A partially linearized sigma point filter for latent state estimation in nonlinear time series models

Paresh Date [a] Luka Jalen [a] Rogemar Mamon [b]

[a] *Center for the Analysis of Risk and Optimization Modelling Applications, Department of Mathematical Sciences, Brunel University, Uxbridge, United Kingdom, UB8 3PH*

[b] *Department of Statistical and Actuarial Sciences, The University of Western Ontario, London, Ontario, Canada N6A 5B7*

Abstract

A new technique for the latent state estimation of a wide class of nonlinear time series models is proposed. In particular, we develop a partially linearized sigma point filter in which random samples of possible state values are generated at the prediction step using an exact moment matching algorithm and then a linear programming-based procedure is used in the update step of the state estimation. The effectiveness of the new filtering procedure is assessed via a simulation example that deals with a highly nonlinear, multivariate time series representing an interest rate process.

Keywords: state estimation, sigma point filters, nonlinear time series

1 Introduction

This paper is concerned with the problem of latent state estimation for a nonlinear time series in discrete time. Our analysis will focus on the general class of systems with the following state space form:

$$\mathcal{X}(k+1) = \mathbf{f}\left(\mathcal{X}(k)\right) + \mathbf{g}\left(\mathcal{X}(k)\right)\mathcal{W}(k+1), \tag{1}$$

$$\mathcal{Y}(k) = \mathbf{h}\left(\mathcal{X}(k)\right) + \mathcal{V}(k), \tag{2}$$

where $\mathcal{X}(k)$ and $\mathcal{Y}(k)$ are the respective state vector and measurement vector at time t_k; $\mathbf{f}, \mathbf{g}$ and $\mathbf{h}$ are given nonlinear (vector-valued) functions; and both

$\mathcal{V}(k)$ and $\mathcal{W}(k)$ are symmetric vector-valued random variables. The time increment $t_k - t_{k-1}$ is assumed constant for all k. Moreover, we assume the noisy measurement vector $\mathcal{Y}(k)$ is available for every t_k. We wish to find an estimate of the random vector $\mathcal{X}(k)$ based on information up to (and including) time t_k.

In the special case when $\mathbf{f}$, $\mathbf{h}$ are affine in $\mathcal{X}(k)$, $\mathbf{g}$ is identity and $\mathcal{V}(k), \mathcal{W}(k)$ are Gaussian, the optimal recursive solution to the state estimation problem is given by linear Kalman filter, as first outlined in [1]. However, these assumptions are often not justified in practice. Nonlinear and non-Gaussian models are used to capture the dynamics of many phenomena occurring in the fields of radar navigation, climatology, geosciences and financial modeling, among others. The optimal recursive solution to the state estimation problem in nonlinear systems requires the propagation of full probability density; for an approximate solution of a more general nonlinear filtering problem, refer to [2], for example.

Current approaches designed to address the nonlinear filtering problems usually fall under one of the following approximate Bayesian filtering methods:

(a) Extended Kalman filter (EKF). Under this filter, equation (1) or its continuous time analogue is locally linearized resulting in a linear state space system. A Kalman filter is then employed to obtain the conditional state density of $\mathcal{X}(k)$. This approach is popular in engineering for more than three decades and standard textbooks such as [3], [4] carry an extensive discussion of its theoretical underpinnings and implementation. Extended Kalman filters based on piecewise linear discretization of the underlying continuous time stochastic differential equation (also called *local linearization filters*) are discussed in [5], among others. [6] offers a derivative-free version of extended Kalman filter which is particularly suitable for parameter estimation in nonlinear oscillators.

    If the system is indeed approximately linear then EKF will work well. Nevertheless, such assumption is often not easy to validate.

(b) Sequential Monte Carlo filtering. This is also known as *particle filtering*. For this technique, the required conditional density function of $\mathcal{X}(k)$ given measurement $\mathcal{Y}(k)$ at time t_k is represented by a set of random samples (or *particles*) and associated probability weights. The particles and weights are updated recursively as new measurements become available. Under fairly general conditions, the estimate approaches the optimal Bayesian estimate as the number of samples becomes sufficiently large; see [7], [8], [9] and the references therein. It can perform significantly better than EKF for highly nonlinear systems. However, as large number of samples need to be generated at each time t_k this technique is computationally quite expensive to implement.

(c) Unscented filter. The class of filters called *unscented filters* or *sigma point filters* provides an increasingly popular alternative to particle filters in signal processing applications and in geosciences. This type of filters may be viewed as a compromise between an EKF and a particle filter. Several applications in communication, tracking and navigation are discussed in [10] and [11]. Applications of this filtering technique are also reported in modeling population dynamics [12], in estimation of a parametric model for earthquake ground motion [13] and state estimation in electrochemical cells for battery management [14]. In [15], approximate methods are developed to deal with the multiplicative uncertainty in the observation equation under sigma point filtering framework .

The sigma point filters use closed-form recursive formulae based on the linear Kalman filter to propagate the mean and the covariance of state vector; this is essentially similar to the propagation equations in EKF. The system equations nonetheless are *not* linearized in this case. A small set of sample points (or *sigma points*) is generated and propagated through the nonlinear transformation to compute the conditional moment estimates. In lieu of using a large number of points and matching the distributions asymptotically (as in a particle filter), the sigma point filter uses a small set of sample points which are chosen such that some of the moment properties of the *a priori* distribution are matched exactly. While these filters have been used successfully in some engineering applications, they suffer from several shortcomings as elaborated below:

(a) The weights corresponding to probability masses are not guaranteed to be non-negative. Thus, the sample points generated in these filters do not necessarily define a valid distribution.

(b) There is no source of randomness in the filtering procedure because the algorithms for generating samples are purely deterministic.

The *ensemble filter*, a variant of sigma point filter, is commonly used in geosciences under which the state is sampled via the traditional Monte Carlo sampling techniques and sample conditional moments are used (i.e. the probability weights are assumed to be equal). This addresses both the shortcomings specified in (a) and (b). This technique was developed in [16] and could also be found in [17]. A review of ensemble filtering techniques appears in [18]. More recently, ensemble filter has been applied for magnetohydrodynamic systems in [19], with a view of using this type of techniques for solar storm prediction. A potential shortcoming of ensemble filter is that a very small number of samples is used as compared to the state space dimensions to compute the sample mean. This can lead to misleading results, as indicated in [20].

(c) The computation of a square root of the state covariance matrix at each time-step is required in a sigma point filter as well as in ensemble filters. If the number of states is very large, this presents a hurdle in its computational feasibility. This is usually the case for most problems in geosciences.

All of the above shortcomings were addressed in an earlier paper by the authors [20], where a new sigma point generation procedure was employed to match the first three moments exactly (as in the case of sigma point filters) while also using randomly generated samples (as in the case of ensemble filters). This algorithm avoided the requirement of repeated covariance matrix factorization in sigma point filters by only generating the samples of exogenous noise at each time-step. However, [20] still uses a formula based on (3) below for the Kalman filter-like update step. For linear Gaussian systems, Kalman filter is a conditional mean estimator. The recursive filtering equations for Kalman filter may be derived using a standard conditional mean relationship for two Gaussian variables $\mathcal{X}, \mathcal{Y}$ [21]:

$$\mathbb{E}\left(\mathcal{X}|\mathcal{Y}\right) = \mathbb{E}\left(\mathcal{X}\right) + \Sigma_{XY}\Sigma_{YY}^{-1}(\mathcal{Y} - \mathbb{E}(\mathcal{Y})), \tag{3}$$

where Σ_{YY} and Σ_{XY} are covariance matrices. Even when $\mathcal{V}(k)$ and $\mathcal{W}(k)$ are not Gaussian, Kalman filter is an optimal linear filter, in the sense that it yields the minimum variance over all linear filters. However, neither of these properties are relevant if the system is nonlinear. Hence, the motivation of using Kalman filtering state estimator equations based on (3) in the sigma point filter and its variants (including the one proposed in [20]) is not always clear.

The purpose of this paper is to propose an alternative heuristic for the state estimation of a nonlinear time series which does not use (3) in state estimation at all and and seeks a state estimate which *best matches* the observations in an appropriate deterministic sense. The algorithm uses linearized measurement equation but preserves the nonlinearity of the state evolution equation. Hence, we shall refer to this new filter as partially linearized sigma point filter (PLSPF). In PLSPF, we generate samples of exogenous noise in the state evolution equation (1) using the exact moment-matching procedure in [22]. These noise samples are used to obtain samples of state prediction. The measurement equation is linearized (similar to an extended Kalman filter) and a set of linear programming problems is solved to obtain samples of the updated state which best match the observations.

This paper is organized as follows. Section 2 sets out the algorithm in implementing the partially linearized sigma point filter while section 3 outlines briefly the underlying algorithm for sigma point generation. We include a demonstration of the algorithm's operation through a numerical example in section 4. More specifically, we illustrate the filtering procedure with a multivariate, nonlinear time series. Finally, some concluding remarks are given in section 5.

2 A partially linearized sigma point filter

Suppose at time t_{k+1}, the sample points (or *sigma points*)

$$\mathcal{W}^{(i)}(k+1),\ i = 1, 2, \ldots 2ns + 1$$

are available for the discrete time state space system (1)-(2) together with their associated probability weights p_i, $i = 1, 2, \ldots 2ns+1$. Here, n is the dimension of the vector $\mathcal{W}(k)$ (or in other words, the dimension of the state space in (1)). We assume that the collection of samples $\mathcal{W}^{(i)}(k+1)$ matches a given mean vector, covariance matrix and zero marginal skewness. The discussion of how to generate $\mathcal{W}^{(i)}(k+1)$ is postponed until the next section.

A set of s probability weights determines the $2ns+1$ support points above; details are given in the next section. In addition, the sample points of the updated state estimate $\mathcal{X}^{(i)}(k \mid k)$ are assumed to be available at time t_{k+1}.

Remark: We observe that $\mathcal{X}^{(i)}(k \mid k)$ is not sampled and the probability p_i for $\mathcal{W}^{(i)}(k+1)$ at each i is effectively assigned as the joint probability for the occurrence of $\left[\mathcal{W}^{(i)}(k+1)^\top \ \mathcal{X}^{(i)}(k \mid k)^\top\right]^\top$, where $\top$ denotes the transpose of a matrix. In this respect the procedure is similar to some of the ensemble filtering algorithms.

We assume that $\mathcal{X}(0)$ is a random vector with a known mean, known covariance matrix and zero marginal skewness in the initialization stage of the procedure. Section 3 describes a procedure that can be employed to generate the sample points $\mathcal{X}^{(i)}(0|0)$ from a prior knowledge about the moments of $\mathcal{X}(0)$. For $k \geq 0$ and whenever the measurement $\mathcal{Y}(k+1)$ becomes available, we present the steps in the computation of sigma points at time t_{k+1}:

$$\begin{aligned}
\mathcal{X}^{(i)}(k+1 \mid k) &= \mathbf{f}\left(\mathcal{X}^{(i)}(k \mid k)\right) + \mathbf{g}\left(\mathcal{X}^{(i)}(k \mid k)\right)\mathcal{W}^{(i)}(k+1), && (4)\\
\hat{\mathcal{V}}^{(i)}_{\mathcal{Y}}(k+1 \mid k) &= \mathcal{Y}(k+1) - \mathbf{h}\left(\mathcal{X}^{(i)}(k+1 \mid k)\right), && (5)\\
\hat{\mathcal{X}}(k+1 \mid k) &= \sum_{i=1}^{2ns+1} p_i \mathcal{X}^{(i)}(k+1 \mid k), && (6)\\
\tilde{\delta}^{(i)}(k+1 \mid k+1) &= \\
\arg \min_{\delta^{(i)}(k+1|k+1)} & \quad \|\hat{\mathcal{V}}^{(i)}_{\mathcal{Y}}(k+1 \mid k) - H^{(i)}(k+1 \mid k)\delta^{(i)}(k+1 \mid k+1)\|_1, \\
& && (7)\\
\mathcal{X}^{(i)}(k+1 \mid k+1) &= \mathcal{X}^{(i)}(k+1 \mid k) + \tilde{\delta}^{(i)}(k+1 \mid k+1), && (8)\\
\hat{\mathcal{X}}(k+1 \mid k+1) &= \sum_{i=1}^{2ns+1} p_i \mathcal{X}^{(i)}(k+1 \mid k+1). && (9)
\end{aligned}$$

The gradient matrix $H^{(i)}(k+1 \mid k)$ for the vector valued function $\mathbf{h}$ at time t_{k+1} is defined by

$$[H^{(i)}(k+1 \mid k)]_{jl} = \left.\frac{\partial \mathbf{h}_j}{\mathcal{X}_l}\right|_{\mathcal{X}^{(i)}(k+1|k)},$$

and $\|\cdot\|_1$ denotes the 1-norm of a vector (which equals the summation of absolute values of all elements).

Implementing the above algorithm yields the sigma points $\mathcal{X}^{(i)}(k+1 \mid k+1)$, $i = 1, 2, \ldots, 2ns+1$, along with the expected values of the predicted and the updated state estimate, i.e., $\hat{\mathcal{X}}(k+1 \mid k)$ and $\hat{\mathcal{X}}(k+1 \mid k+1)$, respectively. Note that the 1-norm minimization in (7) can be achieved by linear programming.

If $\hat{\epsilon}^{(i)}$ is the minimum cost and if $\tilde{\delta}^{(i)}(k+1 \mid k+1)$ are the decision variables which achieve this minimum, it is easy to see that there exist $\mathcal{V}^{(i)}(k+1)$ such that

$$\mathcal{Y}(k+1) = \mathbf{h}(\mathcal{X}^{(i)}(k+1 \mid k)) + H^{(i)}(k+1 \mid k)\delta^{(i)}(k+1 \mid k+1) + \mathcal{V}^{(i)}(k+1)$$

holds and $\|\mathcal{V}^{(i)}(k+1)\|_1 \leq \hat{\epsilon}^{(i)}$. In other words, corresponding to each $\mathcal{X}^{(i)}(k+1 \mid k)$, the procedure finds (vector-valued) measurement noise which causes the smallest error as measured by the 1-norm between the linearized prediction of $\mathbf{h}(\cdot)$ around $\mathcal{X}^{(i)}(k+1 \mid k)$ and the actual observation $\mathcal{Y}(k+1)$.

We re-emphasize that the main idea of this exercise is to preserve some of the nonlinearity in the system dynamics while generating the state estimate and can (possibly) do better than extended Kalman filters *without* having to resort to the computationally expensive sequential Monte Carlo-based estimation. Note that solving a small number of linear programming (LP) based optimization problems with n decision variables will usually be cheaper than doing a Monte Carlo simulation with n correlated random variables. LP problems can be solved extremely efficiently (theoretically, in polynomial time) and several good LP solvers are commercially available; see [23] for more details on linear programming algorithms. The proposed formulation can also be modified easily to account for a situation where there are upper and lower bounds imposed on the unobserved states due to dynamics of the system; see [24] for example. Solving an LP problem with several hundred variables and constraints in a few seconds on an ordinary desktop is a reasonable expectation given today's technological advancement in computing. Furthermore, this procedure eliminates the need of knowing the information about the parametric form of distribution of the measurement noise, which is not always available.

Finally, the special case when $\mathbf{h}$ is linear is worth mentioning. When $\mathbf{h}$ is linear, H is a constant matrix and the measurement equation can be written as

$$\mathcal{Y}(k+1) = H\mathcal{X}(k+1) + \mathcal{V}(k+1).$$

In this case, the 1-norm minimization problem

$$\min_{\mathcal{X}(k+1|k+1)} \|\mathcal{Y}(k+1) - H\mathcal{X}(k+1 \mid k+1)\|_1$$

has a unique solution. One simply needs to solve this single linear programming problem and need not use the samples $\mathcal{X}^{(i)}(k+1 \mid k)$ in computing $\mathcal{X}^{(i)}(k+1 \mid k+1)$. Moreover,

$$\mathcal{X}^{(i)}(k+1 \mid k+1) = \mathcal{X}^{(j)}(k+1 \mid k+1) =: \hat{\mathcal{X}}(k+1 \mid k+1)$$

holds. The sampling procedure is still required for $\mathcal{W}^{(i)}(k+1)$ if the expected value of prediction, $\hat{\mathcal{X}}(k+1 \mid k)$, in (6) needs to be determined. Time series models with nonlinear $\mathbf{f}$ and $\mathbf{g}$ in (1), but a linear $\mathbf{h}$ in (2) commonly occur in econometric models. The most prominent class of models with this structure includes the Cox-Ingersoll-Ross (CIR) model, which is employed to model interest rates. This popular class of models has been widely discussed in the literature; see [25] and [26], among others. The instantaneously compounded interest rate in these type of models is unobservable and has to be inferred from observed interest rates using a nonlinear filter; see [26] for the use of extended Kalman filter in CIR-type interest rate models. Clearly, the algorithm proposed here can provide an intuitively attractive and computationally affordable alternative to EKF, which does not rely on linearization of the state evolution equation.

In the above algorithm, we have assumed that a procedure to generate a set of sigma points $\mathcal{W}^{(i)}(k+1),\ i = 1, 2, \ldots 2ns+1$ with the desired statistical properties is available. The next section outlines such a procedure in generating a symmetric discrete distribution that matches a given mean vector and covariance matrix exactly without requiring an additional optimization. This procedure was first suggested in [22] and has been used in nonlinear filtering context in [20]. A summary of this procedure is provided here for a self-contained presentation of our proposed method in latent state estimation.

3 Generation of sigma points

3.1 Notation

In outlining the sigma point generation algorithm, we shall use the following notation:

n	number of random variables (or dimension of a random vector),
s	number of samples
Φ	target mean vector
R	target covariance matrix

Our aim is to generate samples from a symmetric distribution with a specified mean vector and a specified (positive semi-definite) covariance matrix. Recall that a symmetric matrix R is said to be positive semi-definite if $R \geq 0$, i.e., its eigenvalues are all non-negative. The sigma point generation algorithm given in the next subsection forms a part of the filtering procedure described in section 2, as it is used to generate $\mathcal{G} := \mathcal{W}^{(i)}(k+1),\ i = 1, 2, \ldots, 2ns+1$ which match a given mean vector Φ, a given covariance matrix R and have a symmetric marginal distribution.

3.2 Algorithm for generating sigma points

Following [22], an algorithm adapted for sigma point generation is outlined below.

(i) Decompose a matrix R as $R = LL^\top$ where L is a symmetric positive definite matrix. For a symmetric positive definite R, L is unique and is called the *square root* of the matrix R; see, e.g. [27] and the references therein for the methods of finding L.
(ii) Generate $q_i \in [2ns, \infty], i = 1, 2, \ldots, s$. The q_i's may be generated using any deterministic algorithm or using a random number generator.
(iii) Write $p_i := \frac{1}{q_i}$, $i = 1, 2, \ldots, s$ and $p_{s+1} := 1 - 2n \sum_{i=1}^{s} p_i$.

(iv) Define a multivariate discrete distribution $\mathcal{G}$ over a support of $2ns+1$ points as follows:

$$\begin{aligned}&\mathbb{P}\left(\mathcal{G}=\Phi+\frac{1}{\sqrt{2sp_i}}L_j\right)=\mathbb{P}\left(\mathcal{G}=\Phi-\frac{1}{\sqrt{2sp_i}}L_j\right)=p_i,\\ &j=1,2,\ldots,n,\ i=1,2,\ldots,s,\\ &\mathbb{P}\left(\mathcal{G}=\Phi\right)=p_{s+1}.\end{aligned}\tag{10}$$

where L_j denotes the j^{th} column of a matrix L.

The steps (i)-(iv) constitute the procedure to generate sigma points having Φ as the mean vector, R as the covariance matrix and zero marginal skewness. In sequential state estimation, step (i) is not necessary to be repeated when the covariance matrix has to remain the same throughout multiple time-steps. In various practical applications, the noise covariance matrices are usually assumed to be constant. Hence matrix factorization is not needed at each time-step in the filtering process.

The distributional properties of these samples are summarized in the following result:

Lemma 1 *(1) For p_i defined as above, $p_i \geq 0$, $i=1,2,\ldots,s$ and $2n\sum_{i=1}^{s}p_i+p_{s+1}=1$.*

(2) For $\mathcal{G}$ defined as above,

$$\mathbb{E}[\mathcal{G}]=\Phi,\tag{11}$$

$$\mathbb{E}\left[(\mathcal{G}-\Phi)(\mathcal{G}-\Phi)^\top\right]=R,\tag{12}$$

$$\mathbb{E}\left[(\mathcal{G}_i-\Phi_i)^3\right]=0.\tag{13}$$

Proof: See [20]. ∎

From Lemma 1, note that the exact values of the weights p_i have no impact on matching of moments Φ and R, so long as they form a valid probability measure. In situations where $\mathcal{G}(k)$ itself represents a discrete time stochastic process, we could choose *random* probability weights $\{p_i\}$ at each time k and generate a different realization of $\mathcal{G}(k)$ at each time k. Of course, we may choose to use deterministic p_i's instead if desired.

4 Numerical example

We consider an Euler-discretized version of a 2-factor, square root affine interest rate model as described in [26] to demonstrate the implementation of the new filtering method. This model is a generalization of the CIR model first proposed in [25]. In this model, the two unobservable states $\mathcal{X}_1(k)$ and $\mathcal{X}_2(k)$ are assumed to evolve according to the following equations:

$$\begin{aligned}\mathcal{X}_j(k+1) = &\kappa_j\theta_j\Delta + (1-\kappa_j\Delta)\mathcal{X}_j(k)\\ &+\sigma_j\sqrt{\mathcal{X}_j(k)\Delta}\,\mathcal{W}_j(k+1),\ j=1,\ 2,\end{aligned} \tag{14}$$

where $\mathcal{W}_1(k)$ and $\mathcal{W}_2(k)$ are independent standard normal random variables at each time t_k. The time period between two successive samples is assumed to be $\Delta := t_k - t_{k-1} = 1/250$. The measurable functions of these states, $\mathcal{Y}_i(k)$, are given by

$$\mathcal{Y}_i(k) = \prod_{j=1}^{2} A_{i,j}\exp\left(-\sum_{j=1}^{2} B_{i,j}\mathcal{X}_j(k)\right) + \mathcal{V}_i(k), \tag{15}$$

$$\text{where}\quad A_{i,j} = \left(\frac{2\phi_{j,1}\exp(\phi_{j,2}T_i/2)}{\phi_{j,4}}\right)^{\phi_{j,3}},\quad B_{i,j} = \frac{2\left(\exp(\phi_{j,1}T_i)-1\right)}{\phi_{j,4}}$$

and $\phi_{j,1} = \sqrt{(\kappa_j+\lambda_j)^2+2\sigma_j^2}$, $\phi_{j,2} = \kappa_j+\lambda_j+\phi_{j,1}$, $\phi_{j,3} = 2\kappa_j\theta_j/\sigma_j^2$, $\phi_{j,4} = 2\phi_{j,1}+\phi_{j,2}(\exp(\phi_{j,1}T_i)-1)$. In these equations, $\kappa_j, \theta_j, \sigma_j$ and λ_j are constants. Here, T_i is a non-negative number which, in practice, represents the time to maturity of a pure discount bond and $\mathcal{Y}_i(k)$ is the corresponding price of the bond at time t_k. Note that each T_i only appears in the measurement equation for $\mathcal{Y}_i(k)$. We assume that $\mathcal{Y}_1(k)$, $\mathcal{Y}_2(k)$, etc are observed in noise $\mathcal{V}_i(k)$ which is bounded and have a mean of zero.

Remark 2 *One may use $-\log(\mathcal{Y}_i(k))$ as a measurement, which yields a linear measurement equation in $\mathcal{X}_j(k)$. We shall use $\mathcal{Y}_i(k)$ as a measurement to illustrate the performance of the proposed filter wherein the state space system involves a nonlinear unobservable dynamics as well as a nonlinear measurement equation.*

The parameters used for this model are the same as those used in the numerical demonstration in [26] and are presented in Table 1.

Table 1. Parameters in the implementation of the system specified in (14)-(15).

κ_1	0.0718	σ_1	0.2160
κ_2	0.7830	σ_2	1.2200
θ_1	4.3000	λ_1	-0.2130
θ_2	1.6400	λ_2	-0.9140

We will use $T_1 = 0.5$, $T_2 = 1$, $T_3 = 2$ and use the corresponding $\mathcal{Y}_1(k)$, $\mathcal{Y}_2(k)$ and $\mathcal{Y}_3(k)$ as the observations at each time t_k. This gives a two-state, three-measurement state space system. Based on a simulated observation sample path, we wish to see whether we can predict $\mathcal{Y}_i(k+1 \mid k)$ at each t_k accurately, where

$$\mathcal{Y}_i(k+1 \mid k) = \sum_{l=1}^{4s+1} \nu_l \prod_{j=1}^{2} A_{i,j} exp \left(- \sum_{j=1}^{2} B_{i,j} \mathcal{X}_j^{(l)}(k+1 \mid k) \right).$$

Here, $4s+1$ is the total number of sigma points for $\mathcal{W}^{(i)}(k+1)$ (since $n = 2$) and ν_l's are the corresponding $4s+1$ probability weights. We would like to compare the predictive ability of the PLSPF proposed here to that of the EKF. At time t_{k+1}, EKF uses linearized versions of equations (14) -(15) around the updated state estimate $\hat{\mathcal{X}}(k \mid k)$ at time t_k and then uses the standard Kalman filter for state prediction and update. The formulae for the EKF based on (3) are not repeated here; the reader is referred to standard textbooks such as [4]. Alternatively, [20] provides the formulae with a notation similar to the one used in this paper.

To measure the performance of a filter, we consider the average of root mean squared error (AvRMSE) as well as the average of mean relative absolute error (AvMRAE) in one step ahead predictions. The root mean squared error (RMSE) for a measurement $\mathcal{Y}_j$ and for a particular sample path i is given by

$$RMSE_{(i,j)} = \sqrt{\frac{1}{M} \sum_{k=1}^{M} \left((\mathcal{Y}_j(k+1))_i - (\mathcal{Y}_j(k+1 \mid k))_i \right)^2},$$

where M is the time horizon. Here $(\mathcal{Y}_j(k+1))_i$ (respectively, $(\mathcal{Y}_j(k+1 \mid k))_i$) denotes the noisy observation of $\mathcal{Y}_j(k+1)$ (respectively, the prediction of $\mathcal{Y}_j(k+1 \mid k)$) for the i^{th} sample path. AvRMSE$_j$ is computed as the sample mean of RMSE$_{(i,j)}$ over different sample paths i,

$$\text{AvRMSE}_j = \frac{1}{N} \sum_{i=1}^{N} \text{RMSE}_{(i,j)}, \; j = 1, 2, 3.$$

In a similar fashion, MRAE for measurement $\mathcal{Y}_j$ and sample path i is defined by

$$\text{MRAE}_{(i,j)} = \frac{1}{M}\sum_{k=1}^{M} \frac{|(\mathcal{Y}_j(k+1))_i - (\mathcal{Y}_j(k+1 \mid k))_i|}{(\mathcal{Y}_j(k+1))_i}$$

and AvMRAE$_j$ is computed as the sample mean of MRAE$_{(i,j)}$ over different sample paths i, i.e.,

$$\text{AvMRAE}_j = \frac{1}{N}\sum_{i=1}^{N} \text{AvMRAE}_{(i,j)},\ j = 1, 2, 3.$$

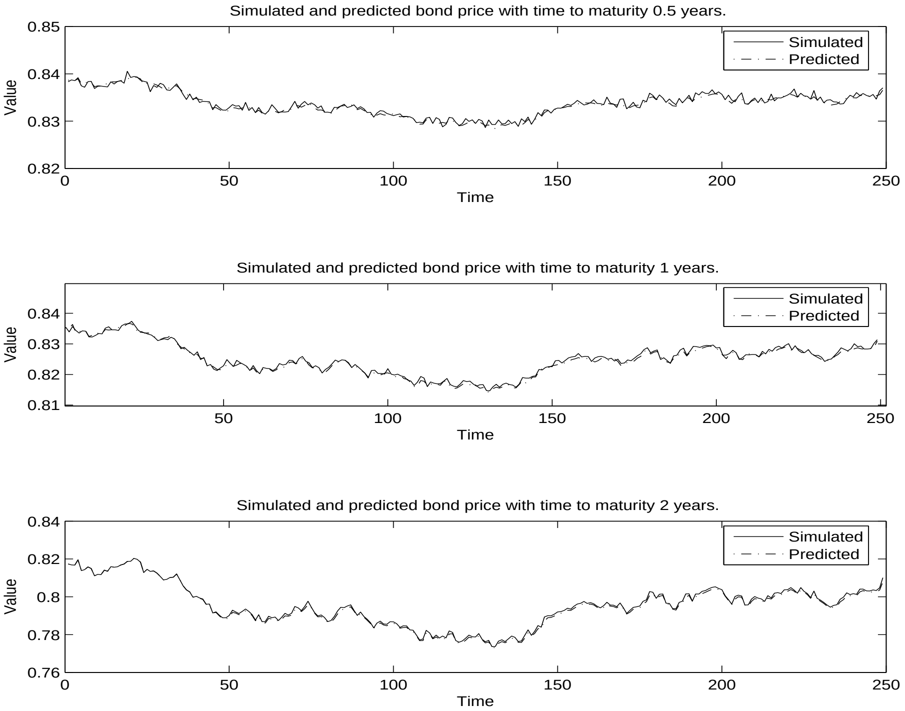
Both these functions of prediction error, AvRMSE$_j$ and AvMRAE$_j$, are computed over N=100 sample paths, with each path consisting of M=250 time-steps, for each of the three measurements $\mathcal{Y}_1(k)$, $\mathcal{Y}_2(k)$ and $\mathcal{Y}_3(k)$. At each time-step only 13 samples or sigma points are generated, which corresponds to choosing $s = 3$ for the algorithm in section 3.2. The results of this error analysis for PLSPF are reported in Table 2. Figure 1 on the other hand displays a graphical comparison between the simulated $\mathcal{Y}_j(k+1)$ (solid line) and the predicted $\mathcal{Y}_j(k+1 \mid k)$ (dashed line) for one particular sample path. The mean computation time per sample path for PLSPF was 65.88 seconds, with the maximum time per sample path being 71.03 seconds. In other words, the performance with PLSPF is achieved at the cost of only around 0.27 seconds per time-step. The experiments were also repeated for four measurements and three states and the mean computation time per sample path in this case was 117.15 seconds (detailed results in this case are omitted for brevity). This computation was carried out on a desktop with Pentium IV core duo processor (2.4 Ghz), running MATLAB version R2007b on Windows XP. The computation time can easily be improved by employing a purpose-written optimization code or a higher specification machine. Clearly, this computation time is affordable even for real time processing involving applications where the estimation of state dynamics is sufficiently slow, such as on-line estimation problems in many chemical processes.

Table 2 : Average errors in predicting $\mathcal{Y}_j(k+1)$ with PLSPF (average over 100 sample paths, with 250 time-steps in each sample path).

	$j = 1$	$j = 2$	$j = 3$
AvMRAE$_j$	0.000498	0.000589	0.000795
AvRMSE$_j$	0.000525	0.000616	0.000764

The state estimation results with EKF in the present example were significantly worse, with the filter diverging in 60 out of 100 sample paths and yielding extremely large errors. The average errors over the remaining 40 sample paths were still high, even with the lowest average error being over ten times

Fig. 1. Prediction for $\mathcal{Y}_1(k)$, $\mathcal{Y}_2(k)$ and $\mathcal{Y}_3(k)$ using PLSPF



the corresponding error with the PLSPF, as can be seen from Table 3.

Table 3 : Average errors in predicting $\mathcal{Y}_j(k+1)$ with EKF (average over 40 sample paths on which the filter did not diverge, with 250 time-steps in each sample path).

	$j=1$	$j=2$	$j=3$
AvMRAE$_j$	0.006915	0.014729	0.034515
AvRMSE$_j$	0.013974	0.030556	0.074106

These numerical experiments clearly indicate the superiority of the proposed algorithm over the EKF for nonlinear systems of the form (1), in the case when the measurement equation is sufficiently smooth.

5 Concluding remarks

In this article, we put forward a new filtering heuristic for nonlinear and non-Gaussian systems, which we refer to as partially linearized sigma point filter (PLSPF). This algorithm shares some of the advantages of the modified sigma point filter proposed in [20], in the sense that the state covariance matrix need not be factorized at each step and the first three moments are exactly matched during sigma point generation. However, unlike conventional sigma point filters, the state update step in PLSPF does not use closed-form formula based on the Gaussianity assumption. Instead, a simple and intuitively appealing optimization is utilized where the measurement equation is linearized and the updated state which best matches the given observations in an appropriate deterministic sense is found. We demonstrated the implementation of the algorithm through a detailed numerical example involving a nonlinear, multivariate time series. The proposed method is a computationally simpler and attractive alternative to particle filtering for nonlinear time series in engineering as well as in econometric modeling applications in decision sciences. Further, it also provides a very useful alternative to traditional sigma point filters in engineering and ensemble filters in geosciences.

References

[1] R.E. Kalman. "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[2] H. Kushner, *Numerical Methods for Stochastic Control Problems in Continuous Time.* Springer Verlag, 2000.

[3] A. Jazwinski, *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

[4] B. Anderson and J. Moore, *Optimal Filtering.* Prentice-Hall, 1979.

[5] J.C. Jimenez and T. Ozaki, "Local linearization filters for nonlinear continuous-discrete state space models with multiplicative noise", *International Journal of Control*, vol. 76, pp. 1159–1170, 2003.

[6] N. Saha and D. Roy, "Extended Kalman filters using explicit and derivative-free local linearizations", *Applied Mathematical Modelling*, vol. 33, pp. 2545–2563, 2009.

[7] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of American Statistical Association*, vol. 93, pp. 1032–1044, 1998.

[8] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[9] H. Tanizaki and R. Mariano, "Prediction, filtering and smoothing in nonlinear and nonnormal cases using Monte Carlo integration," *Journal of Applied Econometrics*, vol. 9, pp. 163–179, 1996.

[10] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, 2004.

[11] A. Farina, B. Ristic, and D. Benvenuti, “Tracking a ballistic target: comparison of several nonlinear filters,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, pp. 854–867, 2002.

[12] G. Wang, “On the latent state estimation of nonlinear population dynamics using Bayesian and non-Bayesian state-space models,” *Ecological Modelling*, vol. 200, pp. 521–528, 2007.

[13] Y. Dong, Y. Li, M. Xiao and M. Lai, “Unscented Kalman filter for time varying spectral analysis of earthquake ground motions”, *Applied Mathematical Modelling*, vol. 33, pp. 398-412, 2009.

[14] G. Plett, “Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1: Introduction and state estimation,” *Journal of Power Sources*, vol. 161, pp. 1356–1368, 2006.

[15] A. Hermoso-Carazo and J. Linares-Perez, “Different approaches for state filtering in nonlinear systems with uncertain observations,” *Applied Mathematics and Computation*, vol. 187, pp. 708–724, 2007.

[16] G. Evensen, “Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics,” *Journal of Geophysical Research*, vol. 10, pp. 143–162, 1994.

[17] H. Mitchell and P. Houtekamer, “Data assimilation using an ensemble Kalman filter technique,” *Monthly Weather Review*, vol. 126, pp. 796–811, 1998.

[18] G. Evensen, “The ensemble Kalman filter: theoretical formulation and practical implementation”, *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[19] O. Barrero Mendoza, B. De Moor and D.S. Bernstein, “Data assimilation for magnetohydrodynamics systems”, *Journal of Computational and Applied Mathematics*, vol. 189, pp. 242–259, 2006.

[20] P. Date, L. Jalen and R. Mamon, “A new algorithm for latent state estimation in nonlinear time series models,” *Applied Mathematics and Computation*, vol. 203, pp. 224–232, 2008.

[21] G. Grimmett and D. Stirzaker, *Probability and Random Processes.* Oxford University Press, 2001.

[22] P. Date, R. Mamon and L. Jalen, “A new moment matching algorithm for sampling from partially specified symmetric distributions,” *Operations Research Letters*, vol. 36, pp. 669–672, 2008.

[23] R. Saigal, *Linear Programming: A Modern Integrated Analysis.* Kluwer Academic Publishers, 1995.

[24] S. Kolas, B.A. Foss and T.S. Schei, “Constrained nonlinear state estimation based on the UKF approach”, *Computers and Chemical Engineering*, vol. 33, pp. 1386–1401, 2009.

[25] J. Cox, J. Ingersoll, and S. Ross, “A theory of the term structure of interest rates,” *Econometrica*, vol. 53, pp. 385–408, 1985.

[26] A. L. J. Geyer and S. Pichler, “State-space approach to estimate and test multifactor Cox-Ingersoll-Ross models of the term structure,” *Journal of Financial Research*, vol. 22, pp. 107-130, 1999.

[27] R. Horn and C. Johnson, *Matrix Analysis.* Cambridge University Press, 1999.