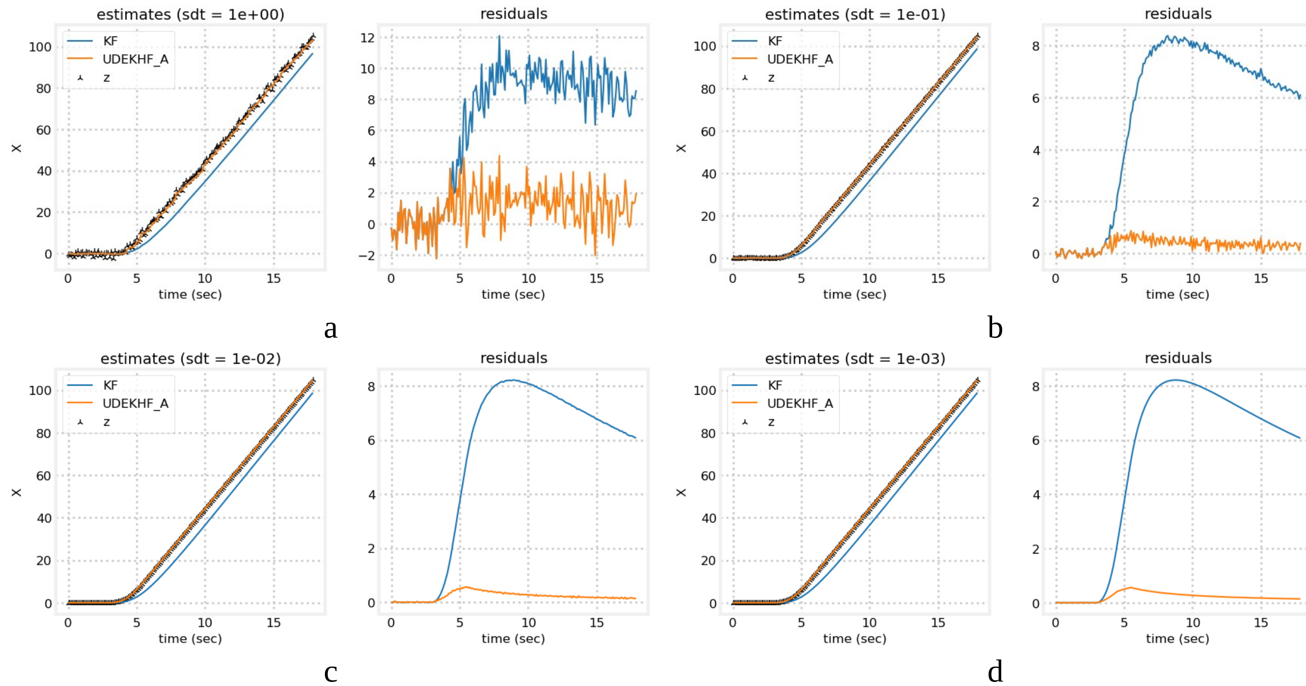


Figure 9: The UDEKHF_A and the Kalman filter operation at various values of the observation noise variance (two-dimensional case).

As in the one-dimensional case, the filter works stably under different conditions and gives more accurate estimates than the Kalman filter.

UDEKHF_B

In fig. 10 shows the results of the UDEKHF_B algorithm.

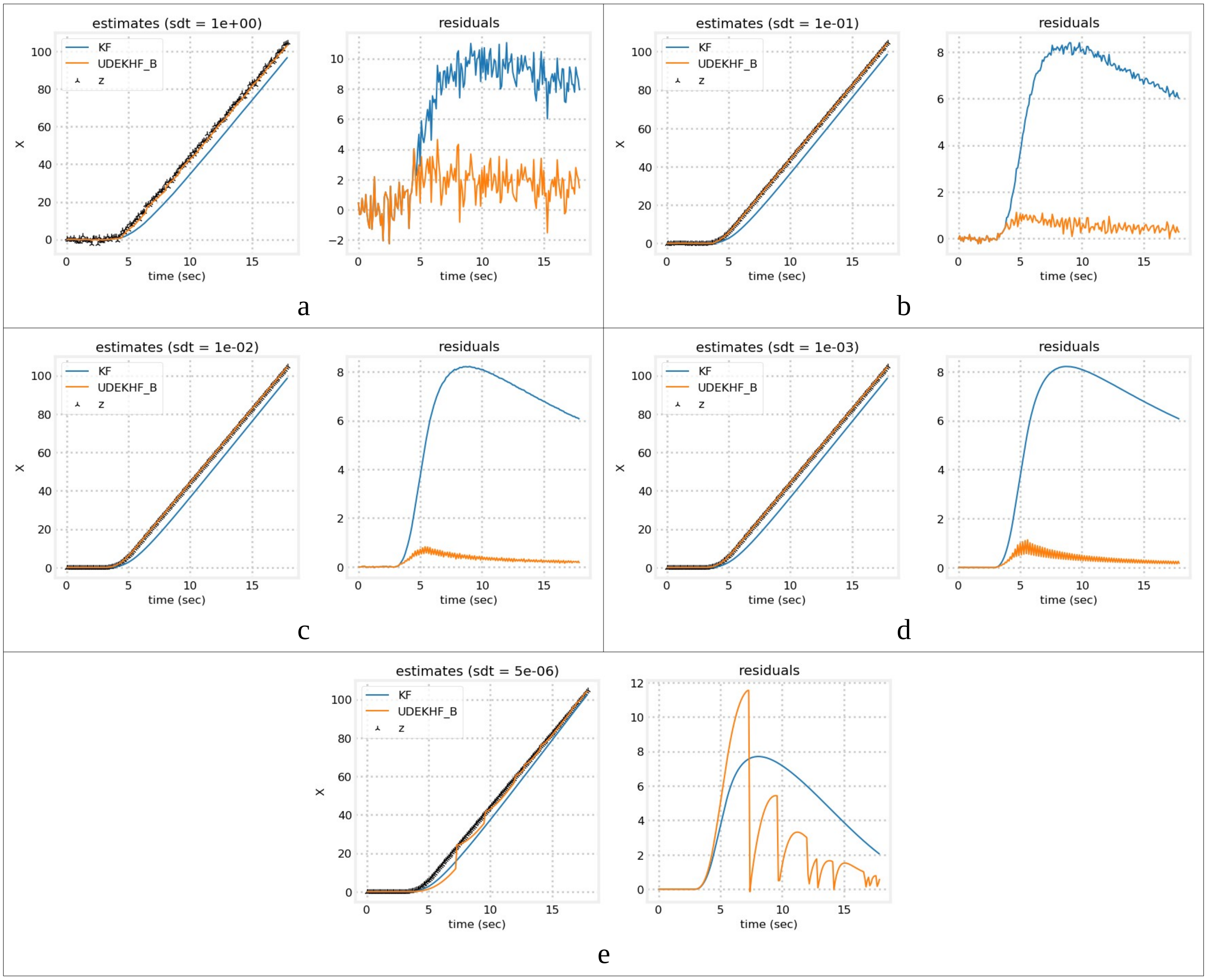


Figure 10: The UDEKHF_B and the Kalman filter operation at various values of the observation noise variance (two-dimensional case).

As in the one-dimensional case, the filter is unstable (Fig. 10c-e) and diverges (Fig. 10e) at relatively low values of the observation noise variance.

EKHF_C and UDEKHF_C

The results of the EKHF_C algorithm for the two-dimensional case are presented in Fig. 11.

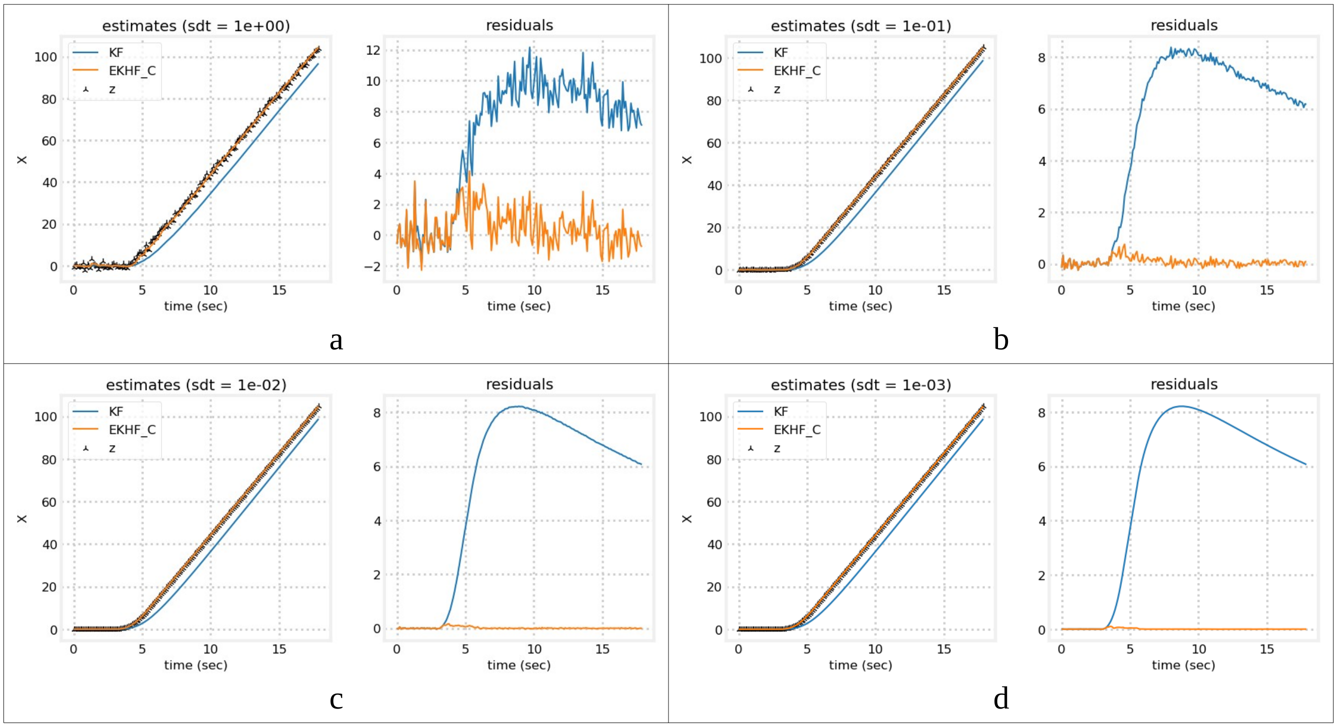


Figure 11: The EKHF_C operation at different values of the observation noise variance (two-dimensional case).

As in the one-dimensional case, the filter is stable under different conditions and gives more accurate estimates than the Kalman filter, it quickly reaches the steady state.

The UDEKHF_C algorithm works similarly to the EKHF_C.

EKHF_D and UDEKHF_D

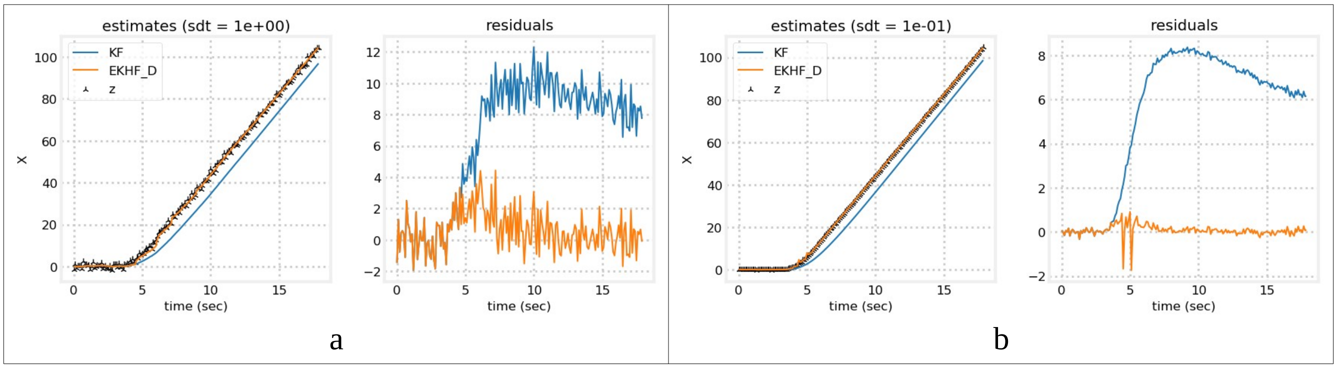


Figure 12: The EKHF_D operation at different values of the observation noise variance (two-dimensional case).

As in the one-dimensional case, the EKHF_D and UDEKHF_D algorithms are not stable at low values of the observation noise variance. The results of EKHF_D are presented in Fig. 12, the UDEKHF_D algorithm exhibits a similar behavior.

Conclusion

During the computational experiment, it was found that hybrid filtering algorithms using a posteriori residuals for adaptive correction showed unstable operation. In addition, they are more difficult to implement than algorithms using a priori residuals.

Algorithms developed based on an approach with a restriction on the parameters of the H_∞ filter are more simple, give more accurate estimates, make fewer adaptive corrections, converge faster. In addition, they can be easily implemented as a modifications to existing implementations of Kalman filters.

Thus, for implementation in embedded systems, we recommend the EKHF_C for cases with time-varying parameters of process noise and observation noise and the UDEKHF_C algorithm for the case with constant parameters of process noise and observation noise.

Ссылки

1. R. Kalman, “Contributions to the theory of optimal control,” Boletin de la Sociedad Matematica Mezicana, 5 , pp. 102-119 (1960).
2. J. Bellantoni and K. Dodge, “A square root formulation of the Kalman-Schmidt filter,” AIAA Journal, 5, pp. 1309-1314, 1967.
3. L.MacGee and S.Schmidt, “Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry,” NASA Technical Memorandum 86847
4. R. Fitzgerald, “Divergence of the Kalman filter,” IEEE Transactions on Automatic Control, AC-16, pp. 736-747 (December 1971).
5. R. Banavar, “A game theoretic approach to linear dynamic estimation,” Doctoral Dissertation, University of Texas at Austin, May 1992.
6. S. Kosanam and D. Simon, “Kalman filtering with uncertain noise covariances,” Intelligent Systems and Control, Honolulu, Hawaii, pp. 375-379, August 2004.
7. W. Haddad, D. Bernstein, and D. Mustafa, “Mixed-norm H_2/H_∞ regulation and estimation: The discrete-time case”, Systems and Control Letters, 16, pp. 235-247 (1991).
8. A.V. Chernodarov, V.N. Kovregin, “An H ∞ Technology for the Detection and Damping of Divergent Oscillations in Updatable Inertial Systems**”,** Proc. of the International Conference “Physics and Control”.– St.Petersburg, 2003, ed.1, pp. 121-126.
9. Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, “Estimation with Application and Tracking and Navigation”, John Wiley & Sons, 2001.
10. G. Bierman, “Factorization Methods f o r Discrete Sequential Estimation”, Academic Press, San Diego, California, 1977.
11. R.R. Labbe, “Kalman and Bayesian Filters in Python”, ch. 14, https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python, 2015.
12. *R.A. Horn, C.R Johnson, Charles R. “Matrix Analysis (2nd ed.).”, Cambridge University Press, 2013, ISBN 978-0-521-38632-6.*