

# A new algorithm for latent state estimation in nonlinear time series models

Paresh Date<sup>a</sup>   Luka Jalen<sup>a</sup>   Rogemar Mamon<sup>b</sup>

<sup>a</sup>*Center for the Analysis of Risk and Optimization Modelling Applications,  
Department of Mathematical Sciences, Brunel University, Middlesex UB8 3PH  
United Kingdom*

<sup>b</sup>*Department of Statistical and Actuarial Sciences, The University of Western  
Ontario, London, Ontario, Canada N6A 5B7*

---

## Abstract

We consider the problem of optimal state estimation for a wide class of nonlinear time series models. A modified sigma point filter is proposed, which uses a new procedure for generating sigma points. Unlike the existing sigma point generation methodologies in engineering where negative probability weights may occur, we develop an algorithm capable of generating sample points that always form a valid probability distribution while still allowing the user to sample using a random number generator. The effectiveness of the new filtering procedure is assessed through simulation examples.

---

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

## 1 Introduction

We consider the problem of latent state estimation in a discrete, nonlinear time series. The general class of the systems considered have the following state space form:

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

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

where  $\mathcal{X}(k)$  is the state vector at time  $t_k$ ,  $\mathcal{Y}(k)$  is the measurement vector at time  $t_k$ ,  $\mathbf{f}$ ,  $\mathbf{g}$ ,  $\mathbf{h}$  are given nonlinear (vector-valued) functions and  $\mathcal{V}(k)$ ,  $\mathcal{W}(k)$  are symmetric vector-valued random variables with bounded mean, variance and marginal kurtosis. We assume that  $t_k - t_{k-1}$  is constant for all  $k$ . At

each time  $t_k$ , the noisy measurement vector  $\mathcal{Y}(k)$  is assumed to be available and an estimate of the random vector  $\mathcal{X}(k)$  based on information up to (and including) time  $t_k$  is desired.

The state estimation problems for these nonlinear models are practically important and occur in a wide spectrum of research areas such as radar navigation, climatology, geosciences and financial modeling, among others. These problems can be numerically quite challenging since the optimal recursive solution to the state estimation problem requires the propagation of full probability density; see, *e.g.* [1] for an approximate solution to a more general nonlinear filtering problem. In the special case of linear Gaussian state space models, a closed-form expression exists for the conditional state density and is given by the linear Kalman filter. In practice, the current approaches addressing the nonlinear filtering type problems make use of one of the following ways of approximation:

- One may use the extended Kalman filter (EKF), which utilizes local linearization of equation (1). This leads to the derivation of a linear state space system and then a Kalman filter is employed to derive the conditional state density of  $\mathcal{X}(k)$ . This approach has been used in engineering for more than three decades [2] and has been extensively discussed in standard textbook such as [3]. EKF works well if the system is, indeed, approximately linear. This assumption is often extremely difficult to verify. [4] discusses a successful implementation of EKF for a nonlinear interest rate model.
- Another approach for nonlinear filtering is sequential Monte Carlo filtering (also called *particle filtering*), where the required density functions are represented by a set of random samples (or *particles*) with discrete probability weights and these samples are then used to compute the necessary conditional moment estimates. As the number of samples becomes large, the estimate approaches the optimal Bayesian estimate under fairly general conditions; see [5], [6], [7] and the references therein. While this method can perform significantly better than EKF for highly nonlinear systems, it is computationally quite expensive since a large number of samples need to be generated at each time  $t_k$ . Some computational saving is possible if the system contains a linear substructure which can be dealt with linear Kalman updates. These *marginalized* filters have found some applications in engineering; see [8] and the references therein.
- A modification of EKF in terms of *unscented filter* or *sigma point filter* has become popular in the recent years. In [9], a survey of several applications of sigma point filters in engineering is provided, specifically in communication, tracking and navigation (also see [10]). Other reported applications of this filtering technique include modeling of population dynamics [11] and state estimation in electrochemical cells for battery management [12]. Approximate methods to deal with multiplicative uncertainty in the observation equation under sigma point filtering framework are discussed in [13]. This

type of filters may be seen as a compromise between an EKF and a particle filter.

Similar to the propagation equations in EKF, 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. However, the system equations are *not* linearized in this case. Instead, a small set of sample points (or *sigma points*) is generated and propagated through the nonlinear transformation to compute the conditional moment estimates. Instead 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 points which are chosen such that some of the moment properties of the *a priori* distribution are matched exactly. The main problem with these type of filters is that the sample points do not necessarily define a valid distribution since the weights corresponding to probability masses are not guaranteed to be non-negative. Further, the algorithms for generating samples are purely deterministic and do not allow for a source of randomness in the filtering procedure.

A sigma point filter requires computing square root of the state covariance matrix at each time-step. This may not be computationally feasible if the number of states is very large, which is the case for most problems in geosciences. A variant of sigma point filter, usually called the *ensemble filter*, is used in geosciences where the state is not sampled at all and only the noise distributions are sampled using traditional Monte Carlo sampling techniques <sup>1</sup>. This technique was introduced in [14] and has also been employed in [15]. The method we propose is closer in spirit to ensemble filters. We will discuss the similarities and differences between the two filtering methods, i.e., ensemble filter vis-a-vis our proposed new method later in section 5.

The purpose of this paper is to propose a new filtering algorithm for state estimation in nonlinear time series which addresses the above mentioned deficiencies of sigma point filters and to assess the performance of this algorithm through numerical examples. The sigma point generation step in this algorithm is adapted from a recently proposed method for generating samples from a discrete distribution with specified moment properties [16]. The rest of the paper is organized as follows. The next section outlines the recursive equations for linear Kalman filter, which are then used in the development of subsequent sections. Section 3 outlines the use of the proposed sigma point filter, while section 4 outlines the underlying algorithm for sigma point generation. The operation of the algorithm is demonstrated through two examples in section 6. Finally, section 7 concludes and outlines the directions for future research.

---

<sup>1</sup> The first author is grateful to Prof. Sachin Patwardhan from Indian Institute of Technology, Mumbai, India for pointing out this technique.

## 2 Linear Kalman filter

For a linear state space system of the form

$$\mathcal{X}(k+1) = A\mathcal{X}(k) + B + U_w\mathcal{W}(k+1), \quad (3)$$

$$\mathcal{Y}(k) = C\mathcal{X}(k) + D + U_v\mathcal{V}(k), \quad (4)$$

where  $A, B, C, D, U_v$  and  $U_w$  are constant matrices, assume that the conditional expectation  $\hat{\mathcal{X}}(k|k)$  and its covariance matrix  $P_{xx}(k|k)$  at time  $t_k$  (derived after measuring  $\mathcal{Y}(k)$ ) are known. The Kalman filtering algorithm for finding conditional moments at the next time  $t_{k+1}$  proceeds as follows.

$$\hat{\mathcal{X}}(k+1|k) = A\hat{\mathcal{X}}(k|k) + B, \quad (5)$$

$$P_{xx}(k+1|k) = AP_{xx}(k|k)A^\top + U_wU_w^\top, \quad (6)$$

$$\hat{\mathcal{V}}(k+1) = \mathcal{Y}(k+1) - C\hat{\mathcal{X}}(k+1|k) - D, \quad (7)$$

$$P_{xv}(k+1|k) = AP_{xx}(k+1|k)C^\top, \quad (8)$$

$$P_{vv}(k+1|k) = CP_{xx}(k+1|k)C^\top + U_vU_v^\top, \quad (9)$$

$$\hat{\mathcal{X}}(k+1|k+1) = \hat{\mathcal{X}}(k+1|k) + P_{xv}(k+1|k)P_{vv}^{-1}(k+1|k)\hat{\mathcal{V}}(k+1), \quad (10)$$

$$P_{xx}(k+1|k+1) = P_{xx}(k+1|k) - P_{xv}(k+1|k)P_{vv}(k+1|k)^{-1}P_{xv}(k+1|k)^\top, \quad (11)$$

where  $\hat{\mathcal{X}}(k+1|k)$  denotes the optimal estimate of  $\mathcal{X}$  at time  $k+1$  given the measurements and other available values up to time  $k$ .  $P^\top$  denotes the transpose of matrix  $P$ . To initialise the procedure, it is assumed that  $\hat{\mathcal{X}}(0|0)$  and  $P_{xx}(0|0)$  are known.

Equation (10) is an optimal linear filter, in the sense that it yields the minimum variance over all linear filters, even when  $\mathcal{V}(k)$ ,  $\mathcal{W}(k)$  are not Gaussian. When  $\mathcal{V}(k)$ ,  $\mathcal{W}(k)$  are Gaussian,  $\hat{\mathcal{X}}(k+1|k)$  is the conditional mean estimator for  $\mathcal{X}(k+1)$ , given  $\mathcal{Y}(k)$ . In fact, equation (10) may be derived using a standard conditional mean relationship for two Gaussian variables  $\mathcal{X}, \mathcal{Y}$  [17]:

$$\mathbb{E}(\mathcal{X}|\mathcal{Y}) = \mathbb{E}(\mathcal{X}) + \Sigma_{XY}\Sigma_{YY}^{-1}(\mathcal{Y} - \mathbb{E}(\mathcal{Y})) \quad (12)$$

where  $\Sigma_{YY}$  and  $\Sigma_{XY}$  are covariance matrices.

The main idea of sigma point filters as well as ensemble filters is to derive approximations to the quantities on the right hand side of (10) through sampling the distributions of  $\mathcal{V}(k)$  and  $\mathcal{W}(k)$  and then use the same, closed-form update formula (10), which is known to be optimal for the linear Gaussian case. In sigma point filters, certain moment properties of the prior distribution are matched exactly using deterministic sigma point generation. In ensemble filters, pseudo-random number generators are used to sample the known distributions of the noise terms.

The next section details a new sigma point filtering algorithm. At the moment, we set aside the question of which statistical properties to match and assume that a method for generating samples matching appropriate statistical properties is available. We shall consider the problem of generating samples in section 4.

### 3 A sigma point filter

At time  $t_{k+1}$ , assume that sample points (or *sigma points*)

$$\left[ \mathcal{W}^{(i)}(k+1)^\top \mathcal{V}^{(i)}(k+1)^\top \right]^\top, \quad i = 1, 2, \dots, 2ns + 1$$

are available for the discrete time state space system (1)-(2), along with the associated joint probability weights  $p_i$ ,  $i = 1, 2, \dots, s$ . Here,  $n$  is the dimension of the composite vector  $\left[ \mathcal{W}^{(i)}(k+1)^\top \mathcal{V}^{(i)}(k+1)^\top \right]^\top$ . As will be seen in the next section, some of the probability weights are common to two or more support points and the set of  $s$  probability weights determine the  $2ns + 1$  support points above. Further, the sample points of the updated state estimate  $\mathcal{X}^{(i)}(k|k)$  are available.

**Remark:** Note that  $\mathcal{X}^{(i)}(k|k)$  is not sampled and the joint probability  $p_i$  for  $\left[ \mathcal{W}^{(i)}(k+1)^\top \mathcal{V}^{(i)}(k+1)^\top \right]^\top$  at each  $i$  is effectively assigned as the joint probability of occurrence of  $\left[ \mathcal{W}^{(i)}(k+1)^\top \mathcal{V}^{(i)}(k+1)^\top \mathcal{X}^{(i)}(k|k)^\top \right]^\top$ . In this respect the procedure is similar to an ensemble filter.

To initialize the procedure, we assume that  $\mathcal{X}(0)$  is a random vector with a known mean, known covariance matrix and zero marginal skewness. The sample points  $\mathcal{X}^{(i)}(0|0)$  can be generated from this prior knowledge about the moments of  $\mathcal{X}(0)$  using the procedure outlined in section 4. For  $k \geq 0$ , The steps involved in the computation of sigma points at time  $t_{k+1}$  once the

measurement  $\mathcal{Y}(k+1)$  becomes available are as follows:

$$\mathcal{X}^{(i)}(k+1|k) = \mathbf{f}(\mathcal{X}^{(i)}(k|k)) + \mathbf{g}(\mathcal{X}^{(i)}(k|k)) \mathcal{W}^{(i)}(k+1), \quad (13)$$

$$\mathcal{Z}^{(i)}(k+1|k) = \mathbf{h}(\mathcal{X}^{(i)}(k+1|k)) + \mathcal{V}^{(i)}(k+1), \quad (14)$$

$$\hat{\mathcal{V}}_{\mathcal{Y}}^{(i)}(k+1|k) = \mathcal{Z}^{(i)}(k+1|k) - \mathcal{Y}(k+1), \quad (15)$$

$$\hat{\mathcal{X}}(k+1|k) = \sum_{i=1}^{2ns+1} p_i \mathcal{X}^{(i)}(k+1|k), \quad (16)$$

$$\hat{\mathcal{V}}_{\mathcal{X}}^{(i)}(k+1|k) = \mathcal{X}^{(i)}(k+1|k) - \hat{\mathcal{X}}(k+1|k), \quad (17)$$

$$P_{xx}(k+1|k) = \sum_{i=1}^{2ns+1} p_i (\hat{\mathcal{V}}_{\mathcal{X}}^{(i)}(k+1|k)) (\hat{\mathcal{V}}_{\mathcal{X}}^{(i)}(k+1|k))^{\top}, \quad (18)$$

$$P_{xv}(k+1|k) = \sum_{i=1}^{2ns+1} p_i (\hat{\mathcal{V}}_{\mathcal{X}}^{(i)}(k+1|k)) (\hat{\mathcal{V}}_{\mathcal{Y}}^{(i)}(k+1|k))^{\top}, \quad (19)$$

$$P_{vv}(k+1|k) = \sum_{i=1}^{2ns+1} p_i (\hat{\mathcal{V}}_{\mathcal{Y}}^{(i)}(k+1|k)) (\hat{\mathcal{V}}_{\mathcal{Y}}^{(i)}(k+1|k))^{\top}, \quad (20)$$

$$\begin{aligned} \mathcal{X}^{(i)}(k+1|k+1) &= \hat{\mathcal{X}}(k+1|k) \\ &+ P_{xv}(k+1|k) P_{vv}^{-1}(k+1|k) \hat{\mathcal{V}}_{\mathcal{Y}}^{(i)}(k+1|k). \end{aligned} \quad (21)$$

Note the similarity between equations (10) and (21). Implementing the above algorithm yields the sigma points  $\mathcal{X}^{(i)}(k+1|k+1)$ ,  $i = 1, 2, \dots, 2ns+1$ . Note that the heuristics we have used to generate the samples for the measurement innovations  $\hat{\mathcal{V}}_{\mathcal{Y}}^{(i)}(k+1|k)$  is different from the more common approach in the literature on sigma point filtering, which replaces (15) and (21) by

$$\begin{aligned} \hat{\mathcal{V}}_{\mathcal{Y}}^{(i)}(k+1|k) &= \mathcal{Z}^{(i)}(k+1|k) - \hat{\mathcal{Z}}(k+1|k) \quad \text{and} \\ \mathcal{X}^{(i)}(k+1|k+1) &= \hat{\mathcal{X}}(k+1|k) \\ &+ P_{xv}(k+1|k) P_{vv}^{-1}(k+1|k) (\mathcal{Z}^{(i)}(k+1|k) - \mathcal{Y}(k+1)) \end{aligned}$$

respectively, with  $\hat{\mathcal{Z}}(k+1|k) = \sum p_i \mathcal{Z}^{(i)}(k+1|k)$ . However, we found that using the heuristics (15) and (21) in place of the above equations improves the state estimation performance significantly. Intuitively, our choice may be justified by the fact that  $\hat{\mathcal{Z}}(k+1|k)$  is simply an estimate of  $\mathcal{Y}(k+1)$  and it makes sense to use the actual measurement value, when it is available, than using its estimate.

To reiterate the point of this exercise, we can preserve the nonlinearity in the system dynamics while generating the state estimate and can do better than linear filters *without* having to resort to the computationally expensive sequential Monte Carlo-based estimation. In particular, instead of asymptotically generating entire distributions which requires a large number of samples, we use only a small number of samples but reproduce some statistical properties exactly.

In the above algorithm, we have assumed that a procedure to generate a set of sigma points  $\left[ \mathcal{W}^{(i)}(k+1)^\top \mathcal{V}^{(i)}(k+1)^\top \right]^\top$ ,  $i = 1, 2, \dots, 2ns + 1$  with the desired statistical properties is available. The next section outlines such a procedure to generate a symmetric discrete distribution to match a given mean vector and covariance matrix exactly and also match the sum of marginal kurtosis exactly in some cases, without requiring an additional optimization.

## 4 Generation of sigma points

### 4.1 Notation

We first outline here the notation used in our development of the sigma point generation algorithm.

$n$	number of random variables (or dimension of a random vector),
$s$	number of samples,
$\Phi$	target mean vector,
$R$	target covariance matrix,
$\kappa_i$	target marginal 4 <sup>th</sup> central moment for the $i^{\text{th}}$ random variable,
$L_{ij}$	entry in the $i^{\text{th}}$ row and $j^{\text{th}}$ column of a matrix $L$ ,

For a symmetric matrix  $R$ ,  $R \geq 0$  indicates that the matrix is positive semi-definite, i.e. has all non-negative eigenvalues.

We aim to generate samples from a symmetric distribution with a specified mean vector and a specified (positive definite) covariance matrix. In the case of the sigma point filtering algorithm described in the last section, the purpose is to generate  $\mathcal{G} := \left[ \mathcal{W}^{(i)}(k+1)^\top \mathcal{V}^{(i)}(k+1)^\top \right]^\top$ ,  $i = 1, 2, \dots, 2ns + 1$  which match a given mean vector  $\Phi$ , a given covariance matrix  $R$  and have a symmetric marginal distribution.

### 4.2 Algorithm for generating sigma points

The algorithm described below is adapted from the scenario generation algorithm in [16].

- (i) Find a symmetric positive definite matrix  $L$  such that  $R = LL^\top$ . For a symmetric positive definite  $R$ ,  $L$  is unique. If  $R$  has distinct eigenvalues, this may be found using singular value decomposition; see, e.g. [18] and the references therein for the methods in finding  $L$ . This matrix  $L$  is usually referred to as the *square root* of the matrix  $R$ .
- (ii) If  $\frac{\sum_{i=1}^n \kappa_i}{\sum_{i,j=1}^n (L_{ij}^4)} > n$ , generate  $s-1$  numbers  $q_i \in [2sn, \psi]$ ,  $i = 1, 2, \dots, s-1$ , where the constant  $\psi$  is given by

$$\psi = \frac{2s^2 \sum_{i=1}^n \kappa_i}{(s-1) \sum_{i,j=1}^n (L_{ij}^4)} - \frac{2ns}{s-1},$$

and set  $q_s = \frac{2s^2 \sum_{i=1}^n \kappa_i}{\sum_{i,j=1}^n (L_{ij}^4)} - \sum_{i=1}^{s-1} q_i$ . It can be easily shown that if  $\frac{\sum_{i=1}^n \kappa_i}{\sum_{i,j=1}^n (L_{ij}^4)} > n$ , it is always possible to choose  $s$  such that  $\psi > 2ns$ . Further, due to the definition of the upper bound on  $q_i$ , it can be shown that  $q_s > 2ns$  holds. For a scalar random variable, the constraint  $\frac{\kappa_1}{(L_{11}^4)} > 1$  implies that the kurtosis is greater than unity, which is always true for elliptic distributions [19].

If the condition above is not satisfied (i.e. if  $\psi \leq 2ns$ ), generate  $s$  numbers  $q_i \in [2sn, \infty]$ ,  $i = 1, 2, \dots, s-1$ . Given its lower bound and (possibly) upper bound in either case,  $q_i$  may be generated using any deterministic algorithm or using a random number generator.

- (iii) Set  $p_i = \frac{1}{q_i}$ ,  $i = 1, 2, \dots, s$  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:

$$\begin{aligned} \mathbb{P}\left(\mathcal{G} = \Phi + \frac{1}{\sqrt{2sp_i}} L_j\right) &= \mathbb{P}\left(\mathcal{G} = \Phi - \frac{1}{\sqrt{2sp_i}} L_j\right) = p_i, \\ j &= 1, 2, \dots, n, i = 1, 2, \dots, s, \\ \mathbb{P}(\mathcal{G} = \Phi) &= p_{s+1}. \end{aligned} \tag{22}$$

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

Steps (i)-(iv) constitute the entire set of procedures needed to construct the required samples. Step (i) need not be repeated in a sequential procedure, if the covariance matrix is to remain the same through multiple time-steps. This is practically important, since noise covariance matrices are usually assumed to be constant and they need not be factorized at each time-step during filtering.

The following result collects together the distributional properties of these samples:

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



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

$$\mathbb{E}[\mathcal{G}] = \Phi, \quad (23)$$

$$\mathbb{E}[(\mathcal{G} - \Phi)(\mathcal{G} - \Phi)^\top] = R, \quad (24)$$

$$\mathbb{E}[(\mathcal{G}_i - \Phi_i)^3] = 0. \quad (25)$$

Further, if  $\psi > 2ns$  holds,

$$\sum_{i=1}^n |\kappa_i - \mathbb{E}(\mathcal{G}_i - \Phi_i)^4| = \sum_{i=1}^n \kappa_i. \quad (26)$$

**Proof:** Since  $q_i \geq 2ns$ ,  $i = 1, 2, \dots, s$ , and  $p_{s+1} = 1 - 2n \sum_{i=1}^s p_i$  by definition, we need to show that  $p_{s+1} \geq 0$  for part (1) to hold. This follows by noting that

$$p_{s+1} \geq 1 - 2ns \frac{1}{\min_i q_i} \geq 0.$$

In part (2), equations (23) and (25) are immediate due to the symmetry of the support points around the target mean vector  $\Phi$ . Equation (24) follows by noting that

$$\mathbb{E}[(\mathcal{G} - \Phi)(\mathcal{G} - \Phi)^\top] = 2 \sum_{i=1}^s p_i \frac{1}{2sp_i} \left( \sum_{j=1}^n L_j L_j^\top \right) = \sum_{j=1}^n L_j L_j^\top = R.$$

Finally,

$$\mathbb{E}[(\mathcal{G}_i - \Phi_i)^4] = \sum_{k=1}^s p_k \frac{1}{2p_k^2 s^2} \sum_{j=1}^n L_{ij}^4 = \left( \sum_{k=1}^s q_k \right) \frac{1}{2s^2} \left( \sum_{j=1}^n L_{ij}^4 \right),$$

so that, when  $\psi > 2ns$  holds,

$$\sum_{i=1}^n \mathbb{E}[(\mathcal{G}_i - \Phi_i)^4] = \sum_{i=1}^n \kappa_i,$$

where the last equality follows from the definition of  $q_s$  in step (ii) of the algorithm.  $\blacksquare$

Lemma 1 and its proof above demonstrate one of the main advantages of our method: provided that the weights  $p_i$  form a valid probability measure, their exact values have no impact on the exact matching of  $\Phi$  and  $R$ . In particular,  $p_i$ 's get cancelled in forming the covariance matrix from the support points and the associated probability weights of  $\mathcal{G}$ . If  $\mathcal{G}(k)$  itself represents a discrete time stochastic process, this crucial fact allows us to choose *random* probability weights  $\{p_i\}$  within the specified bounds,  $(\frac{1}{2ns}, \frac{1}{\psi})$  or  $(\frac{1}{2ns}, 0)$ , at each time  $k$ , thereby generating 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.

## 5 What's new in our approach?

The filtering algorithm described in section 3 is similar to various sigma point and ensemble filtering algorithms described elsewhere. However, the sampling approach involved in our method differs radically from those of others as described in section 4. The main differences between sample generation methods in traditional sigma point filters and the filter using the proposed sampling method, which we will henceforth refer to as modified sigma point filter (MSPF) are as follows.

- In the existing methods, the sigma points or samples generated do not *necessarily* form a valid probability distribution, as some of the probability weights can be negative. This puts the probabilistic interpretation of the whole procedure into question. This problem does not arise in MSPF, as seen from the first part of lemma 1.
- The existing sigma point generation algorithms are deterministic, with no way of incorporating random behavior. The randomness may be desirable to reflect the real dynamics of the system, especially when the system is assumed to have explicit sources of randomness. This is especially true when modeling econometric or financial time series. As mentioned earlier, we have the flexibility of using either a deterministic or probabilistic sigma point generation in our algorithm, since  $q_i$  may be generated in either way.
- In addition to the points mentioned above, we can also match the sum of fourth marginal central moments in cases when the condition  $\psi > 2ns$  holds, as detailed in section 4.2. This criterion will often hold for a system with a single measurement and a single state. In case it does not hold, it is possible to minimize the worst case error in matching the marginal fourth central moments of the components of  $\mathcal{G}$  using a convex optimization procedure. The details of this procedure are described in [16] and are hence omitted here.
- Similar to ensemble filters, we do not sample the state  $\mathcal{X}(k|k)$ . This makes intuitive sense since we are sampling all the exogenous sources of randomness and  $\mathcal{X}(k|k)$  is simply a function of these exogenous random processes. This also has a very significant computational advantage over traditional sigma point filters in terms of not having to compute a new matrix square root of covariance matrix at each time step (which would be the case if we were sampling  $\mathcal{X}(k|k)$ ). The ensemble filters, however, use random number generators with the specified distributional properties to generate samples for  $\left[ \mathcal{W}^{(i)}(k+1)^\top \mathcal{V}^{(i)}(k+1)^\top \right]^\top$  and then use the corresponding sample mean and the sample covariances in (21). However, sampling from an underlying distribution with a very small number of samples and then using the sample moments may yield misleading results and it may be better to match a few moments exactly instead, which is what MSPF is designed to

do.

MSPF integrates the numerical simplicity of the ensemble filter with the exact moment matching of the traditional sigma point filter. Further, it extends the moment matching method in traditional sigma point filters beyond matching the mean vector and the covariance matrix. In MSPF, the third marginal moment is matched exactly in all cases and the sum of fourth marginal moments can also be matched exactly in some cases. Achieving even approximate matching of these higher moments in the case of existing sigma point generation algorithms requires non-convex optimization.

## 6 Numerical examples

We consider two different examples to illustrate the proposed filtering method.

### 6.1 CEV type time series model

The first numerical example is a nonlinear, non-Gaussian time series given by

$$\begin{aligned}x(k+1) &= ax(k) + b + \sigma_w \left\{ (x(k))^2 \right\}^{\frac{\gamma}{2}} w(k+1), \\y(k+1) &= cx(k) + d + \sigma_v v(k+1).\end{aligned}\tag{27}$$

Examples of this specification include the constant elasticity of variance (CEV) model in stock option pricing described in Cox [20] and several exponential affine term structure models including the Cox, Ingersoll and Ross model [21]. We consider a univariate example for simplicity. The parameters of the model are  $a = 0.9$ ,  $b = 0.1$ ,  $\sigma_w = 0.01$ ,  $c = 1$ ,  $d = 0.1$ ,  $\sigma_v = 0.01$ . The noise terms  $w(k+1)$  and  $v(k+1)$  are i.i.d. with standard normal distribution. We consider three different values of  $\gamma$ :  $\gamma = 0.125, 0.25, 0.375$ . Sample paths of the state  $x$  and observation  $y$  are generated by sampling  $w(k)$  and  $v(k)$ . Based on the observation sample path, we wish to see whether we can predict  $x(k+1|k)$  accurately using MSPF proposed here and we want to compare its predictive ability with that of the EKF and the ensemble filter. As a measure of performance of a filter, we consider the average of root mean squared error (AvRMSE) in one step ahead prediction. The root mean squared error (RMSE) for a filter and for a particular sample path  $i$  is given by

$$RMSE^{(i)} = \sqrt{\frac{1}{M} \sum_{k=1}^M (x^{(i)}(k+1) - \hat{x}^{(i)}(k+1|k))^2},$$

where the superscript  $i$  denotes the  $i^{th}$  sample path and  $M$  is the time horizon. AvRMSE, as the sample mean  $\frac{1}{N} \sum_{i=1}^N RMSE^{(i)}$ , and VarRMSE as the corresponding sample variance,

$$\text{VarRMSE} = \frac{1}{N-1} \sum_{i=1}^N \left( RMSE^{(i)} - \text{AvRMSE} \right)^2,$$

are computed over  $N=100$  sample paths with each path consisting of  $M=100$  time-steps. At each time-step only 10 samples or sigma points are generated. Table 1 illustrates the results of this error analysis. We see that MSPF yields the lowest AvRMSE and the lowest VarRMSE among the three filtering methods, for all values of  $\gamma$ .

**Table 1 : Comparison of prediction errors using different filters for system in (27). For a specific filter and a value of gamma, the corresponding numbers in the table give the AvRMSE and VarRMSE.**

Filtering method		$\gamma$		
		0.125	0.25	0.375
EKF	AvRMSE	$2.9563 * 10^{-2}$	$2.9584 * 10^{-2}$	$2.9606 * 10^{-2}$
	VarRMSE	$8.8845 * 10^{-5}$	$8.9478 * 10^{-5}$	$9.0154 * 10^{-5}$
Ensemble filter	AvRMSE	$2.6251 * 10^{-2}$	$2.6258 * 10^{-2}$	$2.6267 * 10^{-2}$
	VarRMSE	$3.9879 * 10^{-5}$	$3.9882 * 10^{-5}$	$3.9916 * 10^{-5}$
MSPF	AvRMSE	$2.4960 * 10^{-2}$	$2.4967 * 10^{-2}$	$2.4974 * 10^{-2}$
	VarRMSE	$2.7946 * 10^{-5}$	$2.7888 * 10^{-5}$	$2.7859 * 10^{-5}$

## 6.2 Univariate non-stationary growth model

The second numerical example is a univariate non-stationary growth model given by

$$x(k) = \alpha x(k-1) + \beta \frac{x(k-1)}{1+x^2(k-1)} + \gamma \cos(1.2(k-1)) + \sigma_w w(k), \quad (28)$$

$$y(k) = \frac{x^2(k)}{20} + \sigma_v v(k), \quad (29)$$

where  $v(k)$  are i.i.d.  $N(0,1)$  random variables and  $w(k)$  are i.i.d. random variables with a zero mean and unit variance following a t-distribution having 10 degrees of freedom. A nonlinear model of this type has been discussed

in [22] and in [23].

We use the parameters  $\alpha = 0.5$ ,  $\beta = 28$ ,  $\gamma = 8$ ,  $\sigma_w = 0.1$ ,  $\sigma_v = 0.1$ . In this case, we compare the performance of the ensemble filter and MSPF for one step ahead prediction, using AvRMSE and VarRMSE as defined in the previous example. As in the previous case, AvRMSE is computed over 100 sample paths, each path consisting of 100 time-steps, and 10 sigma points are generated at each time-step. The AvRMSE for MSPF is 0.61172 while that for the ensemble filter is 0.61558. The VarRMSE for MSPF is 0.00028 while that for ensemble filter is 0.00107. As in the previous example, it is seen that the proposed method outperforms the ensemble filter, with a lower AvRMSE and a lower VarRMSE.

While VarRMSE reflects variation of RMSE across different sample paths, it is also worth commenting on the variation in the one step ahead prediction error across different time steps along the same sample path. Even though both the filtering methods, viz. ensemble filter and MSPF use a small number of randomly generated samples for computing moments, the samples in MSPF are likely to yield locally closer predictions due to moment matching constraints. To illustrate this point, a plot of a simulated sample path (denoted by solid line) and the corresponding one step ahead prediction using MSPF (denoted by dashed line) is plotted in figure 1, while the same sample path and one step ahead prediction using ensemble filter is plotted in figure 2. Indeed, this intuition is confirmed by comparing figures 1 and 2 and observing the contrast between the behaviours of the two predictions occurring most notably within the first 20 time-steps.

We do not compare the methods with a linearized filter in this example since the system is too nonlinear for a local linearization procedure to be effective.

## 7 Concluding remarks

We have developed a modified sigma point filtering algorithm for nonlinear and non-Gaussian systems. This algorithm combines the numerical simplicity of the ensemble filter (in the sense that the state covariance matrix need not be factorized) along with the exact moment matching properties of the traditional sigma point filter. Further, while the traditional sigma point filtering methods match only the first two moments, the exact moment matching is extended to three and in some cases four moments in the MSPF. The use of the algorithm is demonstrated through numerical examples.

To the best of our knowledge, sigma point filters have not been adopted widely by researchers in fields such as econometrics, finance and actuarial science. We

Fig. 1. Prediction for univariate non-stationary growth model using MSPF

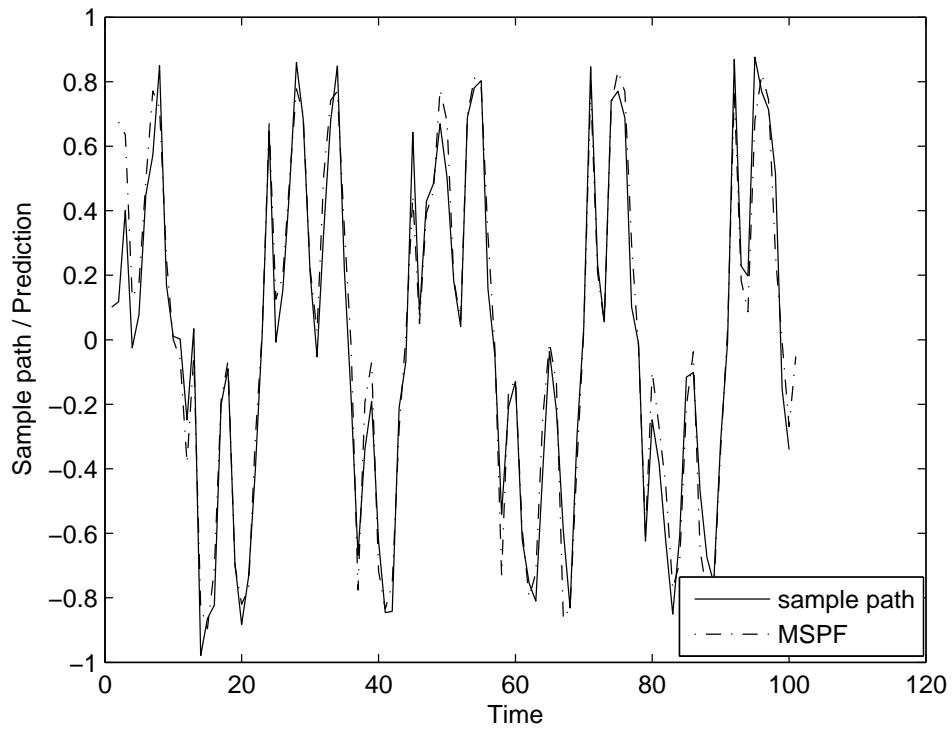
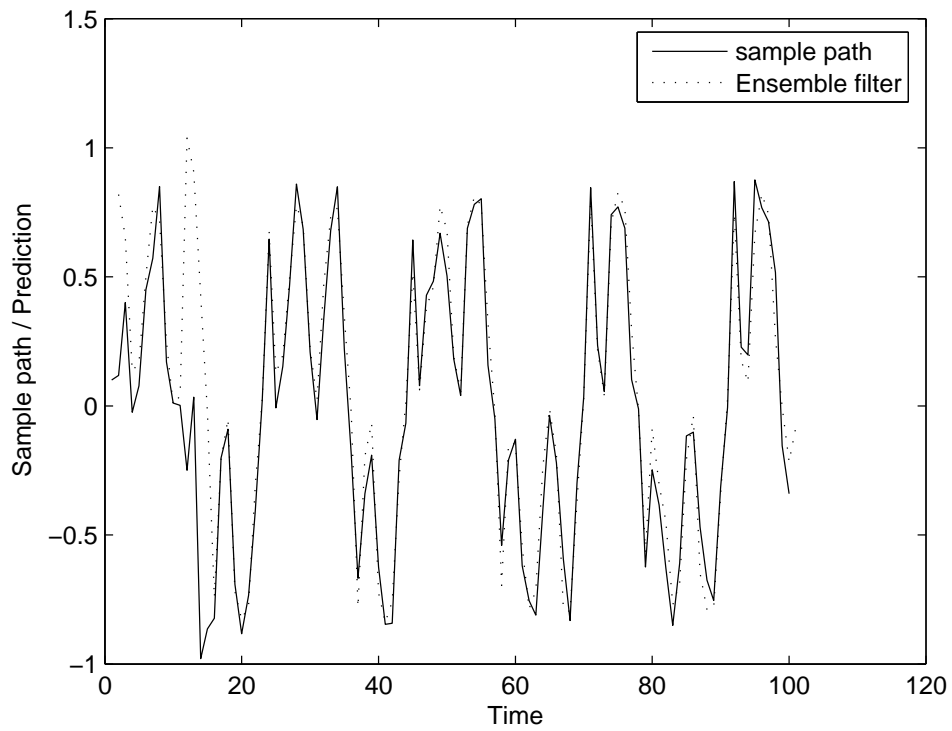


Fig. 2. Prediction for univariate non-stationary growth model using Ensemble filter



feel that with the methodology we develop in this paper that addresses some of the major shortcomings of sigma point filters, the method of sigma point filtering will be more attractive and useful to researchers and practitioners alike in these fields of research. Further, the proposed method also provides a very useful alternative to traditional sigma point filters in engineering and ensemble filters in geosciences.

### *Acknowledgements*

A part of this research was carried out when the first author was visiting University of Western Ontario, Canada on an overseas travel grant from Royal Society, UK.

### **References**

- [1] H. Kushner, *Numerical methods for stochastic control problems in continuous time*. Springer Verlag, 2000.
- [2] A. Jazwinski, *Stochastic processes and filtering theory*. Academic Press, 1970.
- [3] B. Anderson and J. Moore, *Optimal filtering*. Prentice-Hall, 1979.
- [4] J. Lund, “Non-linear Kalman filtering techniques for term structure models,” working paper, The Aarhus School of Business, 1997.
- [5] J. Liu and R. Chen, “Sequential Monte Carlo methods for dynamic systems,” *Journal of American Statistical Association*, vol. 93, pp. 1032–1044, 1998.
- [6] 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.
- [7] 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.
- [8] T. S. R. Karlsson and F. Gustafsson, “Complexity analysis of the marginalized particle filter,” *IEEE Transactions on Signal Processing*, vol. 53, pp. 4408–4411, 2005.
- [9] S. Julier and J. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, pp. 401–422, 2004.
- [10] 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.

- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] H. Mitchell and P. Hotekammer, “Data assimilation using an ensemble Kalman filter technique,” *Monthly Weather Review*, vol. 126, pp. 796–811, 1998.
- [16] P. Date and R. Mamon, “A new moment matching algorithm for sampling from partially specified symmetric distributions.” submitted to *OR Letters*, 2007.
- [17] G. Grimmett and D. Stirzaker, *Probability and Random Processes*. Oxford University Press, 2001.
- [18] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1999.
- [19] P. Bentler and M. Berkane, “Greatest lower bound to the elliptical theory kurtosis parameter,” *Biometrika*, 1986.
- [20] J. Cox, “The constant elasticity of variance option pricing model,” *Journal of Portfolio Management*, vol. 22, pp. 16–17, 1996.
- [21] J. Cox, J. Ingersoll, and S. Ross, “A theory of the term structure of interest rates,” *Econometrica*, vol. 53, pp. 385–408, 1985.
- [22] B. Carlin, N. Polson, and D. Stoffer, “A Monte Carlo approach to nonnormal and nonlinear state-space modeling,” *Journal of the American Statistical Association*, vol. 87, pp. 493–500, 1992.
- [23] G. Kitagawa, “Non-Gaussian state space modeling of nonlinear time series,” *Journal of the American Statistical Association*, vol. 82, pp. 1032–1063, 1987.