We are IntechOpen,
the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Robust Kalman-type Filtering in Positioning Applications

Tommi Perälä
Tampere University of Technology
Finland

1. Introduction

Positioning refers to the estimation of one's location by combining many different sources of information. This information is usually obtained in the form of measurements, which may be, for example, pseudorange or deltarange measurements from satellites. In addition, various wireless networks on Earth, for example, cellular networks, WLAN or Bluetooth provide means for positioning in the form of range measurements, received signal strength indicators and sector information. Portable positioning devices may also contain inertial measurement units that provide information about the movements of the user.

The positioning problem may be formulated as a Bayesian filtering problem. The measurements are related to the position of the user, and the relation is approximately known. In addition, there is a model that describes the process dynamics. The measurements are obtained at discrete time intervals, and the process dynamics are also discretized.

The system may be described mathematically as follows. Let x_k denote the stochastic state vector at time step k, and let y_k be the measurement vector. The system is governed by the following equations:

$$y_k = h_k(x_k) + v_k \tag{1}$$

$$x_{k+1} = g_k(x_k) + w_k \tag{2}$$

Functions h_k and g_k are the measurement and state update functions, respectively. The measurement noise v_k and the state update noise w_k are assumed to be white processes. The initial state is denoted by x_0. The noises and the initial state are assumed to be mutually independent.

Using these assumptions, we want to find the conditional probability density function conditioned on all realized measurements $p_{x_k^+}(x_k|y_{1:k})$, which is also called the posterior probability density function. The set of all realized measurements up to time step k is denoted by $y_{1:k} = \{y_i | i = 1, 2, \ldots, k\}$. The posterior density function contains all the information of the system up to time step k. From this density function we can compute an estimate of the state with respect to any optimality criterion.

Using the above assumptions and the Bayes' rule, the posterior density function may be written as

$$p_{x_k^+}(x_k|y_{1:k}) = \frac{p_{y_k}(y_k|x_k)p_{x_k^-}(x_k|y_{1:k-1})}{p_{y_k^-}(y_k|y_{1:k-1})} \ , \qquad (3)$$

where we have denoted the set of past measurements by $y_{1:k-1}$ (Ristic et al., 2004). The conditional probability density function $p_{x_k^-}(x_k|y_{1:k-1})$ is called the prior density function, and it contains all the information about the system before using the measurements of the current time step. The measurement likelihood function $p_{y_k}(y_k|x_k)$ is given by the measurement model, and it is used to incorporate the measurements of the current time step to the estimate. The normalization factor

$$p_{y_k^-}(y_k|y_{1:k-1}) = \int p_{y_k}(y_k|x_k)p_{x_k^-}(x_k|y_{1:k-1})dx_k \qquad (4)$$

is sometimes called the predicted measurement density or the innovation density. The expression (3) for the posterior density function allows the recursive computation of the conditional probability density function, which is very convenient regarding the computational and memory requirements of the algorithm.

In general, it is difficult to find an analytical expression for the posterior density function. However, if the state update function $g_k(x_k) = G_k x_k$ and the measurement function $h_k(x_k) = H_k x_k$ are linear, and the state model noise and the measurement noise are modeled as zero mean Gaussians with covariance matrices Q_k and R_k, respectively, and the initial state is Gaussian with mean x_0^+ and covariance matrix P_0^+, the posterior density function is also Gaussian. Only little computation is required in order to compute the posterior density function, and its parameters are given by the famous Kalman Filter relations (5) − (9).

The prior mean is obtained by applying the linear state update function to the posterior mean of the previous time step:

$$x_k^- = E(x_k|y_{1:k-1}) = G_{k-1}x_{k-1}^+ \ . \qquad (5)$$

The prior covariance matrix is obtained from the posterior covariance matrix of the previous time step:

$$P_k^- = V(x_k|y_{1:k-1}) = G_{k-1}P_{k-1}^+ G_{k-1}^T + Q_{k-1} \ . \qquad (6)$$

The posterior mean is obtained by adding a linear transformation of the innovation to the prior mean:

$$x_k^+ = x_k^- + K_k(y_k - H_k x_k^-) \ . \qquad (7)$$

The posterior covariance becomes:

$$P_k^+ = (I - K_k H_k)P_k^- \ , \qquad (8)$$

and the Kalman gain is defined as:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \qquad (9)$$

However, in practice the relation between the state and the measurements is rarely linear. Therefore, non-linear extensions of the Kalman Filter have been studied, and in this chapter we will concentrate on one of them, namely, the Extended Kalman Filter (EKF).

EKF linearizes the non-linear measurement and state update functions at the prior mean of the current time step and the posterior mean of the previous time step, respectively. The resulting algorithm is very similar to the Kalman Filter. However, EKF does not solve the posterior density function exactly, but instead, approximates the posterior density function with a Gaussian density function.

EKF has been studied in positioning applications and it is shown to perform poorly when the non-linearities are significant. (Ali-Löytty et al., 2005) show that when using measurements from satellites, the non-linearities do not degrade the performance of EKF, but when using range measurements from terrestrial base stations, EKF may easily veer away from the true solution and get stuck in a wrong solution branch. One reason for this kind of behavior is that the true posterior density might be multimodal, and EKF cannot know which peak represents the correct position of the user. The problem of multimodality has been addressed using Gaussian Mixture Filters (GMF), and they have been shown to perform quite well (Ali-Löytty & Sirola, 2007a); (Ali-Löytty & Sirola, 2007b);(Ali-Löytty, 2008). GMFs approximate the posterior density as a sum of Gaussian densities where each component is an individual EKF. By using GMFs, the problems caused by the non-linearities may be overcome.

Although GMFs perform quite well even in highly non-linear cases, they are still based on the assumption of Gaussian measurement noise. It has been shown that filters based on the assumption of Gaussian noise may perform poorly in cases where the measurement noise is non-Gaussian, and so-called blunder measurements occur (Perälä & Piché, 2007). In positioning applications, blunder measurements occur, for example, due to signal reflections and multipath effects.

In this chapter, we present two methods for making EKF more robust against blunder measurements. The robust modifications of EKF may also be incorporated in Gaussian Mixture Filters that are based on EKF. In the first method, the measurement covariance matrix is modified based on the differences between the predicted and realized measurements, which are also called innovations. The modification is done using weight functions that are derived from M-estimators. In the second method, the predicted measurement density is approximated with a non-Gaussian density and the likelihood score of the corresponding density is used instead of the Gaussian likelihood score that appears in the Kalman Filter. Using these modifications, we try to obtain filters that are robust against blunder measurements.

2. Extended Kalman filter

In this section we present the algorithm of the Extended Kalman Filter and introduce some terminology needed in the following sections. We assume that the state model is linear and only concentrate on linearizing the measurement model. Consider the non-linear measurement equation:

$$y_k = h_k(x_k) + v_k .$$ (10)

The first order Taylor series approximation of the measurement function at the prior mean is

$$h_k(x_k) \approx h_k(x_k^-) + H_k(x_k - x_k^-) ,$$ (11)

where the Jacobian of the measurement function is

$$H_k = \left. \frac{\partial h_k(x_k)}{\partial x_k} \right|_{x_k = x_k^-} .$$ (12)

Denoting

$$\Delta y_k = y_k - h_k(x_k^-)$$ (13)

and

$$\Delta x_k = x_k - x_k^- ,$$ (14)

an approximate measurement equation may be written as

$$\Delta y_k = H_k \Delta x_k + v_k .$$ (15)

Applying the Kalman Filter to this linearized measurement model, the posterior mean becomes

$$x_k^+ = x_k^- + K_k(y_k - h_k(x_k^-))$$ (16)

and the posterior covariance is given by

$$P_k^+ = (I - K_k H_k) P_k^-$$ (17)

where $K_k = P_k^- H_k (H_k^T P_k^- H_k + R_k)^{-1}$ is the Kalman gain matrix. The only differences to the Kalman Filter are that the innovation is computed using the non-linear measurement function, and that H_k is the Jacobian of the measurement function.

The innovation $s_k := y_k - h_k(x_k^-)$, which appears in the posterior mean recursion, describes how much the measurements differ from those expected when we think the user's state is the prior mean x_k^-, which before taking the measurement into account is our best estimate for the state. In EKF, the state is corrected by applying a linear gain to the innovation. The robust filters presented in this chapter are based on a gain that is computed differently. The next section discusses the score functions that are used later to compute the gain.

3. Score function selection

The robust Kalman Filters discussed in this chapter are essentially based on embedding the score function of a robust M-estimator into the Kalman Filter. We use Huber's concept of minimax robustness to find the robust M-estimators. (Huber, 1964) suggests the minimization of the maximum asymptotic variance of an estimator over a predefined class of densities F. The solutions to this problem are pairs (T^0, f^0), where T^0 is the most robust M-estimator and f^0 is called the least favorable density of the class F. We introduce two

classes for densities for which the minimax solution has been found, namely the ε-contaminated normal neighborhood and the p-point family.

3.1 Huber M-estimator

The ε-contaminated normal neighborhood was first proposed in (Huber, 1964) to be used in robust parameter estimation and it is defined as follows.

Definition 1 (ε-contaminated Normal Neighborhood) The set of density functions F_ε is called ε-contaminated normal neighborhood if $F_\varepsilon = \{(1-\varepsilon)\phi(x) + \varepsilon H(x) : H \in S\}$, where ϕ is the standard normal probability density function, S is the set of symmetrical probability density functions, and $0 \leq \varepsilon < 1$ is the known fraction of contamination.

Huber showed that the least favorable density of this class is Gaussian in the middle, but has exponential tails. We denote this density by f_ε^0 and it is given by

$$f_\varepsilon^0(t) = \begin{cases} \dfrac{(1-\varepsilon)}{\sqrt{2\pi}}\exp\left(-\tfrac{1}{2}t^2\right), & |t| \leq k \\[2ex] \dfrac{(1-\varepsilon)}{\sqrt{2\pi}}\exp\left(\tfrac{1}{2}k^2 - k|t|\right), & |t| > k \end{cases} . \tag{18}$$

The connection between the threshold parameter k and the amount of contamination ε is given by

$$\frac{2\phi(k)}{k} - 2\Phi(-k) = \frac{\varepsilon}{1-\varepsilon} , \tag{19}$$

where Φ is the standard normal cumulative distribution function. Usually this equation has to be solved numerically. The influence function of an M-estimator is defined as the negative likelihood score of the least favorable density $\psi^0(t) = -\partial \ln f^0(t)/\partial t$, and for the Huber's M-estimator it is

$$\psi_\varepsilon^0(t) = \begin{cases} t, & |t| \leq k \\ k \cdot \text{sign}(t), & |t| > k \end{cases} . \tag{20}$$

The weight function of an M-estimator is defined as $\omega(t) = \psi(t)/t$, $t \neq 0$ and $\omega(0)$ is chosen so that $\omega(t)$ is continuous. For the Huber's M-estimator the weight function is

$$\omega_\varepsilon^0(t) = \begin{cases} 1, & |t| \leq k \\ \dfrac{k}{|t|}, & |t| > k \end{cases} \tag{21}$$

The influence function and the weight function are needed in the robust filters that are presented in the following sections.

3.2 p-point M-estimator

Another interesting family of densities, namely, the p-point family is used in robust parameter estimation in (Martin & Masreliez, 1975); (Masreliez & Martin, 1977), and it is defined as follows.

Definition 2 (p-point Family) The set of probability density functions F_p is called a p-point family if $F_p = \left\{ f \mid \int_{-\infty}^{-y_p} f(x)dx = p/2 = \Phi(-y_p), f \text{ symmetric and continuous at } \pm y_p \right\}$.

The inclusion of the restriction that F_p contains the standard normal cumulative distribution function $\Phi(x)$ is for standardization purposes, that is, to ensure that F_p is in the neighborhood of the standard normal density.

(Masreliez & Martin, 1977) show that the least favorable density f_p^0 of F_p is

$$f_p^0(t) = \begin{cases} K\cos^2\left(\dfrac{t}{2c_m y_p}\right), & |t| \le y_p \\[3mm] K\cos^2\left(\dfrac{1}{2c_m}\right) \cdot \exp\left(\dfrac{2K}{p}\cos^2\left(\dfrac{1}{2c_m}\right)(y_p - |t|)\right), & |t| > y_p \end{cases}, \tag{22}$$

where K is related to p by the following equation

$$K = \frac{1-p}{y_p(1+c_m \sin(1/c_m))}. \tag{23}$$

For each p there exists s_m that minimizes the asymptotic variance of the estimator. The minimizing value of c_m satisfies the equation

$$2c_m - p\left(1 + \tan^2(1/2c_m)\right)\left(2c_m + \tan(1/2c_m)\right) = 0. \tag{24}$$

The influence function of the least favorable density of the p-point family F_p is

$$\psi_p^0(t) = -\frac{\partial}{\partial t}\ln f_e^0(t) = \begin{cases} \dfrac{1}{c_m y_p}\tan\left(\dfrac{t}{2c_m y_p}\right), & |t| \le y_p \\[3mm] \dfrac{1}{c_m y_p}\tan\left(\dfrac{1}{2c_m}\right) \cdot \text{sign}(t), & |t| > y_p \end{cases} \tag{25}$$

and the weight function is

$$\omega_p^0(t) = \begin{cases} \dfrac{1}{c_m y_p t}\tan\left(\dfrac{t}{2c_m y_p}\right), & |t| \le y_p \\[3mm] \dfrac{1}{c_m y_p |t|}\tan\left(\dfrac{1}{2c_m}\right), & |t| > y_p \end{cases}. \tag{26}$$

3.3 Damped Hampel M-estimator

(Hampel et al., 1981) propose other M-estimators for robust estimation. However, these M-estimators belong to the class of redescending M-estimators, i.e., they have finite rejection point. M-estimators with finite rejection points discard certain measurements that are assumed to be too far from the true parameter. In hybrid positioning, we cannot always afford to discard measurements, and thus we will present here a modified version of the Hampel's three parts redescending M-estimator, namely the Damped Hampel M-estimator. The influence function of the Damped Hampel M-estimator is defined as

$$\psi_{DHA}(t) = \begin{cases} t, & |t| \le k_1 \\ k_1 \cdot \text{sign}(t), & k_1 < |t| \le k_2 \\ \dfrac{k_1 k_2^r}{|t|^r} \cdot \text{sign}(t), & |t| > k_2 \end{cases} \tag{27}$$

where we assume that $r \ge 0$. The corresponding weight function is defined as

$$\omega_{DHA}(t) = \begin{cases} 1, & |t| \le k_1 \\ \dfrac{k_1}{|t|}, & k_1 < |t| \le k_2 \\ \dfrac{k_1 k_2^r}{|t|^{r+1}}, & |t| > k \end{cases} \tag{28}$$

By setting $k_1 = k_2 = k$ and $r = 0$, we see that the Huber's M-estimator is obtained as a special case of the Damped Hampel M-estimator.

The Damped Hampel M-estimator is not derived from a least favorable density function, but instead, it is obtained by designing a piecewise weighting for the observations in a continuous manner. It is possible to calculate the density function by using the definition of the influence function, but for our purposes it is enough to know the influence and weight functions.

The Damped Hampel M-estimator presented here is only one example of piecewise influence functions. It is easy to design a variety of different piecewise influence functions and try to find the best one for the problem at hand using optimization techniques. Although the Huber's M-estimator and the p-point M-estimator are most robust in minimax sense, they require some knowledge of the distribution of the errors. Usually we do not have such knowledge, and thus any M-estimator that has proven to perform well in testing could be used.

4. Re-weighted extended Kalman filter

In the Kalman Filter, the posterior density function is Gaussian, and thus, the posterior mean estimate is the value that maximizes the probability density function:

$$x_k^+ = \arg\max_{x_k} p_{x_k^+}(x_k | y_{1:k}) . \tag{29}$$

Using the Bayes' theorem, inserting the prior density function and the measurement likelihood function, and noting that constant multipliers do not affect the maximization problem, (29) may be written as

$$x_k^+ = \arg\max_{x_k} \left(\exp\left(-\tfrac{1}{2}\|x_k - x_k^-\|^2_{(P_k^-)^{-1}}\right) \cdot \exp\left(-\tfrac{1}{2}\|y_k - H_k x_k\|^2_{(R_k)^{-1}}\right) \right) . \tag{30}$$

Since the exponential function is monotonically increasing, an equivalent problem is the following minimization problem.

$$x_k^+ = \arg\min_{x_k}\left(\left\| x_k - x_k^- \right\|_{(P_k^-)^{-1}}^2 + \left\| y_k - H_k x_k \right\|_{(R_k)^{-1}}^2 \right). \tag{31}$$

Leaving the subscripts out for simplicity, and denoting $n = (P^-)^{-\frac{1}{2}}(x - x^-)$ and $l = R^{-\frac{1}{2}}(y - Hx)$, where the square root is the symmetric square root of a matrix, (31) may be written as

$$x_k^+ = \arg\min_{x_k}\left(\sum_{i=1}^{n_x} n_i^2 + \sum_{j=1}^{n_y} l_j^2 \right). \tag{32}$$

As may be seen in equation (32), the posterior mean estimate of the Kalman Filter is a recursive solution to an ordinary least squares problem. The idea of minimizing the sum of squared errors is tempting since the solution may be computed efficiently. However, the least squares method is not robust, and therefore we modify the quadratic cost function in the second sum by a convex function ρ. The ρ-function is chosen so that the derivative of the score function is the influence function of an M-estimator introduced in Section 3. The minimization problem is changed to

$$x_k^+ = \arg\min_{x_k}\left(\sum_{i=1}^{n_x} n_i^2 + \sum_{j=1}^{n_y} \rho(l_j) \right). \tag{33}$$

The aim is to make the measurement model more robust, and therefore, only the second sum is modified. Since the score function is assumed to be convex, the minimum is found by setting the gradient of the sum to zero

$$\nabla_x \left(\sum_{i=1}^{n_x} n_i^2 + \sum_{j=1}^{n_y} \rho(l_j) \right) = 0 . \tag{34}$$

Denoting the derivative of the score function ρ by ψ, equation (34) may be written as

$$\sum_{i=1}^{n_x} 2n_i (P^-)^{-\frac{1}{2}} e_i + \sum_{j=1}^{n_y} \psi_j(l_j) H^T R^{-\frac{1}{2}} e_j = 0 , \tag{35}$$

where e_j is a vector whose jth element is one and the others are zeros. Thus, the minimization problem is equal to a vector equation

$$(P^-)^{-\frac{1}{2}}\left[n_1, \cdots, n_{n_x} \right]^T + H^T R^{-\frac{1}{2}}\left[\psi_1(l_1), \ldots, \psi_{n_y}(l_{n_y}) \right]^T = 0 . \tag{36}$$

Since ψ is, in general, a non-linear function, equation (36) has to be solved numerically. However, we want to preserve the computationally convenient properties of KF, and thus we proceed as in (Durovic & Kovacevic, 1999) and (Carosio et. al, 2005), and replace the equation by a linear approximation. Thus, we write

$$(P^-)^{-\frac{1}{2}} n + H^T R^{-\frac{1}{2}}\left[\omega_1(l_1^-) l_1, \ldots, \omega_{n_y}(l_{n_y}^-) l_{n_y} \right]^T = 0 . \tag{37}$$

where the weights ω_i are given by

$$\omega_i = \begin{cases} \dfrac{\psi_i(l_i^-)}{l_i^-} & l_i^- \neq 0 \\ 1 & l_i^- = 0 \end{cases} \tag{38}$$

and

$$l^- = R^{-\frac{1}{2}}(y - Hx^-). \tag{39}$$

Now equation (37) may be written as

$$(P^-)^{-\frac{1}{2}}n + H^T R^{-\frac{1}{2}} W_\psi l = 0, \tag{40}$$

where W_ψ is a diagonal matrix with diagonal elements $\omega_1(l_1^-), \ldots, \omega_{n_y}(l_{n_y}^-)$. Inserting $n = (P^-)^{-\frac{1}{2}}(x - x^-)$ and $l = R^{-\frac{1}{2}}(y - Hx)$ yields

$$(P^-)^{-1}(x - x^-) + H^T R^{-\frac{1}{2}} W_\psi R^{-\frac{1}{2}}(y - Hx) = 0. \tag{41}$$

We define the re-weighted measurement covariance matrix as

$$R_W = (R^{-\frac{1}{2}} W_\psi R^{-\frac{1}{2}})^{-1}. \tag{42}$$

This matrix exists assuming that W_ψ is positive definite, which is true if $\omega_i > 0, \forall i$. Inserting (42) into (41) yields

$$(P^-)^{-1}(x - x^-) + H^T R_W^{-1}(y - Hx) = 0, \tag{43}$$

which is the solution for the minimization problem

$$x^+ = \arg\min_x \left(\|x - x^-\|^2_{(P^-)^{-1}} + \|y - Hx\|^2_{(R_W)^{-1}} \right). \tag{44}$$

Equation (44) is similar to equation (31), which was derived from the posterior mean estimate of KF. The only difference is that the measurement covariance matrix R is replaced by the weighted measurement covariance matrix R_W. Thus, the solution of (44) is obtained using the posterior mean relation of KF. The measurement update recursions for the posterior mean estimate may be then written as:

$$x^+ = x^- + K_W(y - Hx^-), \tag{45}$$

Where

$$K_W = P^- H^T (HP^- H^T + R_W)^{-1}, \tag{46}$$

and the posterior covariance estimate becomes

$$P^+ = (I - K_W H)P^-. \tag{47}$$

The Re-weighted Kalman Filter derived in this section consists of computing the transformed innovation l^- and weighting the measurement covariance matrix using the

weight function of an M-estimator. The prior covariance is left intact since only the measurement model is modified. Thus, the filter derived here may be considered as a robust Kalman Filter, which modifies the given measurement covariances according to the innovations so that bigger transformed innovations result in bigger variances. A drawback of this method is that if the prior mean estimate x^- is far away from the true state, uncorrupted measurements might get weighted down, which could result in bad filter performance.

The filter derived here is for linear systems only. Thus we want to extend the filter for non-linear problems using the ideas of EKF. The only difference is that the innovations are computed using the non-linear measurement function, and that H is the Jacobian of the measurement function computed at the prior mean. The resulting filter is called the Re-weighted Extended Kalman Filter (REKF).

5. Approximate Bayesian extended Kalman filter

In the previous section KF was interpreted as a recursive least-squares algorithm, and was "robustified" by replacing the quadratic cost with (33). In this section we present an alternative approach that is directly based on the Bayesian interpretation of KF presented in Section 1. Consider a linear transformation matrix

$$T_k = (H_k P_k^- H_k^T + R_k)^{-\frac{1}{2}} , \qquad (48)$$

where H_k, P_k^- and R_k are the linear measurement function, the prior covariance matrix and the measurement covariance matrix, respectively, that appear in KF. The inverse exists and is symmetric since $H_k P_k^- H_k^T + R_k$ is symmetric and positive definite.

Denote the innovation as $s_k := y_k - H_k x_k^-$. The mean of the innovation is $E(s_k) = 0$ and the covariance matrix is $V(s_k) = H_k P_k^- H_k^T + R_k$. Thus, the mean of the transformed innovation $r_k = T_k s_k$ is $E(r_k) = 0$ and the covariance matrix $V(r_k) = I$. Now consider the posterior mean estimate of KF

$$x_k^+ = \int x_k p_{x_k^+}(x_k | y_{1:k}) dx_k . \qquad (49)$$

Using the Bayes' rule this may be written as

$$x_k^+ = \int x_k \frac{p_{x_k^-}(x_k | y_{1:k-1}) p_{y_k}(y_k | x_k)}{p_{y_k^-}(y_k | y_{1:k-1})} dx_k , \qquad (50)$$

which may be written after some algebra as

$$x_k^+ = x_k^- - \frac{1}{p_{y_k^-}(y_k | y_{1:k-1})} P_k^- \int p_{y_k}(y_k | x_k)(-(P_k^-)^{-1}(x_k - x_k^-)) p_{x_k^-}(x_k | y_{1:k-1}) dx_k . \qquad (51)$$

If the posterior density is approximated at every time step with a Gaussian, the prior density is also Gaussian since the state update function is linear and the noise in the state equation is Gaussian and independent of the state. By noting that for Gaussian prior it holds that

$$\nabla_{x_k} p_{x_k^-}(x_k | y_{1:k-1}) = -(P_k^-)^{-1}(x_k - x_k^-) p_{x_k^-}(x_k | y_{1:k-1}) , \qquad (52)$$

the posterior mean may be written as

$$x_k^+ = x_k^- - \frac{1}{p_{y_k^-}(y_k|y_{1:k-1})} P_k^- \int p_{y_k}(y_k|x_k) \nabla_{x_k} p_{x_k^-}(x_k|y_{1:k-1}) dx_k \; . \tag{53}$$

Noting that it follows from the measurement model that $p_{y_k}(y_k|x_k) = p_{v_k}(y_k - H_k x_k)$, and integrating by parts yields

$$x_k^+ = x_k^- + \frac{1}{p_{y_k^-}(y_k|y_{1:k-1})} P_k^- \int p_{x_k^-}(x_k|y_{1:k-1}) \nabla_{x_k} p_{v_k}(y_k - H_k x_k) dx_k \; . \tag{54}$$

Because

$$\nabla_{x_k} p_{v_k}(y_k - H_k x_k) = -H_k^T \nabla_{y_k} p_{v_k}(y_k - H_k x_k) \; , \tag{55}$$

the posterior mean may be written as

$$x_k^+ = x_k^- - \frac{1}{p_{y_k^-}(y_k|y_{1:k-1})} P_k^- \int p_{x_k^-}(x_k|y_{1:k-1}) H_k^T \nabla_{y_k} p_{v_k}(y_k - H_k x_k) dx_k \; . \tag{56}$$

Changing the order of differentiation and integration yields

$$x_k^+ = x_k^- + P_k^- H_k^T (-\nabla_{y_k} \ln p_{y_k^-}(y_k|y_{1:k-1})) \; . \tag{57}$$

Noting that $y_k = T_k^{-1} r_k + H_k x_k^-$ it can be shown that

$$x_k^+ = x_k^- + P_k^- H_k^T T_k (-\nabla_{r_k} \ln p_{r_k^-}(r_k|y_{1:k-1})) \; . \tag{58}$$

Define the influence function $\psi(r_k)$ as the negative likelihood score of the transformed innovation density

$$\psi(r_k) = -\nabla_{r_k} \ln p_{r_k^-}(r_k|y_{1:k-1}) \; , \tag{59}$$

and insert into (54) to obtain

$$x_k^+ = x_k^- + P_k^- H_k^T T_k \psi(r_k) \; . \tag{60}$$

It is easy to see by straightforward calculation that a Gaussian innovation density, which results from the assumptions of KF, produces the familiar posterior mean update relation of KF. (Masreliez & Martin, 1975) study this kind of estimators and show that if the marginal densities of $p_{r_k^-}(r_k|y_{1:k-1})$ are symmetric densities in F, where $F = F_\varepsilon$ or $F = F_p$, the estimator covariance is bounded by

$$C_k \leq (I - K_k H_k E_F) P_k^- \; , \tag{61}$$

where $E_F = \int (\frac{\partial}{\partial t} \psi(t)) f_F^0(t) dt$, f_F^0 is the least favorable density in F, and K_k is the Kalman gain matrix. It can be shown that it is possible to come arbitrarily close to the bound (Masreliez & Martin, 1975). Thus, the upper bound is chosen as the posterior covariance estimate P_k^+ for ABKF.

For the Huber's M-estimator and the p-point M-estimator E_F is found easily by straightforward calculation. For the Huber's M-estimator it is

$$E_{F_\varepsilon} = (1 - \varepsilon)(1 - 2\Phi(-k)) , \tag{62}$$

and for the p-point M-estimator we get

$$E_{F_p} = (c_m y_p)^{-1} \left(1 - p \left(1 + \tan^2 \left((c_m y_p)^{-1} \right) \right) \right) . \tag{63}$$

For the Damped Hampel M-estimator the integral does not generally have an explicit form, but since the central part of the influence function is the same as in the Huber's M-estimator, we use E_{F_ε} to compute the posterior covariance estimates when using the Damped Hampel M-estimator.

The robust posterior estimates may be computed using (60) and (61), however this does not produce the posterior probability density function. However, we approximate it with a Gaussian density with the mean and covariance matrix as in (60) and (61). This is called the Approximate Bayesian Kalman Filter (ABKF).

In the case where the measurement function is non-linear, the above considerations can be applied in EKF. The only difference is that the innovations are computed using the non-linear measurement function, and that H_k is the Jacobian of the measurement function computed at the prior mean. The resulting non-linear extension is called the Approximate Bayesian Extended Kalman Filter (ABEKF).

6. Positioning example

We consider a positioning scenario where we use satellite pseudorange and deltarange measurements that are obtained at discrete time intervals. The state of the user consists of the 3-dimensional position and velocity vectors of the user and is denoted by $x = [(r^u)^T \ (v^u)^T]^T$. The pseudorange measurements may be written as

$$z_i^p = \left\| r_i^s - r^u \right\| + b^p + \varepsilon_i^p , \tag{64}$$

where the ith pseudorange measurement is denoted by z_i^p, r^u is the position of the user, r_i^s is the position of the ith satellite, b^p is the unknown clock bias and ε_i^p is Gaussian zero mean noise with variance $(\sigma_i^p)^2$. The deltarange measurements may be written as

$$z_i^d = \frac{(r_i^s - r^u)^T}{\left\| r_i^s - r^u \right\|}(v_i^s - v^u) + b^d + \varepsilon_i^d , \tag{65}$$

where the ith deltarange measurement is denoted by z_i^d, v^u is the velocity of the user, v_i^s is the velocity of the ith satellite, b^d is the unknown clock drift and ε_i^d is Gaussian zero mean noise with variance $(\sigma_i^d)^2$.

The positions and velocities of the satellites are assumed to be known, but clock bias and the clock drift that appear in (58) and (59) are unknown. They are, however, the same for all the satellites and thus we may deal with them by using so-called difference measurements. Therefore, we have to introduce the concept of difference mapping.

Definition 3 (Difference Mapping) A difference mapping D is a $(n_s - 1) \times n_s$-matrix with full column rank such that $D1 = 0$, where 1 is a vector of ones.

The difference mapping may be chosen to be, for example, $D = \begin{bmatrix} I & -1 \end{bmatrix}$. Denote

$$D_1 = \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}.$$

(66)

Thus, the measurement vector becomes

$$y = D_1 \begin{bmatrix} \|r_1^s - r^u\| \\ \vdots \\ \|r_{n_s}^s - r^u\| \\ \dfrac{(r_1^s - r^u)^T}{\|r_1^s - r^u\|}(v_1^s - v^u) \\ \vdots \\ \dfrac{(r_{n_s}^s - r^u)^T}{\|r_{n_s}^s - r^u\|}(v_{n_s}^s - v^u) \end{bmatrix} + D_1 \begin{bmatrix} b^p \\ \vdots \\ b^p \\ b^d \\ \vdots \\ b^d \end{bmatrix} + D_1 \begin{bmatrix} \varepsilon_1^p \\ \vdots \\ \varepsilon_{n_s}^p \\ \varepsilon_1^d \\ \vdots \\ \varepsilon_{n_s}^d \end{bmatrix} = D_1 \begin{bmatrix} \|r_1^s - r^u\| \\ \vdots \\ \|r_{n_s}^s - r^u\| \\ \dfrac{(r_1^s - r^u)^T}{\|r_1^s - r^u\|}(v_1^s - v^u) \\ \vdots \\ \dfrac{(r_{n_s}^s - r^u)^T}{\|r_{n_s}^s - r^u\|}(v_{n_s}^s - v^u) \end{bmatrix} + D_1 \begin{bmatrix} \varepsilon_1^p \\ \vdots \\ \varepsilon_{n_s}^p \\ \varepsilon_1^d \\ \vdots \\ \varepsilon_{n_s}^d \end{bmatrix}.$$

(65)

The Jacobian of the measurement function is

$$H = \frac{\partial h(x)}{\partial x}\Big|_{x = x^-} = D_1 \begin{bmatrix} -U & 0 \\ U_{v^s}^D - U_{v_u}^D & -U \end{bmatrix},$$

(66)

where $U_{v^s}^D$ is the derivative of the vector $\mathrm{diag}(UV^T)$ and V is a matrix of satellite velocities, $U_{v_u}^D$ is the derivative of the vector Uv_u, and U is a matrix whose rows are the unit vectors pointing from the user to the satellites. The derivation of the Jacobian is straightforward but tedious and is omitted here.

The measurement covariance matrix becomes

$$R = D_1 \begin{bmatrix} (\sigma_1^p)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & (\sigma_{n_s}^p)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & (\sigma_1^d)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & (\sigma_{n_s}^d)^2 \end{bmatrix} D_1^T,$$

(67)

7. Simulations and testing

The robust filters were implemented in MATLAB and tested in simulations and using real GPS data. In Section 7.1, the simulation setup is described and the results of the simulations are discussed. The tests using real GPS measurement data are discussed in Section 7.2.

7.1 Simulations

The simulation test bench was designed to produce dynamic test data similar to what could be expected in real world personal positioning scenarios. The main difference to real data is that in the simulations the true track and correct measurement and motion models are known.

The test process consisted of first generating a true track of 120 points with one second intervals using a velocity-restricted random walk model, where $\sigma_p^2 = (1.41 \frac{m}{s})^2 s^{-1}$ and $\sigma_a^2 = (0.316 \frac{m}{s}) s^{-1}$. Similar values have been used in (Ma, 2004) to model moving vehicles.

Next, a GPS constellation was simulated with an elevation mask and a shadowing profile that were set so that only a couple of satellites were visible at a time. Finally, noisy measurements were generated for each time step.

Satellite pseudorange and deltarange measurements were used with an average of 2.9 pseudorange and deltarange measurements per time step. Measurement variances were set to $(\sigma_i^p)^2 = ((0.1 + 2x_U)^2 \, m^2$ and $(\sigma_i^d)^2 = ((0.01 + 0.05x_U)^2 \, \frac{m^2}{s^2}$. The term x_U denotes the realization of a stochastic variable with standard uniform distribution $U(0,1)$. Altogether, 100 track and measurement sets were generated. These sets were generated using the Personal Navigation Filter Framework (Raitoharju et al., 2008).

Next, some additional noise was generated to the measurements according to different choices of the blunder probability p_b. For each measurement a sample from the standard uniform distribution $U(0,1)$ was drawn. If the realization of the sample was less than p_b, a realization of a sample from $U(-30\sigma, 30\sigma)$, where σ was the standard deviation of the corresponding measurement, was added to the measurement value. The blunder probabilities were chosen to be 0, 0.01, 0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.35 and 0.5.

The test tracks were filtered with the six robust filters described in this chapter, corresponding to two choices of the filter (ABEKF and REKF) and to three choices of the influence and weight functions (Huber (H), p-point (M) and Damped Hampel (DHA)). The parameters for the influence and weight functions used in the simulations and tests are presented in Table 1. Since DHA M-estimator is not derived from a minimax criterion, and thus does not correspond to any least favorable density presented in this chapter, using it in ABEKF is somewhat questionable. However, since DHA is essentially a generalization of the Huber M-estimator, we use E_F in DHA.

Name	Parameters
Huber	$k = 2.2$
p-point	$y_p = 2.2$
Damped Hampel	$k_1 = 2.2, k_2 = 3.3, r = 1$

Table 1. Estimator parameters used in simulations and testing

The mean and covariance of the posterior distribution were recorded at each time step and compared to the true track. For comparison, the data was also processed with EKF.

Figure 1 shows the mean error of different filters when using the Damped Hampel influence and weight functions. It can be seen that EKF works quite well also when the blunder probability gets bigger. This is not surprising since EKF should be optimal for linear measurements regardless of the density function of the error. Nonetheless, REKF still performs better than EKF. However, ABEKF starts to give meaningless estimates even with moderate blunder probabilities.
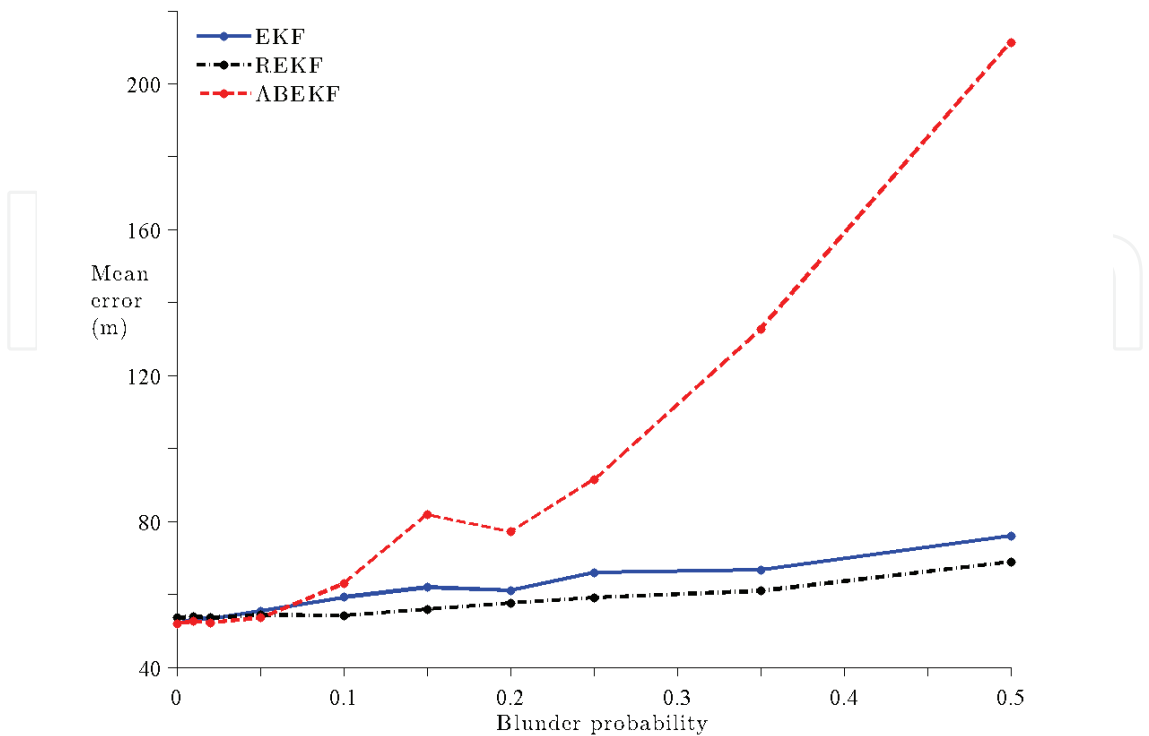
Fig. 1. Comparison of different filters in the simulations

The mean error (m), the 95% percentile of errors (m) and the frequency of inconsistent estimates (%) are presented in Table 2 for the blunder probabilities of 0, 0.1 and 0.25. The inconsistency was determined using the general inconsistency test with risk level 5% (Ali-Löytty et al., 2005), and it tells how often the error estimate was smaller than the actual error.

		$p_b = 0\%$			$p_b = 5\%$			$p_b = 25\%$		
		ME	95%	Inc.	ME	95%	Inc.	ME	95%	Inc.
		(m)	(m)	(%)	(m)	(m)	(%)	(m)	(m)	(%)
	H	52.2	287	0.0	53.5	281	0.1	59.7	300	1.5
REKF	M	52.2	285	0.0	53.8	281	0.0	60.9	302	2.0
	DHA	53.6	286	0.0	54.3	279	0.1	59.1	285	1.1
	H	51.9	284	0.0	53.4	277	0.3	64.9	313	10
ABEKF	M	53.0	275	0.0	54.4	279	0.1	62.6	292	7.2
	DHA	51.9	284	0.0	53.5	276	0.5	91.4	461	22
EKF		51.8	284	0.0	55.4	284	0.4	66.0	313	14

Table 2. Results of the simulations

The filtered solutions were consistent when the blunder probability was small, but with large blunder probability ABEKF and EKF produced more inconsistent solutions. REKF, however, does not suffer from inconsistency even with high blunder probabilities. The mean error and the 95% percentile of errors did not vary much between different filters. However, when the blunder probability was large, the best performance was obtained using REKF. ABEKF also performed better than EKF with respect to any criterion with large blunder

probabilities when using the Huber's M-estimator or the p-point M-estimator. Damped Hampel M-estimator did not perform very well with ABEKF, but gave the best results when used in REKF. The fact that EKF performed quite well even with large blunder probabilities was surprising, but may be justified by noting that with almost linear measurements EKF should be optimal regardless of the noise distribution.

7.2 Tests using real GPS data

The filters were also tested using real GPS data. The test bench consists of 40 sets of measurements, which were recorded with a GPS receiver in Tampere, Finland. The receiver used was a Bluetooth Assisted GPS, BAG (Wirola et al., 2006). The sets consisted of cases where the user was standing still, walking or traveling in a bus with the receiver. The true track was only approximately known and it was transformed into digital format using a digital map of Tampere. Thus, the reporter errors are not exact, but instead, should be considered only as indicative.

The results of the tests using real GPS data are presented in Table 3. ABEKF does not seem to work very well with DHA influence function, whereas the influence function of the Huber M-estimator and the influence function of the p-point M-estimator seem to work well. However, ABEKF outperforms EKF with all choices of influence function. REKF outperforms ABEKF with respect to mean error and 95% percentile but ABEKF is more consistent.

The consistency of the solutions for each filter seems to be a lot worse than in the simulations. The inconsistency results from the fact that the variances of the measurements given by the measurement device are too optimistic. This might also be the cause of the poor performance of ABEKF when using the Damped Hampel M-estimator. Nevertheless, ABEKF and REKF perform better than EKF with respect to any criterion. By optimizing the parameters of these filters it might be possible to obtain even better results.

		ME (m)	95% (m)	Inc. (%)
	H	26	55	37.3
REKF	M	27	55	35.4
	DHA	24	55	36.9
	H	29	57	36.3
ABEKF	M	29	56	34.5
	DHA	1046	9308	36.1
EKF		27051	221527	44.7

Table 3. Results of the tests using real GPS data

An example of a real positioning scenario is presented in Figures 2 and 3. In Figure 2 the measurements were processed with EKF using the Damped Hampel weight function, and in Figure 3 the same measurements were processed with REKF. The true track is denoted by a black dashed line, and the mean estimates of the filtered solutions are denoted by blue and red dots. Blue dots represent consistent estimates and red dots are inconsistent estimates. This example shows how poorly EKF might work even in quite good signal conditions, and how significantly REKF improves the estimation process.
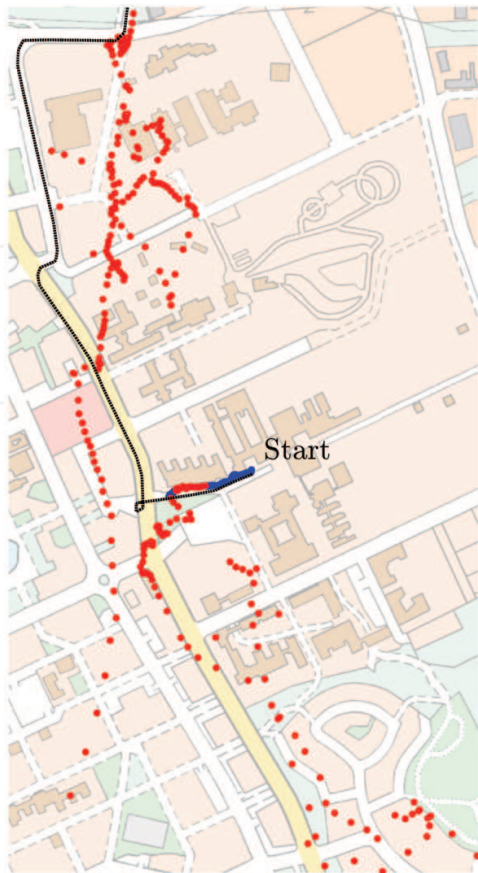
Fig. 2. An example of EKF using real GPS-measurements. (Map © Kaupunkimittaus Tampere 2008)
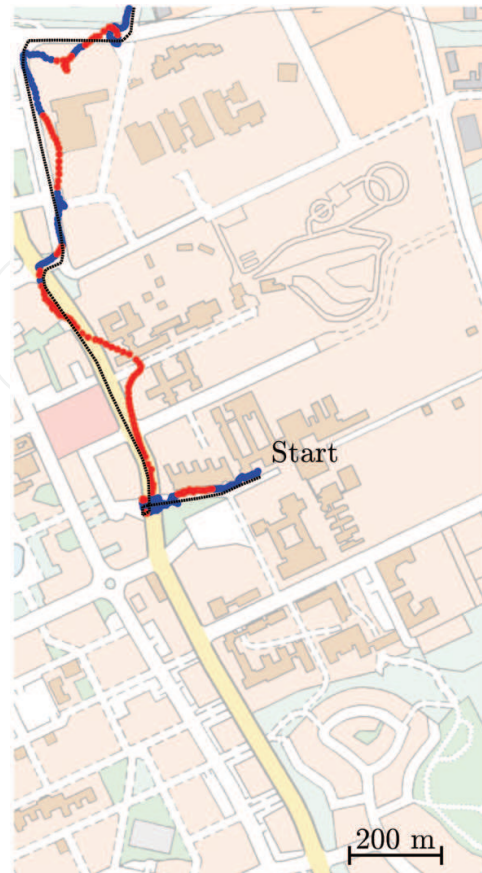
Fig. 3. An example of REKF using real GPS-measurements

8. Conclusions

In this chapter robust filtering techniques for positioning using satellite measurements were presented. The Extended Kalman Filter was chosen as basis for robust filter design. Six filters were presented and tested in the simulations and using real GPS data.

Based on the simulations the proposed filters seem to outperform EKF when blunder measurements occur, and do almost as well in normal cases. However, ABEKF does not seem to work well when using the Damped Hampel M-estimator, but performs better than EKF when using the Huber's M-estimator or the p-point M-estimator. The best performance was obtained using REKF with the Damped Hampel M-estimator, but other M-estimators also seemed to work almost as well.

The tests using real GPS data showed similar trends in the results except that EKF performed very poorly. REKF with the Damped Hampel M-estimator performed best also when using real GPS data.

Therefore, REKF should be taken into consideration when implementing positioning algorithms in mobile positioning devices due to its light computational and memory requirements and relatively high accuracy. The most promising approach was to use the Damped Hampel weight functions, but the performance degrades only a little when using any other influence and weight function proposed in this chapter.

9. References

Ali-Löytty, S.; Sirola, N. & Piché, R. (2005). Consistency of three Kalman Filter extensions inhybrid navigation, *Proceedings of the European Navigation Conference GNSS*, July 2005

Ali-Löytty, S.; Sirola, N. (2007a). Gaussian Mixture Filter in Hybrid Navigation, *Proceedingsof the European Navigation Conference GNSS*, Geneva, Switzerland, May 2007

Ali-Löytty, S.; Sirola, N. (2007b). Gaussian Mixture Filter and Hybrid Positioning,*Proceedings of ION GNSS 2007*, pp. 562−570, Forth Worth, Texas, September 2007

Ali-Löytty, S. (2008). Efficient Gaussian Mixture Filter for Hybrid Positioning, *Proceedings ofIEEE/ION Position Location and Navigation Symposium*, pp. 60−66, Monterey, California, May 2008

Carosio, A.; Cina, A.; Piras, M. (2005). The Robust Statistics Method Applied to the Kalman Filter: Theory and Application, *Proceedings of ION GNSS 2005*, September 2005

Durovic, Z.; Kovacevic, B. (1999). Robust Estimation with Unknown Noise Statistics, *IEEE Transactions on Automatic Control*, Vol. 44, (June 1999), pp. 1292--1296

Huber, P. (1964). Robust Estimation of Location Parameter, *Annals of Mathematical Statistics*, Vol. 35, (1964), pp. 73–101

Ma, C. (2003). Integration of GPS and Cellular Networks to Improve Wireless Location Performance, *Proceedings of ION GPS/GNSS 2003*, pp. 1585−1596, September 2003

Martin, R.; Masreliez, C. (1975). Robust Estimation via Stochastic Approximation, *IEEE Transactions of Information Theory*, Vol. IT-21, (May 1975), pp. 263-271

Masreliez, C.; Martin, R. (1977). Robust Bayesian Estimation for the Linear Model and Robustifying the Kalman Filter, *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 3, (1977), pp. 361−371

Perälä, T.; Piché, R. (2007). Robust Extended Kalman Filtering in Hybrid Positioning Applications, *Proceedings of the 4th Workshop on Positioning, Navigation and Communication (WPNC'07)*, pp. 55−64, Hannover, Germany, March 2007

Raitoharju, M.; Sirola, N.; Ali-Löytty, S.; Piché, R. (2008). PNaFF: A Modular Software Platform for Testing Hybrid Positioning Estimation Algorithms, *Proceedings of the 5th Workshop on Positioning, Navigation and Communication (WPNC'08)*, Hannover, Germany, March 2008

Ristic, B.; Arulampalam, S.; Gordon, N. (2004). *Beyond the Kalman Filter, Particle Filters for Tracking Applications*, Artech House, 1-58053-631-x, Boston, London

Wirola, L.; Alanen, K.; Käppi, J.; Syrjärinne, J. (2006). Bringing RTK to Cellular Terminals Using a Low-cost Single-frequency AGPS Receiver and Inertial Sensors, *Proceedings of 2006 IEEE/ION Position, Location, and Navigation Symposium*, 2006

Kalman Filter

Edited by Vedran Kordic

The Kalman filter has been successfully employed in diverse areas of study over the last 50 years and the chapters in this book review its recent applications. The editors hope the selected works will be useful to readers, contributing to future developments and improvements of this filtering technique. The aim of this book is to provide an overview of recent developments in Kalman filter theory and their applications in engineering and science. The book is divided into 20 chapters corresponding to recent advances in the filed.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tommi Perala (2010). Robust Kalman-type Filtering in Positioning Applications, Kalman Filter, Vedran Kordic (Ed.), ISBN: 978-953-307-094-0, InTech, Available from: http://www.intechopen.com/books/kalman-filter/robust-kalman-type-filtering-in-positioning-applications

INTECH
open science | open minds