Chapter 6

Nonlinear State and Parameter Estimation Using Iterated Sigma Point Kalman Filter: Comparative Studies

Marwa Chaabane, Imen Baklouti, Majdi Mansouri, Nouha Jaoua, Hazem Nounou, Mohamed Nounou, Ahmed Ben Hamida and Marie-France Destain

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/63728

Abstract

In this chapter, iterated sigma-point Kalman filter (ISPKF) methods are used for nonlinear state variable and model parameter estimation. Different conventional state estimation methods, namely the unscented Kalman filter (UKF), the central difference Kalman filter (CDKF), the square-root unscented Kalman filter (SRUKF), the square-root central difference Kalman filter (SRCDKF), the iterated unscented Kalman filter (IUKF), the iterated central difference Kalman filter (ICDKF), the iterated square-root unscented Kalman filter (ISRUKF) and the iterated square-root central difference Kalman filter (ISRCDKF) are evaluated through a simulation example with two comparative studies in terms of state accuracies, estimation errors and convergence. The state variables are estimated in the first comparative study, from noisy measurements with the several estimation methods. Then, in the next comparative study, both of states and parameters are estimated, and are compared by calculating the estimation root mean square error (RMSE) with the noise-free data. The impacts of the practical challenges (measurement noise and number of estimated states/parameters) on the performances of the estimation techniques are investigated. The results of both comparative studies reveal that the ISRCDKF method provides better estimation accuracy than the IUKF, ICDKF and ISRUKF. Also the previous methods provide better accuracy than the UKF, CDKF, SRUKF and SRCDKF techniques. The ISRCDKF method provides accuracy over the other different estimation techniques; by iterating maximum a posteriori estimate around the updated state, it re-linearizes the measurement equation instead of depending on the predicted state. The results also represent that estimating more parameters impacts the estimation accuracy as well as the convergence of the estimated parameters and states. The ISRCDKF provides improved state accuracies than the other techniques even with abrupt changes in estimated states.

Keywords: Kalman filter, sigma point, state estimation, parameter estimation, nonlinear system

1. Introduction

Dynamic state-space models [1–3] are useful for describing data in many different areas, such as engineering [4–8], biological data [9, 10], chemical data [11, 12], and environmental data [8, 13–15]. Estimation of the state and model parameters based on measurements from the observation process is an essential task when analyzing data by state-space models. Bayesian estimation filtering represents a solution of considerable importance for this type of problem definition as demonstrated by many existing algorithms based on the Bayesian filtering [16–25]. The Kalman filter (KF) [26–29] has been extensively utilized in several science applications, such as control, machine learning and neuroscience. The KF provides an optimum solution [28], when the model describing the system is supposed to be Gaussian and linear. However, the KF is limited when the model is considered to be nonlinear and present non-Gaussian modeling assumptions. In order to relax these assumptions, the extended Kalman filter (EKF) [26, 27, 30–32], the unscented Kalman filter (UKF) [33–36], the central difference Kalman filter (CDKF) [37, 38], the square-root unscented Kalman filter (SRUKF) [39, 40], the square-root central difference Kalman filter (SRCDKF) [41], the iterated unscented Kalman filter (IUKF) [42, 43], the iterated central difference Kalman filter (ICDKF) [44, 45], the iterated square-root unscented Kalman filter (ISRUKF) [46] and the iterated square-root central difference Kalman filter (ISRCDKF) [47] have been developed. The EKF [26] linearizes the model describing the system to approximate the covariance matrix of the state vector. However, the EKF is not always performing especially for highly nonlinear or complex models. On behalf of linearizing the model, a class of filters called the sigma-point Kalman filters (SPKFs) [48] uses a statistical linearization technique which linearizes a nonlinear function of a random variable via a linear regression. This regression is done between n points drawn from the prior distribution of the random variable, and the nonlinear functional evaluations of those points. The sigma-point family of filters has been proposed to address the issues of the EKF by making use of a deterministic sampling approach. In this approach, the state distribution is approximated and represented by a set of chosen weighted sample points which capture the true mean and covariance of the state vector. These points are propagated through the true nonlinear system and capture the posterior mean and the covariance matrix of the state vector accurately to the third order (Taylor series expansion) for any nonlinearity. As part of the SPKF family, the UKF [26, 27, 33] has been developed. It uses the unscented transformation, in which a set of samples (sigma points) are propagated and selected by the nonlinear model, providing more accurate approximations of the covariance matrix and mean of the state vector. However, the UKF technique has the limit of the number of sigma-points which are not so large and cannot represent complicated distributions. Another filter in the SPKF family is the central difference Kalman filter (CDKF) [37, 38]. It uses the Stirling polynomial interpolation formula. This filter has the benefit over the UKF in using only one parameter when generating the sigma-point.

http://dx.doi.org/10.5772/63728

To add some benefits of numerical stability, the SRUKF and the SRCDKF [41] have been developed. The advantage of these filters is that they ensured positive semidefiniteness of the state covariances. The iterated sigma-point Kalman filter (ISPKF) methods employ an iterative procedure within a single measurement update step by resampling the sigma-point till a termination criterion, based on the minimization of the maximum likelihood estimate, is satisfied.

The objectives of this chapter are threefold: (i) To estimate nonlinear state variables and model parameters using SPKF methods and extensions through a simulation example. (ii) To investigate the effects of practical challenges (such as measurement noise and number of estimated states/parameters) on the performances of the techniques. To study the effect of measurement noise on the estimation performances, several measurement noise levels will be considered. Then, the estimation performances of the techniques will be evaluated for different noise levels. Also, to study the effect of the number of estimated states/parameters on the estimation performances of all the techniques, the estimation performance will be studied for different numbers of estimated states and parameters. (iii) To apply the techniques to estimate the state variables as well as the model parameters of second-order LTI system. The performances of the estimation techniques will be compared to each other by computing the execution times as well as the estimation root mean square error (RMSE) with respect to the noise-free data.

2. State estimation problem

Next, we present the formulation of the state estimation problem.

2.1. Problem description and formulation

The state estimation problem for a system of nonlinear complex model is described as follows:

$$\begin{aligned} \dot{x} &= g(x, u, \theta, w) \\ y &= l(x, u, \theta, v) \end{aligned} \tag{1}$$

where $x \in \mathrm{R}^{\mathrm{n}}$ is the state variable vector, $y \in \mathrm{R}^{\mathrm{m}}$ is the measurement vector, $\theta \in \mathrm{R}^{\mathrm{q}}$ is the unknown vector, $u \in \mathrm{R}^{\mathrm{p}}$ is the input variable vector, $w \in \mathrm{R}^{\mathrm{n}}$ and $v \in \mathrm{R}^{\mathrm{m}}$ are respectively process and measurement noise vectors, and g and l are nonlinear differentiable functions. The discretization of the model (1) is presented as follows:

$$\begin{aligned} x_k &= \mathrm{f}(x_{k-1}, u_{k-1}, \theta_{k-1}, w_{k-1}) \\ y_k &= h(x_k, u_k, \theta_k, v_k) \end{aligned} \tag{2}$$

which describes the state variables at some time step (k) in terms of their values at a previous time step ($k-1$). Since we are interested to estimate the state vector x_k, as well as the parameter vector θ_k, the parameter vector is assumed to be presented as follows:

$$\theta_k = \theta_{k-1} + \gamma_{k-1} \tag{3}$$

This means that it corresponds to a stationary process, with an identity transition matrix, driven by white noise. In order to include the parameter vector θ_k into the state estimation problem, let us define a new state vector z_k that augments the state vector x_k and the parameter vector θ_k as follows:

$$z_k = \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, u_{k-1}, v_{k-1}, \theta_{k-1}) \\ \theta_{k-1} + \gamma_{k-1} \end{bmatrix} \tag{4}$$

where $z_k \in \mathrm{R}^{\mathrm{n+q}}$. Also, defining the augmented noise vector as:

$$\varepsilon_{k-1} = \begin{bmatrix} v_{k-1} \\ \gamma_{k-1} \end{bmatrix} \tag{5}$$

The model (2) can be written as:

$$z_k = \mathcal{F}(z_{k-1}, u_{k-1}, \varepsilon_{k-1}) \tag{6}$$

$$y_k = \mathcal{R}(z_k, u_k, v_k) \tag{7}$$

where $\mathcal{F}$ and $\mathcal{R}$ are differentiable nonlinear functions. Thus, the objective here is to estimate the augmented state vector z_k, given the measurement vector y_k.

3. Description of state estimation methods

3.1. UKF

The UKF is a SPKF that uses the unscented transformation. This transformation is a method for calculating the statistics of a random variable that undergoes a nonlinear mapping. It is built on the theory that "it is easier to approximate a probability distribution than an arbitrary nonlinear function".

The state distribution is represented by a Gaussian random variable (GRV) and by a set of deterministically chosen points. These points capture the true mean and covariance of the GRV

http://dx.doi.org/10.5772/63728

and also capture the posterior mean and covariance accurately to the second order for any nonlinearity and to the third order for Gaussian inputs. Suppose that GRV $z \in R^L$ characterized by a mean $\bar{z}$ and covariance P_z is used in the model. This variable is transformed by a nonlinear function $y = f(z)$. To reach the statistics of y, a $2L + 1$ sigma vector is defined as follows:

$$\begin{aligned} z_0 &= \bar{z} \\ z_i &= \bar{\bar{z}} + (\sqrt{(L+\lambda)P_z})_i \qquad i = 1,..,L \\ z_i &= \bar{\bar{z}} - (\sqrt{(L+\lambda)P_z})_i \qquad i = L+1,..,2L \end{aligned} \tag{8}$$

where L is the dimension of the state z, $\lambda = e^2(L + \kappa) - L$ is a scaling parameter and $(\sqrt{(L+\lambda)P_z})_i$ denotes the ith column of the matrix square root. The constant $10^{-4} < e < 1$ defines the spread of the sigma-points around $\bar{z}$. The constant κ is a scaling parameter which is usually set to zero or $3 - L$ [30].

Then, these sigma-points are propagated through the nonlinear function,

$$\mathrm{Y_i} = \mathrm{f(Z_i)} \quad \mathrm{i} = 0,..,2\mathrm{L} \tag{9}$$

And the mean and covariance matrix of y can be approximated as weighted sample mean and covariance of the transformed sigma-point of Y_i as follows:

$$\bar{y} = \sum_{i=0}^{2L} W_i^{(m)} Y \quad and \quad P_{\tilde{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} (Y_i - \bar{y})(Y_i - \bar{y}) \tag{10}$$

where the weights are given by

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{\lambda+L} \\ W_0^{(c)} &= \frac{\lambda}{\lambda+L} + (1 - e^2 + \zeta) \\ W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(\lambda+L)} \qquad i = 1,..,2L \end{aligned} \tag{11}$$

The parameter ξ is used to integrate prior knowledge about the distribution of z.

The algorithm of the UKF includes two steps: prediction and update. In the prediction step, we calculate the predicted state estimate $\hat{z}_k^-$ and the predicted estimate covariance P_k^-. In the update step, we calculate the updated state estimate $\hat{z}_k$ and the updated estimate covariance P_k after calculating the innovation residual $P_{z_k y_k}$ and the optimal Kalman gain K_k.

The UKF technique is summarized in Algorithm 1.

3.2. CDKF method

The CDKF is another filter from the family of SPKF. This filter is based on Sterling polynomial interpolation formula instead of the unscented transformation used in UKF. The CDKF is similar to the UKF with the same or superior performance. However, it has an advantage over the UKF that it uses only one parameter instead of three parameters in the UKF. The CDKF uses a symmetric set of $(2L+1)$ sigma-point which are calculated as follows,

$$\begin{aligned} z_0 &= \hat{z} \\ z_i &= \hat{z} + (h\sqrt{P_z})_i \qquad i = 1,..,L \\ z_i &= \hat{z} - (h\sqrt{P_z})_i \quad i = L+1,..,2L \end{aligned} \tag{12}$$

where L is the dimension of the state z, h is a scaling parameter (the optimal value is $h=\sqrt{3}$) and i indicates the ith column of the matrix.

These sigma-points are propagated through the nonlinear function to form the set of the posterior sigma-point,

$$Y_i = f(\Psi_i) \quad i = 0,..,2L \tag{13}$$

Within the above results, the sterling approximation estimates of the mean $\hat{z}$, covariance P_y and cross covariance $P_{z,y}$ are obtained through a linear regression of weighted point,

$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} Y_i \tag{14}$$

$$P_{\bar{y}_k} = \sum_{i=1}^{L} \left[W_i^{(c_1)} (Y_{i,k|k-1} + Y_{i+L,k|k-1})^2 + W_i^{(c_2)} (Y_{i,k|k-1} + Y_{i+L,k|k-1} + 2Y_0)^2 \right] \tag{15}$$

$$P_{z_k y_k} = \sqrt{W_1^{(c_1)} P_z} \left[Y_{1:L,k|k-1} - Y_{L+1:2L,k|k-1} \right]^T \tag{16}$$

The set of corresponding weights for the mean $W_i^{(m)}$ which are used to compute the posterior mean is defined as:

$$W_0^{(m)} = \frac{h^2 - L}{h^2}, W_i^{(m)} = \frac{1}{2h^2} \tag{17}$$

And the set of corresponding weights for the covariance $W_0^{(c)}$ which is used to recover the covariance and the cross-covariance is defined as,

http://dx.doi.org/10.5772/63728

$$W_i^{(c1)} = \frac{1}{4h^2}, \quad W_i^{(c2)} = \frac{h^2-1}{4h^4} \qquad \text{i} = 1,..,2\text{L} \tag{18}$$

The CDKF technique is summarized in Algorithm 2.

3.3. SRUKF method

One drawback of the UKF is that it requires the calculation of the matrix square-root $S_k S_k^T = P_k$, at each time step. That is why a square-root form of the UKF has been developed to reduce the computational complexity. In this new method the covariance matrix S_k will be propagated directly, avoiding to refactorize at each time step [34].

The SRUKF is initialized as follows:

$$\hat{z}_0 = E[z_0] \; And \; S_0 = chol\{E[(z_0 - \hat{z}_0)(z_0 - \hat{z}_0)']\} \tag{19}$$

$$\Psi_{k-1} = [\hat{z}_{k-1} \qquad \hat{z}_{k-1} + hS_{k-1} \qquad \hat{z}_{k-1} - hS_{k-1}] \tag{20}$$

Algorithm 1: UKF algorithm

- *Initialization step:*

$$z_0 = E[z_0] \; and \; P_{z_0} = [(z - \hat{z}_0)(z - \hat{z}_0)^T]$$

- *Prediction step:*

$$\Psi_{k-1} = \left[\hat{z}_{k-1} \quad \hat{z}_{k-1} + \sqrt{(L+\lambda)P_z} \quad \hat{z}_{k-1} - \sqrt{(L+\lambda)P_z}\right]$$

$$\Psi_{k|k-1} = f(\Psi_{k-1})$$

$$\hat{z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \; \Psi_{i,k|k-1}$$

$$P_k^- = \sum_{i=0}^{2L} W_i^{(c)} \; (\Psi_{i,k|k-1} - \hat{z}_k^-)(\Psi_{i,k|k-1} - \hat{z}_k^-)^T$$

$$Y_{k|k-1} = h\left[\Psi_{k|k-1}\right]$$

$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,k|k-1}$$

- *Estimation (update) step:*

$$p_{\tilde{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[Y_{i,k|k-1} - \hat{y}_k^-\right]\left[Y_{i,k|k-1} - \hat{y}_k^-\right]^T$$

$$P_{z_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\Psi_{i,k|k-1} - \hat{z}_k^-\right]\left[Y_{i,k|k-1} - \hat{y}_k^-\right]^T$$

$$K_k = P_{z_k y_k} P_{\tilde{y}_k}^{-1}$$

$$\hat{z}_k = \hat{z}_k^- + K_k(y_k - \hat{y}_k^-)$$

$$P_k = P_k^- - K_k P_{\tilde{y}_k} K_k^T$$

Return the augmented state estimation $\hat{z}_k$

Algorithm 2: CDKF algorithm

- *Initialization step:*

$$z_0 = E\left[z_0\right] and\ P_{z_0} = \left[(z - \hat{z}_0)(z - \hat{z}_0)^T\right]$$

- *Prediction step:*

$$\Psi_{k-1} = \left[\hat{z}_{k-1} \quad \hat{z}_{k-1} + h\sqrt{P_z} \quad \hat{z}_{k-1} - h\sqrt{P_z}\right]$$

$$\Psi_{k|k-1} = f(\Psi_{k-1})$$

$$\hat{z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,k|k-1}$$

http://dx.doi.org/10.5772/63728

$$P_k^- = \sum_{i=0}^{L}\left[W_i^{(c_1)}(\Psi_{i,k|k-1} - \Psi_{L+i,k|k-1})^2 + W_i^{(c_2)}(\Psi_{i,k|k-1} + \Psi_{L+i,k|k-1} - 2\Psi_{0,k|k-1})^2\right]$$

$$Y_{k|k-1} = h\left[\Psi_{k|k-1}\right]$$

$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,k|k-1}$$

- *Estimation (update) step:*

$$P_{\bar{y}_k} = \sum_{i=0}^{L}\left[W_i^{(c_1)}(Y_{i,k|k-1} - Y_{L+i,k|k-1})^2 + W_i^{(c_2)}(Y_{i,k|k-1} + Y_{L+i,k|k-1} - 2Y_0)^2\right]$$

$$P_{z_k y_k} = \sqrt{W_1^{(c_1)} P_k^-}\left[Y_{1:L,k|k-1} - Y_{L+1:2L,k|k-1}\right]^T$$

$$K_k = P_{z_k y_k} P_{\bar{y}_k}^{-1}$$

$$\hat{z}_k = \hat{z}_k^- + K_k(y_k - \hat{y}_k^-)$$

$$P_k = P_k^- - K_k P_{\bar{y}_k} K_k^T$$

Return the augmented state estimation $\hat{z}_k$

The Cholesky factorization decomposes a symmetric, positive-definite matrix into the product of a lower triangular matrix and its transpose. This new matrix is utilized directly to obtain the sigma-point: The scaling constant h is expressed as $h = \sqrt{L\alpha^2}$, where α is a tunable parameter less than one.

In order to predict the current attitude based on each sigma-point, these sigma-points are transformed through the nonlinear process system

$$\Psi_{k/k-1} = f[\Psi_{k-1}] \tag{21}$$

Then, the state mean and the square-root covariance are estimated and calculated through the transformed sigma-point as follows:

$$\hat{z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,k|k-1} \tag{22}$$

$$S_k^- = qr\left\{\left[\sqrt{W_1^{(c)}}\left(\Psi_{1:2L,k/k-1} - \hat{Z}_k^-\right)\sqrt{R^w}\right]\right\} \tag{23}$$

$$S_k^- = cholupdate\left\{S_k^-,\ \Psi_{0,k} - \hat{Z}_k^-,\ W_0^{(c)}\right\} \tag{24}$$

where $W_0^{(c)} = 2\left(1 - \alpha^2 + \frac{1}{2}\beta\right)$, $W_0^{(m)} = 1 - \alpha^2$ and $W_i^{(m)} = W_i^{(c)} = \frac{1}{2L\alpha^2}\beta$, β is a tunable parameter used to include prior distribution. The transformed sigma-point vector is then used to predict the measurements using the measurement model:

$$Y_{k|k-1} = h\left[\Psi_{k|k-1}\right] \tag{25}$$

The expected measurement $\hat{y}_k^-$ and square-root covariance of $\tilde{y}_k = y_k - \hat{y}_k^-$ (called the innovation) are given by the unscented transform expressions just as for the process model:

$$\hat{y}_k^- = \sum_{i=0}^{(m)} W_i^{(m)}\, Y_{i,k|k-1} \tag{26}$$

$$S_{\tilde{y}k} = qr\left\{\left[\sqrt{W_1^{(c)}}\left(Y_{1:2L,k|k-1} - \hat{y}_k\right)\sqrt{R_k^v}\right]\right\} \tag{27}$$

$$S_{\tilde{y}k} = cholupdate\left\{S_{\tilde{y}k},\ Y_{0,k} - \hat{y}_k,\ W_0^{(c)}\right\} \tag{28}$$

In an attempt to find out how much to adjust the predicted state mean and covariance based on the actual measurement, the Kalman gain matrix K_k is calculated as follows:

$$P_{Zkyk} = \sum_{i=0}^{2L} W_i^{(c)}\left[\Psi_{i,k|k-1} - \hat{Z}_k^-\right]\left[Y_{i,k|k-1} - \hat{y}_k^-\right] \tag{29}$$

$$K_K = P_{Zkyk}/S_{\tilde{y}k}^T/S_{\tilde{y}k} \tag{30}$$

Finally, the state mean and covariance are updated using the actual measurement and the Kalman gain matrix:

$$\hat{z}_k = \hat{z}_k^- + K_k(y_k - \hat{y}_k^-) \tag{31}$$

$$U = K_k S_{\tilde{y}k} \tag{32}$$

$$S_k = cholupdate\left\{S_k^-,\ U, -1\right\} \tag{33}$$

http://dx.doi.org/10.5772/63728

where R^w is the process noise covariance, R^v is the measurement noise covariance, chol is Cholesky method of matrix factorization, *qr* is QR matrix decomposition and cholupdate is a Cholesky factor updating.

The SRUKF technique is summarized in Algorithm 3.

3.4. SRCDKF method

Like the SRUKF, the matrix square-root S_k will be propagated directly, avoiding the computational complexity to refactorize at each time step in the CDKF. The SRCDKF is initialized with a state mean vector and the square root of a covariance.

$$\hat{z}_0 = E[z_0] And\ S_0 = chol\{E[(z_0 - \hat{z}_0)(z_0 - \hat{z}_0)']\} \tag{34}$$

After the Cholesky factorization we obtain the sigma-point:

$$\Psi_{k-1} = [\hat{z}_{k-1} \quad \hat{z}_{k-1} + hS_{k-1} \quad \hat{z}_{k-1} - hS_{k-1}] \tag{35}$$

The sigma-point vector is then gone through the nonlinear process system, which predicts the current attitude based on each sigma-point.

$$\Psi_{k/k-1} = f[\Psi_{k-1}] \tag{36}$$

The estimated state mean and square-root covariance are calculated from the transformed sigma-point using,

$$\hat{z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,k|k-1} \tag{37}$$

$$S_k^- = qr\left\{\left[\sqrt{W_1^{(c1)}}\left(\Psi_{1:L,k|k-1} - \Psi_{L+1:2L,k|k-1}\right) \sqrt{W_1^{(c2)}}\left(\Psi_{1:L,k|k-1} + \Psi_{L+1:2L,k|k-1} - 2\Psi_{0,k|k-1}\right)\right]\right\} \tag{38}$$

where $W_i^{(c_1)} = \frac{1}{4h^2}$, $W_i^{(c_2)} = \frac{h^2-1}{4h^4}$, $W_0^{(m)} = \frac{h^2-L}{h^2}$ and $W_i^{(m)} = \frac{1}{2h^2}$. The next step, the sigma-point for measurement update is calculated as,

$$\Psi_{k|k-1} = [\hat{z}_{k|k-1} \quad \hat{z}_{k|k-1} + hS_{k|k-1} \quad \hat{z}_{k|k-1} - hS_{k|k-1}] \tag{39}$$

The transformed sigma-point vector is then used to predict the measurements using the measurement model:

$$Y_{k|k-1} = h\left(\Psi_{k|k-1}\right) \tag{40}$$

The expected measurement $\hat{y}_k^-$ and square-root covariance of $\tilde{y}_k = y_k - \hat{y}_k^-$ (called the innovation) are given by expressions just as for the process model:

$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{k|k-1} \tag{41}$$

$$S_{\tilde{y}_k} = qr\left\{\left[\sqrt{W_1^{(c_1)}}\left(Y_{1:L:2L,k|k-1} - Y_{L+1:2L,k|k-1}\right)\sqrt{W_1^{(c_2)}}\left(Y_{1:L,k|k-1} - Y_{L+1:2L,k|k-1} - 2Y_{0,k|k-1}\right)\right]\right\} \tag{42}$$

In an attempt to find out how much to adjust the predicted state mean and covariance based on the actual measurement, the Kalman gain matrix K_k is calculated as follows:

$$P_{z_k y_k} = W_1^{(c_1)} S_k^- \left[Y_{1:L,k|k-1} - Y_{L+1:2L,k|k-1}\right]^T \tag{43}$$

$$K_k = P_{z_k y_k} / S_{\tilde{y}_k}^T / S_{\tilde{y}_k} \tag{44}$$

Then, the state mean and covariance are updated using the actual measurement and the Kalman gain matrix is:

$$\hat{z}_k = \hat{z}_k^- + K_k(y_k - \hat{y}_k^-) \tag{45}$$

$$U = K_k S_{\tilde{y}_k} \tag{46}$$

$$S_k = cholupdate\{S_k^-, U, -1\} \tag{47}$$

The SRCDKF technique is summarized in Algorithm 4.

3.5. ISPKF

In order to achieve superior performance of the statical linearization methods in terms of efficiency and accuracy, the ISPKFs have been developed. These filters include IUKF, ICDKF, ISRUKF and ISRCDKF. The major difference between the ISPKFs and the noniterated SPKFs is shown in the step where the updated state estimation is calculated using the predicted state and the observation. Instead of relying on the predicted state, the observation equation is relinearized over times by iterating an approximate maximum a posteriori estimate, so the state estimate will be more accurate.

http://dx.doi.org/10.5772/63728

3.5.1. IUKF

The difference between the UKF and the IUKF consists in the iteration strategy.

After generating the prediction and the update steps, and getting both the state estimate $\hat{z}_k$ and the covariance matrix P_k, an iteration loop is set up with the following initializations:

$\hat{z}_{k,0} = \hat{z}_k^-$, $P_{k,0} = P_k^-$, $\hat{z}_{k,1} = \hat{z}_k$, $P_{k,1} = P_k$ and $j = 2$ with j is the jth iterate.

In this loop, for each j, new sigma-points are generated in the same way as the standard UKF

$$\Psi_{k,j} = \left[\hat{z}_{k,j-1} \quad \hat{z}_{k,j-1} + \sqrt{(L+\lambda)P_{k,j-1}} \quad \hat{z}_{k,j-1} - \sqrt{(L+\lambda)P_{k,j-1}}\right] \tag{48}$$

Algorithm 3: SRUKF algorithm

- *Initialization step:* $\hat{z}_0 = E\left[z_0\right]$

$$And\ S_0 = chol\left\{E\left[\left(z_0 - \hat{z}_0\right)\left(z_0 - \hat{z}_0\right)'\right]\right\}$$

- *Prediction step:*

$$\Psi_{k-1} = \left[\hat{z}_{k-1} \qquad \hat{z}_{k-1} + hS_{k-1} \qquad \hat{z}_{k-1} - hS_{k-1}\right]$$

$$\Psi_{k/k-1} = f\left[\Psi_{k-1}\right]$$

$$\hat{Z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,k|k-1}$$

$$S_k^- = qr\left\{\left[\sqrt{W_1^{(c)}}\left(\Psi_{1:2L,k/k-1} - \hat{Z}_k^-\right)\sqrt{R^w}\right]\right\}$$

$$S_k^- = cholupdate\left\{S_k^-, \Psi_{0,k} - \hat{Z}_k^-, W_0^{(c)}\right\}$$

$$Y_{k|k-1} = h\left[\Psi_{k|k-1}\right]$$

$$\hat{y}_k^- = \sum_{i=0}^{(m)} W_i^{(m)} Y_{i,k|k-1}$$

- *Estimation (update) step:*

$$S_{\tilde{y}_k} = qr\left\{\left[\sqrt{W_1^{(c)}}\left(Y_{1:2L,k|k-1} - \hat{y}_k\right)\sqrt{R_k^v}\right]\right\}$$

$$S_{\tilde{y}_k} = cholupdate\left\{S_{\tilde{y}_k}, Y_{0,k} - \hat{y}_k, W_0^{(c)}\right\}$$

$$P_{Zky k} = \sum_{i=0}^{2L} W_i^{(c)} [\Psi_{i,k|k-1} - \hat{Z}_k^-][Y_{i,k|k-1} - \hat{y}_k^-]$$

$$K_K = P_{Zky k} / S_{\tilde{y}k}^T / S_{\tilde{y}k}$$

$$\hat{z}_k = \hat{z}_k^- + K_k\left(y_k - \hat{y}_k^-\right)$$

$$U = K_k S_{\tilde{y}k}$$

$$S_k = cholupdate\left\{S_k^-, U, -1\right\}$$

Return the augmented state estimation $\hat{z}_k$

Algorithm 4: SRCDKF algorithm

- *Initialization step:* $\hat{z}_0 = E[z_0]$

$$And\ S_0 = chol\left\{E\left[\left(z_0 - \hat{z}_0\right)\left(z_0 - \hat{z}_0\right)'\right]\right\}$$

- *Prediction step:*

$$\Psi_{k-1} = \left[\hat{z}_{k-1} \quad \hat{z}_{k-1} + hS_{k-1} \quad \hat{z}_{k-1} - hS_{k-1}\right]$$

$$\Psi_{k/k-1} = f\left[\Psi_{k-1}\right]$$

http://dx.doi.org/10.5772/63728

$$z_k^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,k|k-1}$$

$$S_k^- = qr\left\{\left[\sqrt{W_1^{(c1)}}\left(\Psi_{1:L,k|k-1} - \Psi_{L+1:2L,k|k-1}\right)\sqrt{W_1^{(c_2)}}\left(\Psi_{1:L,k|k-1} + \Psi_{L+1:2L,k|k-1} - 2\Psi_{0,k|k-1}\right)\right]\right\}$$

$$\Psi_{k|k-1} = \left[\hat{z}_{k|k-1} \quad \hat{z}_{k|k-1} + hS_{k|k-1} \quad \hat{z}_{k|k-1} - hS_{k|k-1}\right]$$

$$Y_{k|k-1} = h(\Psi_{k|k-1})$$

$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{k|k-1}$$

- *Estimation(update) step:*

$$S_{\bar{y}_k} = qr\left\{\left[\sqrt{W_1^{(c_1)}}\left(Y_{1:L,k|k-1} - Y_{L+1:2L,k|k-1}\right)\sqrt{W_1^{(c_2)}}\left(Y_{1:L,k|k-1} - Y_{L+1:2L,k|k-1} - 2Y_{0,k|k-1}\right)\right]\right\}$$

$$P_{z_k y_k} = \sqrt{W_1^{(c_1)}} S_k^-\left[Y_{1:L,k|k-1} - Y_{L+1:2L,k|k-1}\right]^T$$

$$K_k = P_{z_k y_k} / S_{\bar{y}_k}^T / S_{\bar{y}_k}$$

$$\hat{z}_k = \hat{z}_k^- + K_k(y_k - \hat{y}_k^-)$$

$$U = K_k S_{\bar{y}_k}$$

$$S_k = cholupdate\left\{S_k^-, U, -1\right\}$$

Return the augmented state estimation $\hat{z}_k$

Then the prediction step and the update step are executed as follows:

$$\hat{z}_{k,j}^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,j} \tag{49}$$

$$Y_{k,j} = h\left[\Psi_{i,j}\right] \tag{50}$$

where $Y_{k,j}$ represents the ith component of Y_j

$$\hat{y}_{k,j}^{-} = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,j} \tag{51}$$

$$P_{\tilde{y}_{k,j}} = \sum_{i=0}^{2L} W_i^{(c)} \left[Y_{i,j} - \hat{y}_{k,j}^{-}\right] \left[Y_{i,j} - \hat{y}_{k,j}^{-}\right]^T \tag{52}$$

$$P_{z_{k,j} y_{k,j}} = \sum_{i=0}^{2L} W_i^{(c)} \left[\Psi_{i,,j} - \hat{z}_{k,j}^{-}\right] \left[Y_{i,j} - \hat{y}_{k,j}^{-}\right]^T \tag{53}$$

$$K_{k,j} = P_{z_{k,j} y_{k,j}} P_{\tilde{y}_k, j}^{-1} \tag{54}$$

$$\hat{z}_{k,j} = \hat{z}_{k,j}^{-} + g.K_{k,j}(y_k - \hat{y}_{k,j}^{-}) \tag{55}$$

$$P_{k,j} = P_{k,j}^{-} - K_{k,j} P_{\tilde{y}_{k,j}} K_{k,j}^T \tag{56}$$

Those steps are repeated many times until a following inequality is not satisfied.

$$\hat{z}_{k,j}^T P_{k,j-1}^T + \tilde{y}_{k,j}^T R_k^{-1} \tilde{y}_{k,j} < \tilde{y}_{k,j-1}^T R_k^{-1} \tilde{y}_{k,j-1} \; or \; j < N \tag{57}$$

The IUKF is summarized in Algorithm 5.

3.5.2. ICDKF

The iterated sigma-point methods have the ability to provide accuracy over other estimation methods since it relinearizes the measurement equation by iterating an approximate maximum a posteriori estimate around the updated state, instead of relying on the predicted state.

In the ICDKF, the prediction step is calculated as the standard CDKF and we get $\hat{z}_k^{-}$ and P_k^{-}.

Then the sigma-point in measurement updating is calculated as follows:

$$\Psi_{(k|k-1)} = \left[\hat{z}_{(k|k-1\ )} \quad \hat{z}_{(k|k-1)} + h\sqrt{P_k^{-}} \quad \hat{z}_{(k|k-1\ )} - h\sqrt{P_k^{-}}\ \right] \tag{58}$$

After that, the initialization $z_0 = \hat{z}_k^{-}$ is set up and then the iteration step is executed, so the following equations are repeated m times.

http://dx.doi.org/10.5772/63728

$$\Psi_j = \begin{bmatrix} z_j & z_j + h\sqrt{P_k^-} & z_j - h\sqrt{P_k^-} \end{bmatrix} \tag{59}$$

$$Y_j = h[\Psi_j] \tag{60}$$

$$\hat{y}_{k,j}^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,j} \tag{61}$$

$$P_{\bar{y}_k} = \sum_{i=0}^{2L} W_i^{(c_1)} \left(Y_{i,j} - Y_{L+i,j}\right)^2 + W_i^{(c_2)} \left(Y_{i,j} + Y_{L+i,j} - 2Y_{0,j}\right)^2 \tag{62}$$

$$P_{z_k y_k} = \sqrt{W_1^{(c_1)} P_z} \left[Y_{1:L,j} - Y_{L+1:2L,j}\right]^T \tag{63}$$

$$K_{k,j} = P_{z_k y_k} P_{\bar{y}_k}^{-1} \tag{64}$$

$$\hat{z}_{k,j} = \hat{z}_k^- + K_{k,j}(y_k - \hat{y}_{k,j}^-) \tag{65}$$

$$P_k = P_k^- - K_{k,j} P_{\bar{y}_k} K_{k,j}^T \tag{66}$$

The algorithm of the ICDKF is summarized in Algorithm 6.

3.5.3. ISRUKF

The ISRUKF has the same principle as the IUKF. After executing the standard SRUKF, an iteration loop is started. The predicted estimated state $\left(z_k^-, \hat{z}_k\right)$ and the predicted and estimated covariance matrix (S_k^-, S_k) obtained through the prediction and the update steps will be the initialization inputs for the iteration loop ($\hat{z}_{k,0} = \hat{z}_k^-$, $S_{k,0} = S_k^-$ and $\hat{z}_{k,1} = \hat{z}_k$, $S_{k,1} = S_k$). Also let *j*=2 where *j* is the *j*th iteration.

In the iteration loop, and for each *j*, the new sigma-point vector is generated as follows:

$$\Psi_{k,j} = [\hat{z}_{k,j-1} \quad \hat{z}_{k,j-1} + hS_{k,j-1} \quad \hat{z}_{k,j-1} - hS_{k,j-1}] \tag{67}$$

Then, the prediction and the update steps are executed as follows:

$$\Psi_j = f(\Psi_{k,j}) \tag{68}$$

$$\hat{z}_{k,j}^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,j} \tag{69}$$

$$S_k^- = qr\left\{\left[\sqrt{W_1^{(c)}}\left(\Psi_{1:2L,j} - \hat{Z}_{k,j}^-\right)\sqrt{R^w}\right]\right\} \tag{70}$$

$$S_k^- = cholupdate\left\{S_{k,j}^-,\ \Psi_{i,j} - \hat{Z}_{k,j}^-,\ W_1^{(c)}\right\} \tag{71}$$

$$Y_{i,j} = h\left[\Psi_{i,j}\right] \tag{72}$$

$$\hat{y}_{k,j}^- = \sum_{i=0}^{(m)} W_i^{(m)}\, Y_{i,j} \tag{73}$$

$$S_{\tilde{y}k,j} = qr\left\{\left[\sqrt{W_1^{(c)}}\left(Y_{1:2L,j} - \hat{y}_{k,j}\right)\sqrt{R_k^v}\right]\right\} \tag{74}$$

$$S_{\tilde{y}k,j} = cholupdate\left\{S_{\tilde{y}k,j}, Y_{0,j} - \hat{y}_{k,j},\ W_0^{(c)}\right\} \tag{75}$$

$$P_{Zkyk} = \sum_{i=0}^{2L} W_i^{(c)}\left[\Psi_{i,j} - \hat{Z}_{k,j}^-\right]\left[Y_{i,j} - \hat{y}_{k,j}^-\right] \tag{76}$$

$$K_{K,j} = P_{Zkyk}/S_{\tilde{y}k,j}^T/S_{\tilde{y}k,j} \tag{77}$$

$$\hat{z}_{k,j} = \hat{z}_{k-j}^- + K_{k,j}\left(y_k - \hat{y}_{k,j}^-\right) \tag{78}$$

$$U = K_{k,j}S_{\tilde{y}k,j} \tag{79}$$

$$S_{k,j} = cholupdate\left\{S_{k,j-1}^-, U, -1\right\} \tag{80}$$

The equations in the iterative loop are repeated m times.

The ISRUKF algorithm is summarized in Algorithm 7.

3.5.4. ISRCDKF

The ISRCDKF has the ability to provide accuracy over other SRCDKF since it relinearizes the measurement equation by iterating an approximate maximum a posteriori estimate around the updated state, instead of relying on the predicted state.

The algorithm of the ISRCDKF consists of generating the prediction step as the standard SRCDKF, then applying *m* iterations over the update step described as follows:

$$\Psi_j = \left[z_j \quad z_j + hS_k^- \quad z_j - hS_k^-\right] \tag{81}$$

$$Y_j = h\left(\Psi_j\right) \tag{82}$$

$$\hat{y}_{k,j}^{-} = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,j} \tag{83}$$

$$S_{\tilde{y}_k} = qr\left\{\left[\sqrt{W_1^{(c_1)}}\left(Y_{1:L,j} - Y_{L+1:2L,j}\right)\sqrt{W_1^{(c_2)}}\left(Y_{1:L,j} - Y_{L+1:2L,j} - 2Y_{0,j}\right)\right]\right\} \tag{84}$$

$$P_{z_k y_k} = W_1^{(c_1)} S_k^{-}\left[Y_{1:L,j} - Y_{L+1:2L,j}\right]^T \tag{85}$$

$$K_{k,j} = P_{z_k y_k} / S_{\tilde{y}_k}^t / S_{\tilde{y}_k} \tag{86}$$

$$\hat{z}_{k,j} = \hat{z}_k^{-} + K_{k,j}\left(y_k - \hat{y}_{k,j}^{-}\right) \tag{87}$$

$$U = K_{k,j} S_{\tilde{y}_k} \tag{88}$$

$$S_k = cholupdate\{S_k^{-}, U, -1\} \tag{89}$$

The ISRCDKF algorithm is summarized in Algorithm 8.

In the next section, the SPKF method performances will be assessed and compared to ISPKF methods. The performances of UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF methods will be evaluated through a simulation example with two comparative studies in terms of estimation accuracy, convergence and execution times.

4. Simulation results

4.1. State and parameter estimations for a second-order LTI system

Consider a second-order LTI described by the following state variable,

$$\dot{x}_k = Ax_k + BU_1(k) + v_k \tag{90}$$

where v_k is a Gaussian process noise $(v_k; 0, 10^{-1})$, and $A = \begin{pmatrix} 1.9223 & -0.9604 \\ 1 & 0 \end{pmatrix}$ is a matrix with scalar parameter $B = \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}$.

Algorithm 5: IUKF algorithm

- *Initialization step:*

$$Z_0 = E\left[z_0\right]$$

$$P_{z_0} = \left[(z - \hat{z}_0)(z - \hat{z}_0)^T \right]$$

- *Prediction step:*

Generate the UKF prediction step and return $\hat{z}_k^-$ *and* P_k^-

- *Estimation (update) step:*
- *Generate the UKF update step and return* P_k *and* $\hat{z}_k$
- *Iteration: Let* $\hat{z}_{k,0} = \hat{z}_k^-, \quad P_{k,0} = P_k^-$

And $\hat{z}_{k,1} = \hat{z}_k, \quad P_{k,1} = P_k$

Also let $g = 1$ and $j = 2$.

$$\Psi_{k,j} = \left[\hat{z}_{k,j-1} \quad \hat{z}_{k,j-1} + \sqrt{(L+\lambda)P_{k,j-1}} \quad \hat{z}_{k,j-1} - \sqrt{(L+\lambda)P_{k,j-1}} \right]$$

$$\hat{z}_{k,j}^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,j}$$

$$Y_{k,j} = h\left[\Psi_{i,j} \right]$$

$$\hat{y}_{k,j}^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,j}$$

$$P_{\tilde{y}_{k,j}} = \sum_{i=0}^{2L} W_i^{(c)} \left[Y_{i,j} - \hat{y}_{k,j}^- \right] \left[Y_{i,j} - \hat{y}_{k,j}^- \right]^T$$

$$P_{z_{k,j} y_{k,j}} = \sum_{i=0}^{2L} W_i^{(c)} \left[\Psi_{i,j} - \hat{z}_{k,j}^- \right] \left[Y_{i,j} - \hat{y}_{k,j}^- \right]^T$$

$$K_{k,j} = P_{z_{k,j} y_{k,j}} P_{\tilde{y}_{k,j}}^{-1}$$

$$\hat{z}_{k,j} = \hat{z}_{k,j}^- + g.K_{k,j}(y_k - \hat{y}_{k,j}^-)$$

$$P_{k,j} = P_{k,j}^- - K_{k,j} P_{\tilde{y}_{k,j}} K_{k,j}^T$$

http://dx.doi.org/10.5772/63728

Define these equations:

$$\hat{y}_{k,j} = h(\hat{z}_{k,j})$$

$$\tilde{z}_{k,j} = \hat{z}_{k,j} - \hat{z}_{k,j-1}$$

$$\tilde{y}_{k,j} = y_k - \hat{y}_{k,j}$$

If the inequality is fulfilled

$$\hat{z}_{k,j}^T P_{k,j-1}^T + \tilde{y}_{k,j}^T R_k^{-1} \tilde{y}_{k,j} < \tilde{y}_{k,j-1}^T R_k^{-1} \tilde{y}_{k,j-1}$$

And $j < N$ *then set* $g = \eta \cdot g, \quad j = j + 1$, *and return to the iterated loop.*

Otherwise set

$\hat{z}_k = \hat{z}_{k,j}$ *And* $P_k = P_{k,j}$ *Return the augmented state estimation* $\hat{z}_k$

Algorithm 6: ICDKF algorithm

- *Initialization step:*

$$Z_0 = E[z_0]$$

$$P_{z_0} = \left[(z - \hat{z}_0)(z - \hat{z}_0)^T\right]$$

- *Prediction step:*

$$\Psi_{k-1} = \left[\hat{z}_{k-1} \quad \hat{z}_{k-1} + h\sqrt{P_z} \quad \hat{z}_{k-1} - h\sqrt{P_z}\right]$$

$$\Psi_{k|k-1} = f(\Psi_{k-1})$$

$$\hat{z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,k|k-1}$$

$$P_k^- = \sum_{i=0}^{2L} W_i^{(c_1)} (\Psi_{i,k|k-1} - \Psi_{L+i,k|k-1})^2 + W_i^{(c_2)} (\Psi_{i,k|k-1} + \Psi_{L+i,k|k-1} - 2\Psi_{0,k|k-1})^2$$

$$\Psi_{(k|k-1)} = \left[\hat{z}_{(k|k-1)} \; \hat{z}_{(k|k-1)} + h\sqrt{P_k^-} \; \hat{z}_{(k|k-1)} - h\sqrt{P_k^-} \right]$$

- *Estimation(update) step:*

$$z_0 = \hat{z}_k^-$$

- *Iteration: for j=0,.., m*

$$\Psi_j = \left[z_j \quad z_j + h\sqrt{P_k^-} \quad z_j - h\sqrt{P_k^-} \right]$$

$$Y_j = h\left[\Psi_j \right]$$

$$\hat{y}_{k,j}^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,j}$$

$$P_{\bar{y}_k} = \sum_{i=0}^{2L} W_i^{(c_1)} \left(Y_{i,j} - Y_{L+i,j} \right)^2 + W_i^{(c_2)} \left(Y_{i,j} + Y_{L+i,j} - 2Y_{0,j} \right)^2$$

$$P_{z_k y_k} = \sqrt{W_1^{(c_1)}} P_z \left[Y_{1:L,j} - Y_{L+1:2L,j} \right]^T$$

$$K_{k,j} = P_{z_k y_k} P_{\bar{y}_k}^{-1}$$

End

$$\hat{z}_{k,j} = \hat{z}_k^- + K_{k,j} (y_k - \hat{y}_{k,j}^-)$$

$$P_k = P_k^- - K_{k,j} P_{\bar{y}_k} K_{k,j}^T$$

http://dx.doi.org/10.5772/63728

Return the augmented state estimation $\widehat{z}_k$

Algorithm 7: ISRUKF algorithm

- *Initialization step:* $z_0 = E[z_0]$

$$S_0 = chol\left\{E\left[(z_0 - \hat{z}_0)(z_0 - \hat{z}_0)'\right]\right\}$$

- *Prediction step:*
- *Generate the SRUKF prediction step and return* $\hat{z}_k^-$ *and* S_k^-
- *Estimation (update) step:*

Generate the SRUKF update step and return S_k *and* $\hat{z}_k$

- *Iteration: Let* $\hat{z}_{k,0} = \hat{z}_k^-$, $S_{k,0} = S_k^-$ and $\hat{z}_{k,1} = \hat{z}_k$, $S_{k,1} = S_k$. Also let *j*=2

$$\Psi_{k,j} = \left[\hat{z}_{k,j-1} \quad \hat{z}_{k,j-1} + hS_{k,j-1} \quad \hat{z}_{k,j-1} - hS_{k,j-1}\right]$$

$$\Psi_j = f(\Psi_{k,j})$$

$$\hat{z}_{k,j}^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,j}$$

$$S_k^- = qr\left\{\left[\sqrt{W_1^{(c)}}\left(\Psi_{1:2L,j} - \hat{Z}_{k,j}^-\right)\sqrt{R^w}\right]\right\}$$

$$S_k^- = cholupdate\left\{S_{k,j}^-, \Psi_{i,j} - \hat{Z}_{k,j}^-, W_1^{(c)}\right\}$$

$$Y_{i,j} = h\left[\Psi_{i,j}\right]$$

$$\hat{y}_{k,j}^- = \sum_{i=0}^{(m)} W_i^{(m)} Y_{i,j}$$

$$S_{\tilde{y}k,j} = qr\left\{\left[\sqrt{W_1^{(c)}}\left(Y_{1:2L,j} - \hat{y}_{k,j}\right)\sqrt{R_k^v}\right]\right\}$$

$$S_{\tilde{y}k,j} = cholupdate\left\{S_{\tilde{y}k,j}, Y_{0,j} - \hat{y}_{k,j}, W_0^{(c)}\right\}$$

$$P_{Zkyk} = \sum_{i=0}^{2L} W_i^{(c)} \left[\Psi_{i,j} - \hat{Z}_{k,j}^{-}\right]\left[Y_{i,j} - \hat{y}_{k,j}^{-}\right]$$

$$K_{K,j} = P_{Zkyk} / S_{\tilde{y}k,j}^{T} / S_{\tilde{y}k,j}$$

$$\hat{z}_{k,j} = \hat{z}_{k-j}^{-} + K_{k,j}\left(y_k - \hat{y}_{k,j}^{-}\right)$$

$$U = K_{k,j} S_{\tilde{y}k,j}$$

$$S_{k,j} = cholupdate\left\{S_{k,j-1}^{-}, U, -1\right\}$$

End

$$U = K_{k,m} S_{\tilde{y}_k}$$

$$S_k = cholupdate\left\{S_{k,m}, U, -1\right\}$$

Return the augmented state estimation $\hat{z}_k$

Algorithm 8: ISRCDKF algorithm

- *Initialization step:* $\hat{z}_0 = E\left[z_0\right]$

$$S_0 = chol\left\{E\left[\left(z_0 - \hat{z}_0\right)\left(z_0 - \hat{z}_0\right)'\right]\right\}$$

- *Prediction step:*

$$\Psi_{k-1} = \left[\hat{z}_{k-1} \quad \hat{z}_{k-1} + hS_{k-1} \quad \hat{z}_{k-1} - hS_{k-1}\right]$$

$$\Psi_{k/k-1} = f\left[\Psi_{k-1}\right]$$

http://dx.doi.org/10.5772/63728

$$z_k^- = \sum_{i=0}^{2L} W_i^{(m)} \Psi_{i,k|k-1}$$

$$S_k^- = qr\left\{\left[\sqrt{W_1^{(c_1)}}\left(\Psi_{1:L,k|k-1} - \Psi_{L+1:2L,k|k-1}\right)\sqrt{W_1^{(c_2)}}\left(\Psi_{1:L,k|k-1} + \Psi_{L+1:2L,k|k-1} - 2\Psi_{0,k|k-1}\right)\right]\right\}$$

$$\Psi_{(k|k-1)} = \left[\hat{z}_{(k|k-1)} \quad \hat{z}_{(k|k-1)} + hS_{(k|k-1)} \quad \hat{z}_{(k|k-1)} - hS_{(k|k-1)}\right]$$

- *Estimation(update) step:*

$$z_0 = \hat{z}_k^-$$

- *Iteration: for j=0: m*

$$\Psi_j = \left[z_j \quad z_j + hS_k^- \quad z_j - hS_k^-\right]$$

$$Y_j = h\left(\Psi_j\right)$$

$$\hat{y}_{k,j}^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,j}$$

$$S_{\tilde{y}_k} = qr\left\{\left[\sqrt{W_1^{(c_1)}}\left(Y_{1:L,j} - Y_{L+1:2L,j}\right)\sqrt{W_1^{(c_2)}}\left(Y_{1:L,j} - Y_{L+1:2L,j} - 2Y_{0,j}\right)\right]\right\}$$

$$P_{z_k y_k} = W_1^{(c_1)} S_k^-\left[Y_{1:L,j} - Y_{L+1:2L,j}\right]^T$$

$$K_{k,j} = P_{z_k y_k} / S_{\tilde{y}_k}^t / S_{\tilde{y}_k}$$

$$\hat{z}_{k,j} = \hat{z}_k^- + K_{k,j}(y_k - \hat{y}_{k,j}^-)$$

$$U = K_{k,j} S_{\tilde{y}_k}$$

$$S_k = cholupdate\left\{S_k^-, U, -1\right\}$$

End $U = K_{k,m} S_{\tilde{y}_k}$

$$S_k = cholupdate\left\{S_{k,m}, U, -1\right\}$$

Return the augmented state estimation $\hat{z}_k$

The nonstationary observation model is given by,

$$y_k = Cx_k + DU_2(k) + n_k \tag{91}$$

where $C = [1\ \ 0]$ and $D = \begin{bmatrix} 0 \\ 0.2 \end{bmatrix}$. The observation noise n_k is a Gaussian noise $\mathcal{N}(n_k; 0, 3.10^{-1})$. Given only the noisy observations $y_{k'}$ the different filters were used to estimate the underlying clean state sequence x_k for $k = 1...100$.

4.1.1. Generation of dynamic data

It must be noted that this simulated state is assumed to be noise-free. They are contaminated with Gaussian noise. Given noisy observations $y_{k'}$ the various KFs were used to estimate the clean state sequence $x_k = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ for k = 1...100. **Figure 1** shows the changes in the state variable x_1.

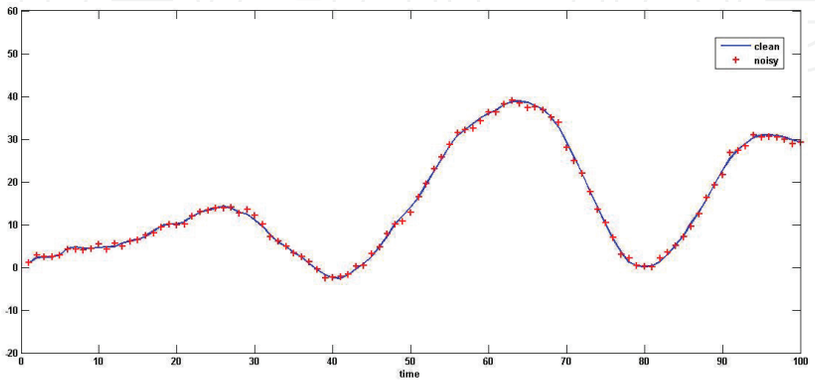

Figure 1. Simulated data used in estimation: state variable (x_1).

http://dx.doi.org/10.5772/63728

Here, the number of sigma-points is fixed to 9 for all the techniques ($L = 4$). The process noise $(v_k; 0, 10^{-1})$ was added. The observation noise is $\mathcal{N}(n_k; 0, 3.10^{-1})$. The initial value of the state vector is $x_0 = [1 \ \ 0]'$.

4.1.2. Comparative study: estimation of state variables from noisy measurements

The purpose of this study is to compare the estimation accuracy of UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF methods when they are utilized to estimate the state variable of the system. Hence, it is considered that the state vector to be estimated $z_k = x_k$ and the model parameters P_1, P_2 are assumed to be known. The simulation results for state estimations of state variable x_k using UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF methods are shown in **Figures 2** and **3**, respectively. Also, the performance comparison of the state estimation techniques in terms of RMSE and execution times is presented in **Table 1**.

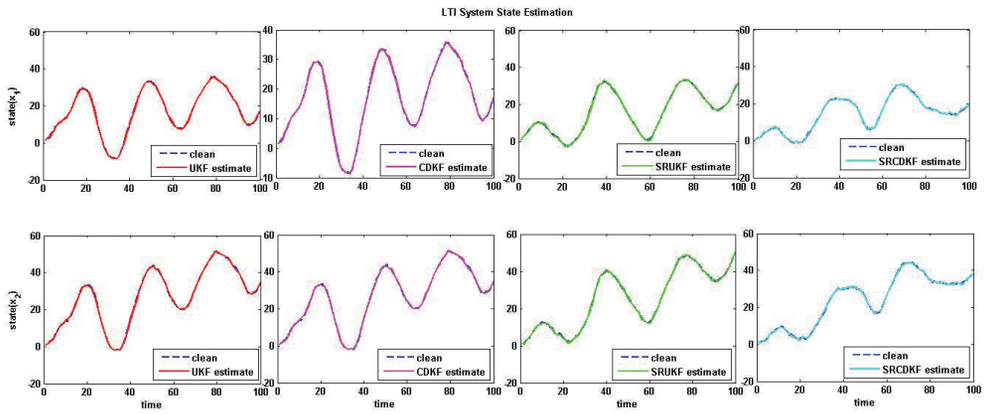

Figure 2. Estimation of state variables using various state estimation techniques (UKF, CDKF, SRUKF and SRCDKF).

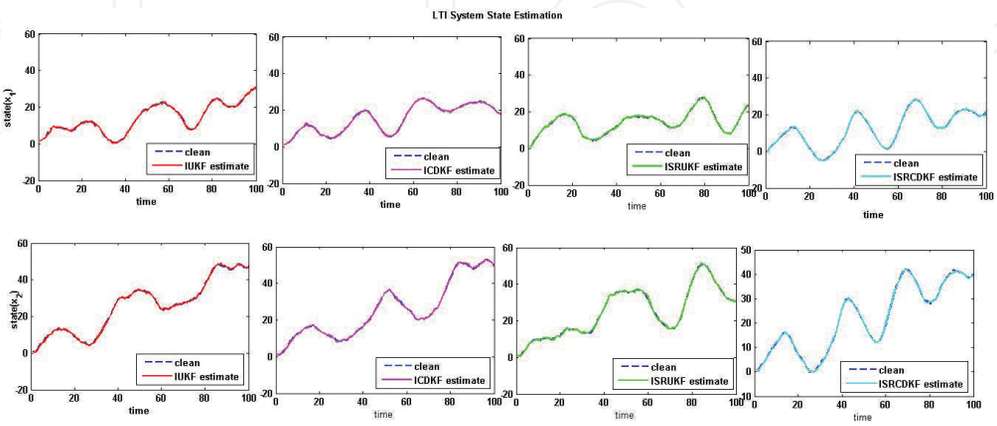

Figure 3. Estimation of state variables using various state estimation techniques (IUKF, ICDKF, ISRUKF and ISRCDKF).

Technique	x_1 (RMSE)	x_2 (RMSE)	Time execution(s)	Technique	x_1 (RMSE)	x_2 (RMSE)	Time execution(s)
UKF	0.3539	0.4658	0.3577	IUKF	0.3342	0.4341	0.5952
CDKF	0.3512	0.4583	0.3367	ICDKF	0.3265	0.4315	0.4351
SRUKF	0.3495	0.4590	0.3354	ISRUKF	0.3254	0.4256	0.5803
SRCDKF	0.3324	0.4593	0.2586	ISRCDKF	0.3121	0.4213	0.4229

Table 1. Comparison of state estimation techniques.

It is easily observed from **Figures 2** and **3** as well as **Table 1** that the ISRCDKF method achieves a better accuracy than the other methods.

4.1.3. Comparative study: simultaneous estimation of state variables and model parameters

The state variables and parameters are estimated and performed using the state estimation techniques UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF. The results of estimation for the model parameters using the estimation techniques (UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF) are shown in **Figures 4** and **5**, respectively. It can be seen from the results presented in **Figures 4** and **5** that the IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF methods outperform the UKF method, and that the ISRCDKF shows relative improvement over all other techniques. These results confirm the results obtained in the first comparative study, where only the state variable is estimated. The advantages of the ISRCDKF over the other techniques can also be seen through their abilities to estimate the model parameters.

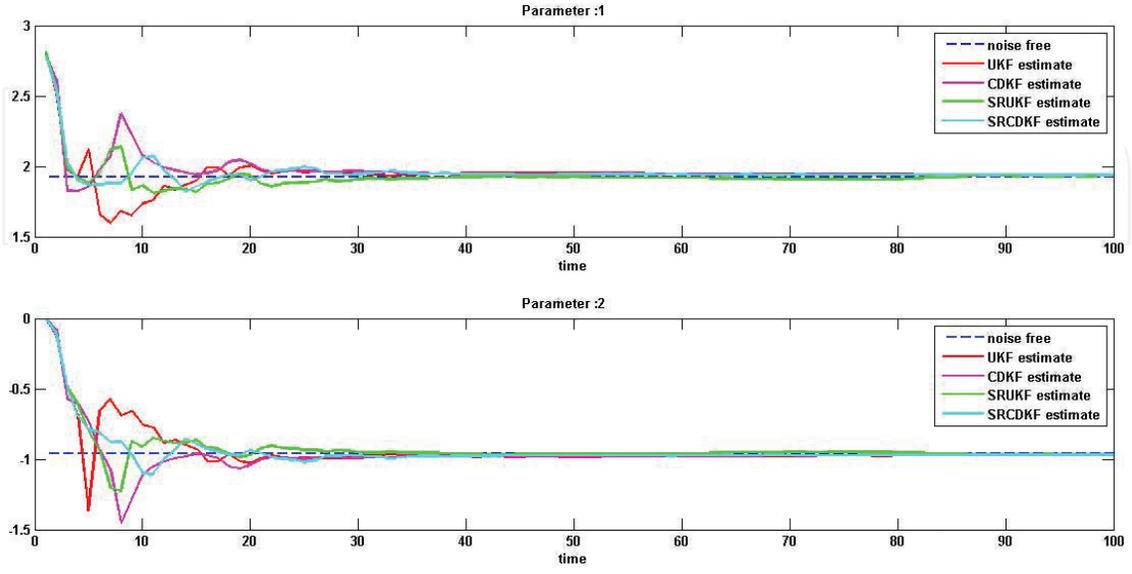

Figure 4. Estimation of the model parameters (P_1, P_2) using UKF, CDKF, SRUKF and SRCDKF.

http://dx.doi.org/10.5772/63728

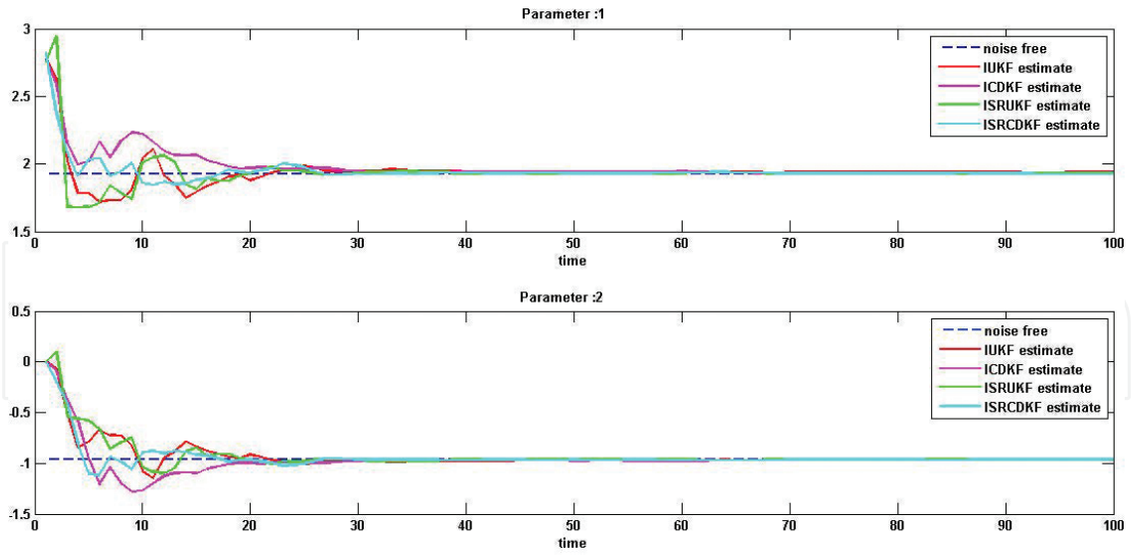

Figure 5. Estimation of the model parameters (P_1, P_2) using IUKF, ICDKF, ISRUKF and ISRCDKF.

4.1.3.1. Root Mean Square Error analysis

The effects of the practical challenges on the performances of the UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF for state and parameter estimation are investigated in the next section.

4.1.3.1.1. Effect of number of state and parameter to estimate on the estimation RMSE

To study the effect of the number of states and parameters to be estimated on the estimation performances of UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF, the estimation performance is analyzed for different numbers of estimated states and parameters. Here, we will consider two cases, which are summarized below. In all cases, it is assumed that the state x_k is measured.

Case 1: the state x_k along with the first parameter P_1 will be estimated.

Case 2: the state x_k along with the two parameters P_1 and P_2 will be estimated.

The estimation of the state variables and parameter(s) for these two cases is performed using UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF, and the simulation results for the state variables and the model parameters for the two cases are shown in **Tables 2** and **3**. For example, for case 1, **Table 2** compares the estimation RMSEs for the two state variables x_k (with respect to the noise-free data) and the mean of the estimated parameter P_1 at steady state (i.e., after convergence of parameter(s)) using the estimation methods. **Tables 2** and **3** also present similar comparisons for cases 1 and 2, respectively.

Technique	x_1 (RMSE)	x_2 (RMSE)	P_1 (mean)	Time execution (s)	Technique	x_1 (RMSE)	x_2 (RMSE)	P_1 (mean)	Time execution (s)
UKF	0.4221	0.5418	2.2453	0.3906	IUKF	0.3854	0.5093	1.9826	0.6963
CDKF	0.4192	0.5205	2.2232	0.3696	ICDKF	0.3827	0.4920	1.9786	0.5160
SRUKF	0.4063	0.4978	2.2228	0.3835	ISRUKF	0.3757	0.4748	1.9661	0.6798
SRCDKF	0.3970	0.4943	2.1858	0.3420	ISRCDKF	0.3737	0.4720	1.9297	0.5154

Table 2. Root mean square errors of estimated state variables and mean of estimated parameter: case 1.

The results also show that the number of parameters to estimate affects the estimation accuracy of the state variables. In other words, for all the techniques the estimation RMSE of x_k increases from the first comparative study (where only the state variables are estimated) to case 1 (where the states and one parameter P_1 is estimated) to case 2 (where the states and two parameters, P_1 and P_2, are estimated). For example, the RMSEs obtained using ISRCDKF for x_1 in the first comparative study and cases 1–2 of the second comparative study are 0.3121, 0.3737 and 0.3846, respectively, which increase as the number of estimated parameters increases (see **Tables 2** and **3**). This observation is valid for the other state estimation techniques.

It can also be shown from **Tables 2** and **3** that, for all the techniques, estimating more model parameters affects the estimation accuracy. The ISRCDKF method, however, still provides advantages over other methods in terms of the estimation accuracy.

Technique	x_1 (RMSE)	x_2 (RMSE)	P_1 (mean)	P_2 (mean)	Technique	x_1 (RMSE)	x_2 RMSE)	P_1 (mean)	P_2 (mean)
UKF	0.1962	0.6590	1.9484	-0.9798	IUKF	0.4056	0.4927	19408	-0.9721
CDKF	0.4170	0.4932	1.9482	-0.9786	ICDKF	0.4012	0.4908	1.9389	-0.9720
SRUKF	0.4133	0.4977	1.9481	-0.9776	ISRUKF	0.3989	0.4843	1.9342	-0.9677
SRCDKF	0.4090	0.4956	1.9436	-0.9741	ISRCDKF	0.3846	0.4875	1.9305	-0.9486

Table 3. Root mean square errors of estimated state variables and mean of estimated parameter: case 2.

4.1.3.1.2. Effect of noise content on the estimation RMSE

It is assumed that a noise is added to the state variable. In order to show the performance of the estimation algorithms in the presence of noise, three different measurement noise values, 10^{-1}, 10^{-2} and 10^{-3}, are considered. The simulation results of estimating the state x_k using the UKF, IUKF, CDKF, ICDKF, SRUKF, ISRUKF, SRCDKF and ISRCDKF methods when the noise levels vary in $\{10^{-1}, 10^{-2} \text{ and } 10^{-3}\}$ are shown in **Table 4**.

http://dx.doi.org/10.5772/63728

Noise levels		UKF	CDKF	SRUKF	SRCDKF	IUKF	ICDKF	ISRUKF	ISRCDKF
10^{-1}	x_1	0.3539	0.3512	0.3495	0.3324	0.3342	0.3265	0.3254	0.3121
	x_2	0.4658	0.4593	0.4590	0.4593	0.4341	0.4315	0.4256	0.4213
10^{-2}	x_1	0.1293	0.1264	0.1208	0.1174	0.1134	0.1095	0.1075	0.1066
	x_2	0.3564	0.3493	0.3474	0.3457	0.3440	0.3371	0.3355	0.3314
10^{-3}	x_1	0.0460	0.0454	0.0448	0.0446	0.0436	0.0415	0.0394	0.0376
	x_2	0.3426	0.3360	0.3188	0.3062	0.2989	0.2918	0.2875	0.2830

Table 4. Root mean square errors (RMSEs) of the estimated states for different noise levels.

In other words, for the estimation techniques, the estimation RMSEs of x_k increase from the first comparative study (noise value = 10^{-1}) to case (where the noise value = 10^{-3}). For example, the RMSEs obtained using ISRCDKF for x_1 where the noise level in $\{10^{-1}, 10^{-2}$ and $10^{-3}\}$ are 0.3121, 0.1066 and 0.0376, which increase as the noise variance increases (refer to **Table 4**).

5. Conclusions

In this chapter, various SPKF-based methods are used to estimate nonlinear state variables and model parameters. They are compared for the estimation performance in two comparative studies. In the first comparative study, the state variables are estimated from noisy measurements of these variables, and the several estimation methods are compared by estimating the RMSE with respect to the noise-free data. In the second comparative study, of the state variables as well as that the model parameters are estimated. Comparing the performances of the several state estimation extensions, the impact of the number of estimated model parameters on the convergence and accuracy of these methods is also evaluated. The results of the second comparative study show that, for all the techniques, estimating more model parameters affects the estimation accuracy as well as the convergence of the estimated states and parameters. The iterated square-root central difference Kalman method, however, still provides advantages over other methods in terms of the estimation accuracy, convergence and execution times.

Acknowledgements

This work was made possible by NPRP grant NPRP7-1172-2-439 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Author details

Marwa Chaabane[1,2], Imen Baklouti[2,4], Majdi Mansouri[1*], Nouha Jaoua[2], Hazem Nounou[1], Mohamed Nounou[3], Ahmed Ben Hamida[2] and Marie-France Destain[4]

*Address all correspondence to: majdi.mansouri@qatar.tamu.edu

1 Electrical and Computer Engineering Program, Texas A&M University at Qatar, Doha, Qatar

2 Advanced Technologies for Medicine and Signals, National Engineering School of Sfax, Tunisia

3 Chemical Engineering Program, Texas A&M University at Qatar, Doha, Qatar

4 Biosystems Engineering Department, GxABT, University of Liege, Gembloux, Belgium

References

[1] W. Guo. Dynamic state-space models. Journal of Time Series Analysis. 2003;24(2):149–158.

[2] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. Journal of Computational. 1996;5(1):1–25.

[3] J. H. Kotecha and P. M. Djurić. Gaussian sum particle filtering for dynamic state space models. Acoustics, Speech and Signal Processing. 2001;6(IEEE, 2001): 3465–3468.

[4] S.-G. Cao, N. W. Rees, and G. Feng. Analysis and design of fuzzy control systems using dynamic fuzzy-state space. Fuzzy Systems, IEEE Transactions on. 1999;7(2):192–200.

[5] C.-J. Kim and C. R. Nelson. State-space models with regime switching: classical and Gibbs-sampling approaches with. MIT Press, Cambridge. 1999;2.

[6] A. Nabavi-Niaki and M. R. Iravani. Steady-state and dynamic models of unified power flow controller (UPFC) for power. Power Systems, IEEE Transactions. 1996;11(4):1937–1943.

[7] M. Mansouri, O. Ilham, H. Snoussi, and C. Richard. Adaptive quantized target tracking in wireless sensor networks. Wireless Networks. 2011;17(7):1625–1639.

[8] M. Mansouri, M. Mohamed-Seghir, H. N. Nounou, M. N. Nounou, and H. Abu-Rub. Bayesian methods for time-varying state and parameter estimation in induction machines. International Journal of Adaptive Control and Signal Processing, 2015, 29(7), 905-924.

[9] M. Mansouri, H. N. Nounou, M. N. Nounou, and A. A. Datta. State and parameter estimation for nonlinear biological phenomena modeled by S-systems. Digital Signal Processing. 2014;28:1–17.

[10] M. Mansouri, H. Nounou, and M. Nounou. Modeling of nonlinear biological phenomena modeled by S-systems. Mathematical Biosciences. 2014;249:75–91.

[11] A. Simoglou, E. Martin, and A. Morris. Statistical performance monitoring of dynamic multivariate processes using state space modelling. Computers and Chemical Engineering. 2002;26(6):909–920.

[12] M. Mansouri, H. Nounou, and M. Nounou. State estimation of a chemical reactor process model-a comparative study. IEEE 10th International Multi-Conference, in Systems, Signals & Devices (SSD), 2013. 2013:1–6.

[13] M. Mansouri, B. Dumont, and M.-F. Destain. Modeling and prediction of time-varying environmental data using advanced Bayesian methods. Exploring Innovative and Successful Applications of Soft Computing. 2013:112.

[14] H. van der Kooij, R. Jacobs, B. Koopman, and F. van der Helm. An adaptive model of sensory integration in a dynamic. Biological Cybernetics. 2001;84(2):2103–2115.

[15] M. Mansouri and M.-F. Destain. Predicting biomass and grain protein content using Bayesian methods. Stochastic Environmental Research and Risk Assessment. 2015;29(4):1167–1177.

[16] G. L. Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. Journal of Power Sources. 2004;134(2):277–292.

[17] D. Dochain. State and parameter estimation in chemical and biochemical processes: a tutorial. Journal of Process. 2003;13(8):801–818.

[18] T. A. Wenzel, K. Burnham, M. Blundell, and R. Williams. Dual extended Kalman filter for vehicle state and parameter. Vehicle System Dynamics. 2006;44(2):153–171.

[19] G. Evensen. The ensemble Kalman filter for combined state and parameter estimation. Control Systems, IEEE. 2009;29(3):83–104.

[20] J. Ching, J. L. Beck, and K. A. Porter. Bayesian state and parameter estimation of uncertain dynamical systems. Probabilistic Engineering Mechanics. 2006;21(1):81–96.

[21] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser. Dual state–parameter estimation of hydrological models using ensemble Kalman filter. Advances in Water Resources. 2005;28(2):135–147.

[22] P. Moireau, D. Chapelle, and P. Le Tallec. Joint state and parameter estimation for distributed mechanical systems. Computer Methods in Applied Mechanics and Engineering. 2008;197(6):659–677.

[23] S. Aborhey and D. Williamson. State and parameter estimation of microbial growth processes. Automatica. 1978;14(5):493–498.

[24] S. Rao, M. Buss, and V. Utkin. Simultaneous state and parameter estimation in induction motors using first-and second order sliding modes. Industrial Electronics, IEEE Transactions on. 2009;56(9):3369–3376.

[25] A. V. Savkin and I. R. Petersen. Recursive state estimation for uncertain systems with an integral quadratic constraint. IEEE Transactions on Automatic Control. 1995;40(6): 1080–1083.

[26] D. Simon. Optimal state estimation: Kalman, H8, and nonlinear approaches. John Wiley and Sons, 2006, pages 552, 978-0-471-70858-2.

[27] M.S. Grewal, and A.P. Andrews, Kalman filtering: theory and practice using MATLAB. Theory and Practice, 2001, 5(5), pp.634-6.

[28] V. Aidala. Parameter estimation via the Kalman filter. IEEE Trans. on Automatic Control. 1977;22(3):471–472.

[29] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. International Journal of Computer Vision. 1989;3(3):209–238.

[30] S. Julier and J. Uhlmann. New extension of the Kalman filter to nonlinear systems. Proceedings of SPIE. 1997;3(1):182–193.

[31] L. Ljung. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. IEEE Trans. 1979;24(1):36–50.

[32] Y. Kim, S. Sul, and M. Park. Speed sensorless vector control of induction motor using extended Kalman filter. IEEE Trans. on Industrial Applications. 1994;3(5):1225–1233.

[33] E. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000. 2000;153–158.

[34] R. Van Der Merwe and E. Wan. The square-root unscented Kalman filter for state and parameter-estimation. In Acoustics, Speech, and Signal Processing, 2001. Proceedings (ICASSP'01) 2001 IEEE International Conference. 2001;6:3461–3464.

[35] S. Sarkka. On unscented Kalman filtering for state estimation of continuous-time nonlinear systems. IEEE Trans Automatic Control. 2007;52(9):1631–1641.

[36] M. Mansouri, H. Nounou, M. Nounou, and A. A. Datta. Modeling of nonlinear biological phenomena modeled by S-systems. In: Biomedical Engineering and Sciences (IECBES), 2012 IEEE EMBS Conference on IEEE. 2012;305–310.

[37] J. Zhu, N. Zheng, Z. Yuan, Q. Zhang, X. Zhang, and Y. He. A slam algorithm based on the central difference Kalman filter. In: Intelligent Vehicles Symposium, 2009 IEEE. 2009;123–128.

[38] M. M. Andreasen. Non-linear DSGE models and the central difference Kalman filter. Journal of Applied Econometrics. 2013;28(6):929–955.

[39] S. Holmes, G. Klein, and D. W. Murray. A square root unscented Kalman filter for visual monoslam. In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE. 2008;3710–3716.

[40] M. Huang, W. Li, and W. Yan. Estimating parameters of synchronous generators using square-root unscented Kalman filter. Electric Power Systems Research. 2010;80(9):1137–1144.

[41] M. NøRgaard, N. K. Poulsen, and O. Ravn. New developments in state estimation for nonlinear systems. Automatica. 2000;36(11):1627–1638.

[42] R. Zhan and J. Wan. Iterated unscented Kalman filter for passive target tracking. Aerospace and Electronic Systems, IEEE Transactions on. 2007;43(3):1155–1163.

[43] XK. Yiyu. Iterated unscented kalman filter. Journal of Huazhong University of Science and Technology (Nature Science Edition) 2007, 11, 006.

[44] L.M. Sibley, Gabe, Gaurav S. Sukhatme, and Larry Matthies. "The Iterated Sigma Point Kalman Filter with Applications to Long Range Stereo." In Robotics: Science and Systems, vol. 8, no. 1, pp. 235-244. 2006.

[45] G. Sibley, G. Sukhatme, and L. Matthies. The iterated sigma point Kalman filter with applications to long range stereo. In Robotics: Science and Systems. 2006;8(1):235–244.

[46] M. Mansouri, O. Avci, H. Nounou, and M. Nounou. A comparative assessment of nonlinear state estimation methods for structural health monitoring. Model Validation and Uncertainty Quantification. 2015;3:45–54.

[47] J. Heming, M. Hongwei, Gao Wei, Che Yanting. Application of ISRCDKF in SINS initial alignment with large azimuth misalignment angle. AISS (Advances in Information Sciences and Service Sciences). 2013;5:746–755.

[48] R. Van Der Merwe, E. A. Wan, S. Julier et al. Sigma-point Kalman filters for nonlinear estimation and sensor-fusion: applications to integrated navigation. In: Proceedings of the AIAA Guidance, Navigation & Control Conference. 2004;16–19.

IntechOpen