

# Particle-Based Tuning of the Unscented Kalman Filter

Leonardo Azevedo Scardua<sup>1,2</sup> · José Jaime da Cruz<sup>2</sup>

Received: 27 May 2015 / Revised: 5 October 2015 / Accepted: 8 November 2015 / Published online: 11 December 2015  
© Brazilian Society for Automatics–SBA 2015

**Abstract** The unscented Kalman filter (UKF) is one of the most used approximate solutions to the problem of nonlinear filtering. It is relatively easy to implement, and it produces better state estimates than the extended Kalman filter, especially when the nonlinearities of the dynamic system are significant. The quality of the estimates yielded by the UKF is dependent on the tuning of the parameters that govern the unscented transform (UT). To the user, manually tuning the UT means picking proper values for three scalar variables with almost no theoretical guidance. To help relieve the user from this burden, we approach the tuning of the UT parameters as an optimization problem and propose a tuning algorithm based on ideas of the bootstrap particle filter. The proposed algorithm is analyzed and numerically tested against both a set of popular nonlinear filters and a recently published model-based tuning algorithm.

**Keywords** UKF automatic tuning · UT automatic tuning · Nonlinear filtering

## 1 Introduction

According to [Turner and Rasmussen \(2012\)](#), three of the most used approximate solutions to the problem of nonlinear filtering are the unscented Kalman filter (UKF; [Julier and](#)

[Uhlmann 1997](#)), the extended Kalman filter (EKF; [Maybeck 1979](#)) and the cubature Kalman filter (CKF; [Arasaratnam and Haykin 2009](#)).

The EKF has been the most popular nonlinear filter of the last decades ([Psiaki 2013](#)). It is nonetheless known that the first-order Taylor series approximations used by the filter can hinder its applicability in situations where nonlinearities are significant, or when the necessary Jacobians are difficult to obtain. In spite of its known shortcomings, the EKF is still popular, probably due to its very low computational cost.

The CKF was described by its authors as being numerically more accurate and stable than the UKF [see Section VII of [Arasaratnam and Haykin \(2009\)](#)]. It is though known that the CKF can be reduced to a particular case of the UKF ([Turner and Rasmussen 2012](#)), when  $\alpha = 1$ ,  $\beta = 0$  and  $\kappa = 0$ .

The UKF is known to be able to achieve better estimation performance than the EKF, although both filters display the same order of computational complexity ([Wan and van der Merwe 2000](#)). To yield better state estimates than the EKF, the UKF resorts to the UT ([Julier and Industries 2002](#)) to properly propagate the mean and covariance of the state estimate through the nonlinear functions of the dynamic system. To work well, the UT demands the tuning of its three scalar parameters ( $\alpha$ ,  $\beta$  and  $\kappa$ ), a difficult task given the lack of comprehensive theoretical guidance for picking the parameter values. A common tuning strategy is given in [Julier et al. \(2000\)](#), which is to make  $\kappa = 3 - n_x$ <sup>1</sup>, where  $n_x$  is the dimension of the state vector. We call the UKF tuned with such heuristic the default UKF (Ukfd). This strategy, though, may produce poor results.

✉ Leonardo Azevedo Scardua  
lascardua@ifes.edu.br

José Jaime da Cruz  
jaime@lac.usp.br

<sup>1</sup> Federal Institute of Espírito Santo, Serra, ES, Brazil

<sup>2</sup> Automation and Control Laboratory, University of São Paulo, São Paulo, SP, Brazil

<sup>1</sup> [Julier et al. \(2000\)](#) use the nonscaled version of the UT, which can be obtained by making  $\alpha = 1$  and  $\beta = 0$  in the regular (scaled) UT.

To fulfill the need for more effective tuning strategies, a number of recently published works approach the tuning of the UT as a numerical optimization problem.

Abbeel et al. (2005) propose to use machine learning algorithms (Barber 2012) to automatically tune the covariance matrices of both the process and measurement noises of a Kalman filter. Despite not dealing with the UT, this reference is important because it approaches the tuning of the noise covariance matrices as an optimization problem guided by the maximization of the likelihood of the measurements taken from the dynamical system. The maximization of such likelihood was later used in a number of UT tuning works (Dunik et al. 2012; Turner and Rasmussen 2012; Straka et al. 2014).

In Sakai and Kuroda (2010), values for both the UT parameters and the noise covariance matrices are obtained by offline optimization. The optimization process, nonetheless, demands the availability of highly accurate measurements of the state vector or at least of part of it, reducing the applicability of the approach.

An online adaptation strategy of the scaling parameter  $\kappa$  is proposed in Dunik et al. (2010, 2012), Straka et al. (2012, 2014) and Scardua and da Cruz (2015). Under the proposed strategy, at each filtering iteration, the scaling parameter  $\kappa$  is selected from a discrete set of possible values, while keeping  $\alpha = 1$  and  $\beta = 0$ . A value for  $\kappa$  is selected each time a measurement is received by the filter. The chosen value is the one that maximizes a given performance criterion, such as the likelihood of the measurements received by the filter. The proposed strategy is claimed to be able to cope with changes in the operating point of the dynamical system, but the selection of values for  $\kappa$  demands the realization of extra measurement update steps, entailing reasonable computational cost.

The machinery of model-based optimization (Forrester et al. 2008) was used for tuning all three UT parameters by Turner and Rasmussen (2010) and Turner and Rasmussen (2012). The optimization seeks to find values for the UT parameters that maximize the popular upper confidence bound (UCB) criterion (Cox and John 1997). The only data needed from the original dynamical system are the measurements which would be regularly available for the filter. The proposed approach does not increase the computational cost of the UKF, but confidence bound criteria are known to be sensitive to the tuning of the parameter that controls the exploration/exploitation trade-off (Jones 2001). Moreover, the algorithms involved in model-based optimization can be relatively complex to implement. We call UkfO the UKF that stems from this optimization approach.

We also approach the tuning of the UT parameters as an optimization problem. We propose a search method that is based on ideas of the BSF (Gordon et al. 1993). The optimization process is guided solely by noisy measurements of

the nonlinear dynamic system that is to be filtered. The search space is a cube in the Euclidean space, with each dimension corresponding to one of the UT parameters. The points in the cube are called particles. The tuning is performed offline, thus incurring no extra cost for the filter during runtime. We call PfUkf the UKF tuned with the proposed optimizer.

The rest of this paper is organized as follows. Section 2 gives the necessary background on the UT, showing how the UT parameters influence the estimates produced by the UKF. Section 3 describes how the tuning of the UT parameters was approached as an optimization problem. Section 4 justifies why a direct search algorithm would be suitable to the tuning of the UT. Section 5 describes the optimization algorithm developed for the task. Section 6 describes the numerical testing of the filters, and Sect. 7 presents the final comments.

## 2 The Unscented Kalman Filter

The discrete-time nonlinear filtering problem consists in estimating  $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ , the marginal distribution of the current state  $\mathbf{x}_k$  of a discrete-time dynamic system, given the sequence of measurements  $\mathbf{y}_{1:k} = \{\mathbf{y}_1, \dots, \mathbf{y}_k\}$ .

The sequential Bayesian filter (Candy 2009) provides optimal solution to the discrete-time nonlinear filtering problem. To this end, the filter alternates between prediction and correction steps. The prediction step estimates  $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ , the conditional probability distribution of the state at time instant  $k$ , given both the measurements available at time instant  $k - 1$  and the transition probability  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ , which is determined by the dynamic model. The correction step uses measurement  $\mathbf{y}_k$  and the distribution  $p(\mathbf{y}_k | \mathbf{x}_k)$  given by the measurement model to generate  $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ , the corrected conditional probability distribution of  $\mathbf{x}_k$ . The sequential Bayesian filter equations are

$$\begin{aligned}
 p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) &= \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} \\
 p(\mathbf{y}_k | \mathbf{y}_{1:k-1}) &= \int p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) d\mathbf{x}_k \\
 p(\mathbf{x}_k | \mathbf{y}_{1:k}) &= \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})}. \tag{1}
 \end{aligned}$$

A closed-form solution to (1) can only be found in specific situations. One such situation arises when the dynamic system is linear with additive Gaussian noise. In this case, the required distributions are Gaussian and the solution is the Kalman filter [Section 5.3 of Candy (2009)]. When the dynamic system is nonlinear, it is necessary to resort to approximate solutions.

One type of approximate solution to (1) is obtained when the probability distributions involved are assumed to be Gaussian. This approach yields the so-called Gaussian filters

(Wu et al. 2006). It then suffices to calculate the necessary means and covariances, in order to completely specify such distributions.

Gaussian filters feature a general form, from which stem a number of popular nonlinear filters (Wu et al. 2006), such as the EKF, the CKF and the UKF. To provide further context to the present discussion, let us define that the discrete-time nonlinear dynamic system model used in the nonlinear filtering problem is given by

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{y}_k &= h(\mathbf{x}_k) + \mathbf{v}_k,\end{aligned}\quad (2)$$

where  $f(\cdot)$  and  $h(\cdot)$  are known functions,  $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$  is the process noise, and  $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$  is the measurement noise. Both noises are white and uncorrelated with each other and with the initial system state.

The general Gaussian filter uses a Kalman-like structure to completely address the estimation task for system (2) (Wu et al. 2006). The equations of the additive noise general Gaussian filter are [Algorithm 6.3 of Sarkka (2013)]:

*Prediction*

$$\begin{aligned}\mathbf{m}_k^- &= \int f(\mathbf{x}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1} | \mathbf{m}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_{k-1} \\ \mathbf{P}_k^- &= \int (f(\mathbf{x}_{k-1}) - \mathbf{m}_k^-) (f(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T \\ &\quad \mathcal{N}(\mathbf{x}_{k-1} | \mathbf{m}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_{k-1} + \mathbf{Q}_{k-1}\end{aligned}\quad (3)$$

*Update*

$$\begin{aligned}\boldsymbol{\mu}_k &= \int h(\mathbf{x}_k) \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^-, \mathbf{P}_k^-) d\mathbf{x}_k \\ \mathbf{S}_k &= \int (h(\mathbf{x}_k) - \boldsymbol{\mu}_k) (h(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^-, \mathbf{P}_k^-) d\mathbf{x}_k \\ &\quad + \mathbf{R}_k \\ \mathbf{C}_k &= \int (\mathbf{x}_k - \mathbf{m}_k^-) (h(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^-, \mathbf{P}_k^-) d\mathbf{x}_k \\ \mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1} \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k (\mathbf{y}_k - \boldsymbol{\mu}_k) \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T\end{aligned}\quad (4)$$

Different approaches to solving the moment integrals in (3) and (4) give rise to different Gaussian filters (Wu et al. 2006). Cubature rules give rise to the CKF. Approximating functions  $f(\cdot)$  and  $g(\cdot)$  with first-order Taylor series gives rise to the first-order EKF, and using the UT to approximately calculate those moment integrals gives rise to the UKF.

The UKF thus uses the UT to approximately compute the mean and covariance of a random variable  $\mathbf{y}$  that results from a nonlinear transformation  $\mathbf{h}(\cdot)$  of a Gaussian random variable  $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P}_x)$ , with  $\mathbf{x} \in \mathfrak{R}^{n_x}$ .

To this end, the dimensionless UT parameters  $\{\alpha, \beta, \kappa\} \in \mathfrak{R}^+$  are used to calculate a set of  $2n_x + 1$  deterministic points (sigma points) and weights. The sigma points are given by

$$\begin{aligned}\mathcal{X}_0 &= \bar{\mathbf{x}} \\ \mathcal{X}_i &= \bar{\mathbf{x}} + \left( \sqrt{(n_x + \lambda) \mathbf{P}_x} \right)_i \quad i = 1, \dots, n_x \\ \mathcal{X}_i &= \bar{\mathbf{x}} - \left( \sqrt{(n_x + \lambda) \mathbf{P}_x} \right)_i \quad i = n_x + 1, \dots, 2n_x,\end{aligned}\quad (5)$$

where  $(\cdot)_i$  denotes the  $i$ th column of the matrix and  $\lambda$  is a scaling parameter given by

$$\lambda = \alpha^2 (n_x + \kappa) - n_x. \quad (6)$$

The corresponding weights are

$$\begin{aligned}w_0^{(m)} &= \frac{\lambda}{n_x + \lambda} \\ w_0^{(c)} &= \frac{\lambda}{n_x + \lambda} + (1 - \alpha^2 + \beta) \\ w_i^{(m)} &= w_i^{(c)} = \frac{\lambda}{2(n_x + \lambda)}.\end{aligned}\quad (7)$$

Sigma points  $\mathcal{X}_i$  are then submitted to the nonlinear transformation  $\mathbf{h}(\cdot)$ , yielding

$$\mathcal{Y}_i = \mathbf{h}(\mathcal{X}_i) \quad i = 0, \dots, 2n_x. \quad (8)$$

Finally, the first two moments of  $\mathbf{y}$  are approximated as

$$\begin{aligned}\bar{\mathbf{y}} &\approx \sum_{i=0}^{2n_x} w_i^{(m)} \mathcal{Y}_i \\ \mathbf{P}_y &\approx \sum_{i=0}^{2n_x} w_i^{(c)} (\mathcal{Y}_i - \bar{\mathbf{y}}) (\mathcal{Y}_i - \bar{\mathbf{y}})^T.\end{aligned}\quad (9)$$

The results yielded by the UT thus depend heavily on the values of  $\alpha$ ,  $\beta$  and  $\kappa$ . Accordingly, the quality of the UKF state estimates also depends heavily on such values. Properly tuning the UT parameters is hence essential to obtaining good UKF performance.

### 3 Approaching the Tuning of the UKF as an Optimization Problem

The UT tuning process described in this article seeks to improve the state estimation performance of the UKF without imposing extra computational costs to the filter. Moreover, the tuning process must be able to deal with the fact that it would be difficult to obtain an explicit model of the UKF applied to a particular dynamic system. These requirements

lead us to resort to an offline optimization strategy that could cope with black-box functions.

To perform the optimization of the UT parameters, it would thus be necessary to conceive a scalar goal function  $g(\cdot)$  that could properly weight points of the UT domain. The weight of a given point should be an indication of the state estimation performance of the UKF when tuned with that point. The tuning process would then consist in the maximization of a scalar goal function  $g(\theta)$ , where  $\theta$  is a point in the UT parameter space.

The tuning process described in this article adopts the goal function used in Turner and Rasmussen (2010). This goal function maximizes the likelihood of the measurements that serve as training data. This strategy was also applied to the offline tuning of the EKF noise covariance matrices in Abbeel et al. (2005) and to the online tuning of the  $\kappa$  UT parameter in Dunik et al. (2010).

For a given point  $\theta$ , this goal function runs a UKF tuned with  $\theta$  on one or more time series

$$Y_i = \{y_i(1), y_i(2), \dots, y_i(T)\}, \quad i = 1, \dots, n_Y \quad (10)$$

where the  $y_i(t)$  are the sequential noisy measurements of the dynamic system state that comprise time series  $Y_i$ .

For each  $Y_i$ , the means and covariances of the measurement predictions yielded by the UKF tuned with  $\theta$  are used to calculate the log of the conditional likelihood  $p(Y_i|\theta)$ , which is given by

$$\log p(Y_i|\theta) = \frac{\sum_{t=1}^T \log(\mathcal{N}(y_i(t)|\hat{y}_i(t), S_i(t)))}{T}, \quad (11)$$

where  $\hat{y}_i(t)$  and  $S_i(t)$  are, respectively, the mean and covariance of the measurement estimates yielded by the UKF tuned with  $\theta$ . The sequence of calculations performed by the goal function is shown in Algorithm 1. The goal function receives the UT parameters  $\theta$ , the training time series  $Y = \{Y_1, \dots, Y_{n_Y}\}$  and both the mean  $M_0$  and covariance  $P_0$  of the initial state. It then returns the conditional log-likelihood of  $Y$ .

---

**Algorithm 1** Goal Function

---

```

1: procedure GOALFUN( $\theta, Y, M_0, P_0$ )
2:   for each  $Y_i \in Y$  do
3:      $[\hat{Y}_i, S_i] \leftarrow$  UKF( $\theta, Y_i, M_0, P_0$ )  $\triangleright$  Run the UKF tuned with  $\theta$ 
4:      $\text{loglik}(Y_i) = \log p(Y_i, \theta)$   $\triangleright$  The conditional log-likelihood of  $Y_i$  is given by (11)
5:    $w_\theta \leftarrow \sum_1^{n_Y} \text{loglik}(Y_i)$   $\triangleright$  Return the conditional likelihood of  $Y$ 
6:   return  $w_\theta$ 

```

---

It is important to note that the maximization of the log-likelihood of the training data  $Y$  does not guarantee the

maximization of the state estimation performance of the filter, since no information about the true states that gave rise to the measurements is used (Turner and Rasmussen 2010). It would be more convenient to maximize the likelihood of the true states that gave rise to the measurements, but in many cases, it would be costly or even impossible to obtain such ground-truth data. Fortunately, the results described in Sect. 6 show that it is possible to significantly improve the performance of the UKF without having to obtain ground-truth data.

## 4 Stochastic Tuning of the UKF

As seen in Sect. 3, the tuning of the UT parameters will be approached as the optimization of a black-box function. Though its analytic form is unknown, a black-box function can be evaluated to yield its value, entailing the adoption of gradient-free optimization algorithms, such as pattern search (Hooke and Jeeves 1961) or genetic algorithms (Goldberg 1989). Nonetheless, existing gradient-free optimizers usually require a large number of function evaluations, and Algorithm 1 may be costly to evaluate in a number of situations, such as when the time series that comprise the training data are long or when the dynamic model (2) takes significant computational time to run.

A better alternative would thus be to conceive a direct search algorithm able to find a good tuning of the UT parameters without the need to perform too many evaluations of the goal function. This can be accomplished by assuming that the goal function shows some degree of smoothness.

If this is the case, then it is reasonable to expect that two points in the input space that are close to each other are likely to have similar values of the goal function. Hence, the value of the goal function at a given point can be estimated by the values of the goal function at points that are near. This property means that close to points where the value of the goal function is high, there are points where the value of the goal function is similarly high, or maybe even higher.

As a result, an optimization algorithm that searches close to points that display high values of the goal function described by Algorithm 1 would have the potential to find points where this goal function displays even higher values. Nonetheless, to minimize the chance of getting stuck at poor local maxima, the optimizer must implement some sort of exploration of the input space.

To balance the conflicting goals of searching near points where the goal function is known to be high (exploitation of what is already known) and searching in regions of the input space where there are few or no samples at all (exploration of the input space), an optimization algorithm could evaluate the goal function at points sampled from a given probability distribution. This proba-

bility distribution should be conceived in a way that it would produce points located near and also points located not so near existing samples. The former points would allow the algorithm to exploit what is already known about the shape of the goal function, while the latter points would allow the algorithm to explore the input space.

## 5 The Optimizer

The proposed tuning algorithm is based on ideas of the BSF (Gordon et al. 1993). The BSF approximates the probability distribution of the system state  $\mathbf{x}_k$  at time step  $k$  by a set of random samples (particles). The mean and covariance of the state estimate  $\hat{\mathbf{x}}_k$  yielded by the filter at time step  $k$  are, respectively, given by the mean and covariance of the existing particles at time step  $k$ . The filter works sequentially, performing two basic steps, a prediction step and an update step.

To briefly describe how the BSF works when applied to system (2), let us assume that, at time step  $k - 1$ , the set of particles is  $\mathbf{X}_{k-1} = \{\mathbf{x}_{k-1}(i) : i = 1, \dots, N\}$ , where  $\mathbf{x}_{k-1}(i)$  is the  $i$ th particle in  $\mathbf{X}_{k-1}$ . This set of particles is indeed a set of random samples from the probability distribution  $p(\mathbf{x}_{k-1}|\mathbf{D}_{k-1})$ , where  $\mathbf{D}_{k-1} = \{\mathbf{y}_i, i = 1, \dots, k - 1\}$  is the set of measurements available at  $k - 1$ .

The prediction step is aimed at predicting what would be the next state  $\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{D}_k)$ . To this end, each particle in  $\mathbf{X}_{k-1}$  is transformed by the state transition function  $f(\cdot)$ , yielding prior samples  $\mathbf{x}_k^-(i) = f(\mathbf{x}_{k-1}(i)) + \mathbf{w}_{k-1}$ , with  $\mathbf{w}_{k-1}$  sampled from the probability distribution of the system process noise.

The update step uses measurement  $\mathbf{y}_k$  to correct the prior samples. To this end, normalized weights are attributed to the prior samples, so that the weight  $q_i$  associated with prior sample  $\mathbf{x}_k^-(i)$  is proportional to the conditional likelihood  $p(\mathbf{y}_k|\mathbf{x}_k^-(i))$ . Each weight  $q_i$  is seen as the probability mass associated with the prior sample  $\mathbf{x}_k^-(i)$  and is hence used to form a discrete probability distribution over the prior samples. The corrected samples (particles)  $\mathbf{x}_k(i)$  are then obtained by sampling from this discrete distribution, so that the probability that a given prior particle will be resampled is equal to its weight.

The operations performed by the proposed tuning algorithm resemble those of the update step of the BSF. At each iteration of the proposed optimizer, the particles associated with the  $N_s$  highest values of the goal function are selected, forming a subset  $\Theta^*$ . The values of the goal function corresponding to the points in  $\Theta^*$  are then normalized to interval  $[0, 1]$ , in a way that the points in  $\Theta^*$  and the normalized weights form a discrete probability distribution. A population  $\Theta_{\text{best}}$  is then formed by randomly sampling  $N_s$  points from

this discrete distribution. In Sect. 6, the resampling is performed according to the stratified resampling strategy (Douc and Cappe 2005).

This resampling strategy steadily increases the number of copies of particles with high weights and decreases the number of copies of particles with low weights. This can be seen as a way to exploit the information conveyed by the existing particles, but it does not bring any new information to the optimization process. To bring new information to the optimization process, the algorithm samples  $N_s$  new particles from the distribution

$$p(\theta_{\text{new}}) = \mathcal{N}(\mathbf{M}_{\text{best}}, \mathbf{S}_{\text{best}}), \quad (12)$$

where  $\mathbf{M}_{\text{best}}$  and  $\mathbf{S}_{\text{best}}$  are, respectively, the mean and covariance of  $\Theta_{\text{best}}$ . The normal distribution was chosen because the UKF is a Gaussian filter.

The goal function is evaluated at  $\theta_{\text{new}}$ , providing them with weights. The new particles and their weights are then added to the current particle population, thus bringing new information to the optimization process.

Probability distribution (12) is interesting because it can yield samples located within varying distances of the best existing particles. Samples that are close to existing particles allow the tuning algorithm to exploit existing information about the goal function, while points that are farther away allow the algorithm to explore the input space.

A more systematic description of these general tuning steps is given in Algorithm 2, which receives as inputs the range of possible values for the particles, the budget of evaluations of the goal function, a set  $\Theta = \{\theta_1, \dots, \theta_N\}$  of  $N$  initial particles and the number  $N_s$  of new particles generated at each iteration.

---

### Algorithm 2 Particle-Based UT Tuning

---

- 1: **procedure** PFUKF- OPT( $N_s$ , Budget, Range,  $\Theta$ )
  - 2:    $\mathbf{W}_\Theta \leftarrow$  Evaluate the goal function on the particles in  $\Theta$  yielding weights for each of them.
  - 3:    $\Theta_{\text{best}} \leftarrow \Theta$
  - 4:   **for**  $i = 1$  to  $\frac{\text{Budget}}{N_s}$  **do**
  - 5:      $\mathbf{M}_{\text{best}} \leftarrow$  Calculate the mean of particles in  $\Theta_{\text{best}}$ .
  - 6:      $\mathbf{P}_{\text{best}} \leftarrow$  Calculate the covariance of particles in  $\Theta_{\text{best}}$ .
  - 7:      $\Theta_{\text{new}} \leftarrow$  Sample  $N_s$  particles from  $\mathcal{N}(\mathbf{M}_{\text{best}}, \mathbf{P}_{\text{best}})$ .
  - 8:      $\Theta_{\text{new}} \leftarrow \text{trim}(\Theta_{\text{new}}, \text{Range})$   $\triangleright$  The coordinate values of the particles  $\Theta_{\text{new}}$  may be trimmed to enforce the limits defined by Range.
  - 9:      $\mathbf{W}_{\text{new}} \leftarrow$  Evaluate the goal function on the new particles in  $\Theta_{\text{new}}$  yielding weights for each of them.
  - 10:      $\Theta \leftarrow \{\Theta; \Theta_{\text{new}}\}$   $\triangleright$  Append the new particles to  $\Theta$ .
  - 11:      $\mathbf{W}_\Theta \leftarrow \{\mathbf{W}_\Theta; \mathbf{W}_{\text{new}}\}$   $\triangleright$  Append the new weights to  $\mathbf{W}_\Theta$ .
  - 12:      $\mathbf{W}_\Theta \leftarrow$  Keep only the  $N_s$  highest weights in  $\mathbf{W}_\Theta$ .
  - 13:      $\Theta \leftarrow$  Keep only the particles corresponding to the new  $\mathbf{W}_\Theta$ .
  - 14:      $\Theta_{\text{best}} \leftarrow$  resample  $N_s$  particles from a discrete distribution formed with  $\Theta$  and  $\mathbf{W}_\Theta$
  - 15:    $\theta^* \leftarrow$  Return the particle associated with the highest weight.
-

## 6 Numerical Experiments

All numerical experiments were executed according to the following steps:

1. The training data  $\mathbf{Y}^{\text{train}} = \{\mathbf{Y}_1^{\text{train}}, \dots, \mathbf{Y}_{N_{\text{tr}}}^{\text{train}}\}$  were comprised of  $N_{\text{tr}}$  time series generated by Monte Carlo simulation of the nonlinear system;
2. The tuning for PfUkf was obtained by feeding Algorithm 2 with  $\mathbf{Y}^{\text{train}}$  and the other necessary parameters;
3. The tuning for UkfO was obtained by feeding the optimizer provided in Turner (2014) with  $\mathbf{Y}^{\text{train}}$  and the other necessary parameters;
4. The test data  $\mathbf{Y}^{\text{test}} = \{\mathbf{Y}_1^{\text{test}}, \dots, \mathbf{Y}_{N_{\text{ts}}}^{\text{test}}\}$  were comprised of  $N_{\text{ts}}$  different time series generated by Monte Carlo simulation of the nonlinear system;
5. The testing of the filters consisted in assessing the quality of the state estimates yielded by the filters on the true states corresponding to  $\mathbf{Y}^{\text{test}}$ .

In all experiments, the optimizations corresponding to steps 2 and 3 used the same set of 27 initial samples also used in Turner and Rasmussen (2010). The maximum number of evaluations of the goal function (budget) was 100, and the range of possible values for the UT parameters was

$$\begin{aligned} 0.01 &\leq \alpha \leq 4 \\ 0 &\leq \beta \leq 4 \\ 0 &\leq \kappa \leq 5. \end{aligned} \tag{13}$$

The numerical experiments also assumed that the covariance noise matrices  $\mathbf{Q}$  and  $\mathbf{R}$  were known, and the  $N_s$  free parameter of Algorithm 2 was always  $N_s = 10$ . The distribution of the initial state of the filters was the same distribution used to generate the training and test data.

### 6.1 Benchmarks

Three commonly adopted nonlinear filtering problems are used in the numerical experiments.

#### 6.1.1 Kitagawa

Variations of the Kitagawa (1996) model have been used as benchmark for nonlinear filters (Gordon et al. 1993; Turner and Rasmussen 2010, 2012). The model equations are

$$\begin{aligned} x_{k+1} &= 0.5x_k + \frac{25x_t}{(1+x_t^2)} + w_k, \quad w_k \sim \mathcal{N}(0, 0.2^2) \\ y_k &= 5\sin(2x_k) + v_k, \quad v_k \sim \mathcal{N}(0, 0.01^2). \end{aligned} \tag{14}$$

For the numerical experiment with Kitagawa, the training data consisted of a single time series of 1000 sequential measurements, and the test data consisted of 500 independent time series of 10 sequential measurements each. The initial state estimate for generating both the training and test data was  $\mathbf{x}_0 \sim \mathcal{N}(0, 0.5^2)$ .

#### 6.1.2 Sinusoid

The Sinusoid problem (Turner and Rasmussen 2010, 2012) is given by

$$x_{k+1} = 3 \sin(x_k) + w_k, \quad w_k \sim \mathcal{N}(0, 0.1^2) \tag{15}$$

$$y_k = \sigma(x_k/3) + v_k, \quad v_k \sim \mathcal{N}(0, 0.1^2), \tag{16}$$

where  $\sigma(\cdot)$  represents the logistic sigmoid.

For the numerical experiment with Sinusoid, the training data consisted of a single time series of 1000 sequential measurements, and the test data consisted of 500 independent time series of 100 sequential measurements each. The initial state distribution for both the training and test data was  $\mathbf{x}_0 \sim \mathcal{N}(0, 1^2)$ .

#### 6.1.3 Bearings Only

Different versions of the Bearings only are also popular (Dunik et al. 2012; Bar-Shalom et al. 2002). The model represents the problem of tracking a moving target based solely on measurements provided by one angular sensor. In this paper, the system state is the position of the target in the Cartesian plane. The model equations are

$$\begin{aligned} \mathbf{x}_{k+1} &= \begin{bmatrix} 0.9 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \\ y_k &= \tan^{-1} \left( \frac{x_{2,k} - \sin(k)}{x_{1,k} - \cos(k)} \right) + v_k, \quad v_k \sim \mathcal{N}(0, R), \end{aligned} \tag{17}$$

where

$$\begin{aligned} \mathbf{Q} &= \begin{bmatrix} 0.1 & 0.01 \\ 0.01 & 0.1 \end{bmatrix} \\ R &= 0.025. \end{aligned} \tag{18}$$

For the numerical experiment with Bearings Only, the training data consisted of a single time series of 1000 sequential measurements, and the test data consisted of 500 independent time series of 100 sequential measurements each. The distribution of the initial state for both the training and test data was

**Table 1** State errors computed from 500 Monte Carlo simulations for Kitagawa

	UkfD	CKF	EKF	UkfO	PfUkf
RMSE	2.824	3.416	6.387	0.432	0.391
MAE	0.491	0.967	1.921	0.179	0.169

$$\mathbf{x}_0 \sim \mathcal{N}\left(\begin{bmatrix} 20 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}\right). \quad (19)$$

## 6.2 Performance Criteria

The performances of the different filters are assessed with the root-mean-squared error (RMSE) and the mean absolute error (MAE) of the estimated states. Considering that the prediction errors in each component  $i$  of the state vector  $\mathbf{x}$  at time step  $k$  are given by

$$e_{i,k} = \hat{x}_{i,k} - x_{i,k}, \quad (20)$$

where  $\hat{x}_{i,k}$  and  $x_{i,k}$  are, respectively, the predicted and the real value of the  $i_{th}$  component of  $\mathbf{x}$  at time step  $k$ , the equations of the error criteria are

$$\text{RMSE} = \sqrt{\frac{1}{n_T} \sum_{k=1}^{n_T} \sum_{i=1}^{n_x} e_{i,k}^2} \quad (21)$$

and

$$\text{MAE} = \frac{1}{n_T} \sum_{k=1}^{n_T} \sum_{i=1}^{n_x} |e_{i,k}|, \quad (22)$$

where  $n_T$  is the number of time steps of the series of true data.

The two criteria provide different views of the estimation performances. In MAE, all errors are equally weighted, while in RMSE, bigger errors have more impact than small errors. We also employ boxplots of the squared state estimation errors, in order to provide further assessment of the variability of the predictions yielded by the filters.

## 6.3 Results

The state estimation errors for all three benchmark problems are shown in Tables 1, 2 and 3. The measurement estimation errors are shown in Tables 4, 5 and 6. The UT parameters used in the UKF filters are shown in Tables 7, 8 and 9.

**Table 2** State errors computed from 500 Monte Carlo simulations for Sinusoid

	UkfD	CKF	EKF	UkfO	PfUkf
RMSE	0.980	1.201	1.241	0.870	0.871
MAE	0.781	0.889	0.938	0.686	0.687

**Table 3** State errors computed from 500 Monte Carlo simulations for Bearings

	UkfD	CKF	EKF	UkfO	PfUkf
RMSE	3.263	4.153	7.230	1.765	1.701
MAE	2.371	3.135	7.490	1.504	1.483

**Table 4** Measurement errors computed from 500 Monte Carlo simulations for Kitagawa

	UkfD	CKF	EKF	UkfO	PfUkf
RMSE <sub>y</sub>	2.292	2.312	2.962	1.920	1.749
MAE <sub>y</sub>	1.285	1.323	1.637	1.050	1.007

**Table 5** Measurement errors computed from 500 Monte Carlo simulations for Sinusoid

	UkfD	CKF	EKF	UkfO	PfUkf
RMSE <sub>y</sub>	0.151	0.152	0.172	0.135	0.136
MAE <sub>y</sub>	0.121	0.119	0.135	0.107	0.107

**Table 6** Measurement errors computed from 500 Monte Carlo simulations for Bearings

	UkfD	CKF	EKF	UkfO	PfUkf
RMSE <sub>y</sub>	0.955	0.975	1.247	0.986	0.969
MAE <sub>y</sub>	0.580	0.634	0.883	0.532	0.526

**Table 7** Kitagawa—UT parameters computed from a budget of 100 points and 1 training time series

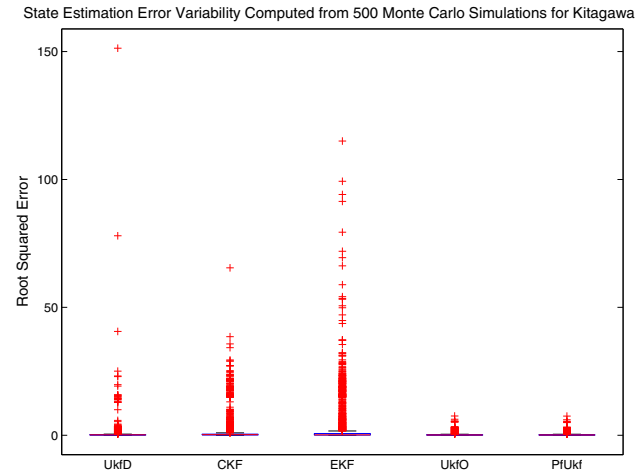
	$\alpha$	$\beta$	$\kappa$
UkfO	0.31929	1.99620	1.71978
PfUkf	0.82318	2.16289	0.02967

**Table 8** Sinusoid—UT parameters computed from a budget of 100 points and 1 training time series

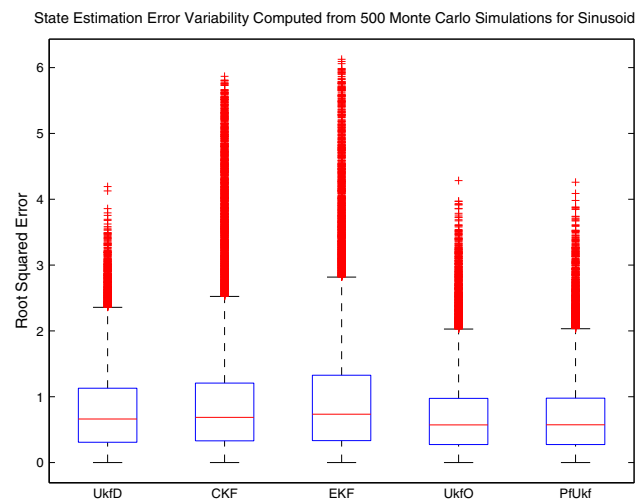
	$\alpha$	$\beta$	$\kappa$
UkfO	2.03334	0.24597	0.37981
PfUkf	2.37510	1.80140	0.01101

**Table 9** Bearings—UT parameters computed from a budget of 100 points and 1 training time series

	$\alpha$	$\beta$	$\kappa$
UkfO	1.55542	3.78321	0.55737
PfUkf	1.03510	3.69540	3.45110



**Fig. 1** Kitagawa

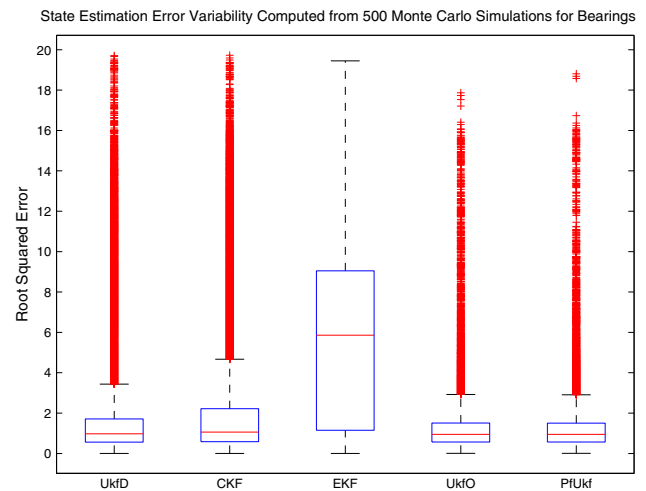


**Fig. 2** Sinusoid

The variability of the root-squared errors of the state estimates yielded by the filters is shown in Figs. 1, 2 and 3.

**6.4 Analysis**

Tables 1, 2 and 3, together with Figs. 1, 2 and 3, show that PfUkf performed better than the other filters in Kitagawa and Bearings only. In Sinusoid, its state estimation performance was on a par with that of UkfO. Except for UkfD in Sinusoid, the remaining filters state estimation performances were significantly worse than the performances of the tuned filters.



**Fig. 3** Bearings only

As for the measurement estimation performances, Tables 4, 5 and 6 show that there can be some disagreement with the state estimation performances, as expected.

**7 Conclusion**

We have presented a particle-based optimization algorithm for automatically tuning the UT parameters. The main advantage of the proposed approach is that it can significantly enhance the state estimation performance of the UKF without imposing extra computational cost to the filter at runtime.

Numerical experiments showed that the proposed algorithm yielded state estimation results better than filters EKF, CKF and UkfD. The results also were comparable to or better than the results of UkfO, a recently published model-based optimization algorithm. The proposed optimizer, though, is much easier to implement and also computationally simpler than UkfO.

**References**

Abbeel, P., Coates, A., Montemerlo, M., Ng, A. Y., & Thrun, S. (2005). Discriminative training of Kalman filters. In S. Thrun, G. S. Sukhatme, & S. Schaal (Eds.), *Robotics: Science and systems* (pp. 289–296). Cambridge: The MIT Press.

Arasaratnam, I., & Haykin, S. (2009). Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6), 1254–1269.

Bar-Shalom, Y., Kirubarajan, T., & Li, X. R. (2002). *Estimation with applications to tracking and navigation*. New York, NY: Wiley.

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge: Cambridge University Press.

Candy, J. V. (2009). *Bayesian signal processing: Classical modern and particle filtering methods*. New York, NY: Wiley.

Cox, D. D., & John, S. (1997). Sdo: A statistical method for global optimization. In *Multidisciplinary design optimization: State-of-the-art* (pp. 315–329).



- Douc, R., & Cappe, O. (Sept 2005). Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th international symposium on image and signal processing and analysis, 2005* (pp. 64–69).
- Dunik, J., Simandl, M., & Straka, O. (2010). Adaptive choice of scaling parameter in derivative-free local filters. In *13th conference on information fusion (FUSION), 2010* (pp. 1–8).
- Dunik, J., Simandl, M., & Straka, O. (2012). Unscented Kalman filter: Aspects and adaptive setting of scaling parameter. *IEEE Transactions on Automatic Control*, *57*(9), 2411–2416.
- Forrester, A. I. J., Sobester, A., & Keane, A. J. (2008). *Engineering design via surrogate modelling: A practical guide*. New Jersey: Wiley.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning* (1st ed.). Boston: Addison-Wesley Professional.
- Gordon, N., Salmond, D., & Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F, Radar and Signal Processing*, *140*(2), 107–113.
- Hooke, R., & Jeeves, T. A. (1961). "Direct search" solution of numerical and statistical problems. *Journal of the ACM (JACM)*, *8*(2), 212–229. doi:10.1145/321062.321069.
- Jones, D. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, *21*(4), 345–383.
- Julier, S. J., & Industries, I. (2002). The scaled unscented transformation. In *Proceedings of the IEEE American control conference* (pp. 4555–4559).
- Julier, S. J., Uhlmann, J. K., & Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, *45*(3), 477–482.
- Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of aerosense: The 11th international symposium on aerospace/defense sensing, simulations and controls*.
- Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, *5*(1), 1–25. doi:10.2307/1390750.
- Maybeck, P. S. (1979). *Stochastic models, estimation and control* (Vol. 1)., Mathematics in science and engineering Waltham: Academic Press.
- Psiaki, M. (2013). The blind tricyclist problem and a comparative study of nonlinear filters: A challenging benchmark for evaluating nonlinear estimation methods. *Control Systems, IEEE*, *33*(3), 40–54.
- Sakai, A., & Kuroda, Y. (2010). Discriminatively trained unscented Kalman filter for mobile robot localization. *Journal of Advanced Research in Mechanical Engineering (JARME)*, *1*(3), 153–161.
- Sarkka, S. (2013). *Bayesian filtering and smoothing*. Cambridge: Cambridge University Press.
- Scardua, L. A., & da Cruz, J. J. (2015). Adaptively tuning the scaling parameter of the unscented Kalman filter. In A. P. Moreira, A. Matos, G. Veiga (Eds.), *CONTROLO2014: Proceedings of the 11th Portuguese conference on automatic control, Lecture notes in electrical engineering, vol. 321* (pp. 429–438). Springer.
- Straka, O., Dunik, J., & Simandl, M. (June 2012). Scaling parameter in unscented transform: Analysis and specification. In *American control conference (ACC), 2012* (pp. 5550–5555).
- Straka, O., Dunik, J., & Simandl, M. (2014). Unscented Kalman filter with advanced adaptation of scaling parameter. *Automatica**50*(10), 2657–2664. <http://www.sciencedirect.com/science/article/pii/S000510981400346X>.
- Turner, R. (2014). Code for mlsp 2010. <http://mlg.eng.cam.ac.uk/rturner/code.html> (2011). Accessed 4 Nov 2014.
- Turner, R., & Rasmussen, C. E. (August 2010). Model based learning of sigma points in unscented Kalman filtering. In *Machine learning for signal processing (MLSP '10)* (pp. 178–183). Finland: Kittila.
- Turner, R., & Rasmussen, C. E. (2012). Model based learning of sigma points in unscented Kalman filtering. *Neurocomputing*, *80*, 47–53. doi:10.1016/j.neucom.2011.07.029.
- Wan, E. A., & van der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *The IEEE 2000 Adaptive systems for signal processing, communications, and control symposium 2000. AS-SPCC*. (pp. 153–158).
- Wu, Y., Hu, D., Wu, M., & Hu, X. (2006). A numerical-integration perspective on gaussian filters. *IEEE Transactions on Signal Processing*, *54*(8), 2910–2921.