Generalized unscented transformation for forecasting non-Gaussian processes

Donald Ebeigbe,[1,*] Tyrus Berry,[2] Andrew J. Whalen,[3,4] Michael M. Norton,[5] Dan Simon,[6] Timothy D. Sauer,[2] and Steven J. Schiff[3,7]

[1]*Department of Electrical Engineering, Pennsylvania State University, University Park, Pennsylvania, USA*
[2]*Department of Mathematical Sciences, George Mason University, Fairfax, Virginia, USA*
[3]*Department of Neurosurgery, School of Medicine, Yale University, New Haven, Connecticut, USA*
[4]*Department of Neurosurgery, Massachusetts General Hospital, Harvard Medical School, Boston, Massachusetts, USA*
[5]*Center for Neural Engineering, Department of Engineering Science and Mechanics, Pennsylvania State University, University Park, Pennsylvania, USA*
[6]*Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, Ohio, USA*
[7]*Department of Epidemiology of Microbial Diseases, Yale University, New Haven, Connecticut, USA*

(Received 12 July 2024; revised 6 February 2025; accepted 26 March 2025; published 27 May 2025)

The observations of linear and nonlinear physical processes are subject to random errors, which can be represented by a wide variety of probability distributions. In contrast, most estimation and inference techniques rely on a Gaussian assumption, which may limit our ability to make model-based predictions. There is a need for data assimilation methods that can capture and leverage the higher moments of these physical processes for state estimation and forecasting. In this paper, we develop the generalized unscented transform (GenUT), which uses a minimal number of sample points to accurately capture elements of the higher moments of most probability distributions. Constraints can be analytically enforced on the sample points while guaranteeing at least second-order accuracy. The GenUT is widely applicable to non-Gaussian distributions, which can substantially improve the assimilation of observations of nonlinear physics, such as the modeling of infectious diseases.

DOI: 10.1103/PhysRevE.111.054135

I. INTRODUCTION

The observations of physical processes are subject to random errors, which can be represented by a wide variety of probability distributions. We often rely on the central limit theorem as a tool to handle these uncertainties in the form of Gaussian statistics. However in practice, the noise distributions encountered in natural systems (as in biology, weather forecasting, oceanography, etc. [1–4]) often violate this assumption of Gaussianity. Additionally, the nonlinear nature of most real-world physical systems limits the ability of Gaussian statistics to accurately parameterize the uncertainties of such systems.

For example, the need to effectively monitor, predict, and control the spread of infectious diseases [5] such as measles, influenza, neonatal sepsis, Ebola, and the novel coronavirus (SARS-CoV-2) causing COVID-19 has led to the application of numerous state estimation techniques [6–13]. Distributions such as Poisson, negative-binomial, and binomial are typically used for modeling infectious disease from count data. When such models are highly nonlinear, the implementation of techniques that can not account for the higher-order moments of such distributions can lead to poor performance.

Standard data assimilation techniques, such as the Kalman filter, provide the basis for most of the popular state estimation techniques used for linear and nonlinear dynamic systems. The linear Kalman filter works by propagating the means and covariance of the state of a dynamic system [14,15]. Originally developed under the assumption of finite second-order moments for measurement and process noise, the Kalman filter is the optimal estimator when this assumption is satisfied [16,17]. Despite its origins, "the Kalman filter provides rigorous and optimal performance guarantees that do not rely on any distribution assumptions beyond mean and error covariance information" [17]. Under non-Gaussian noise, Kalman filter performance can sometimes deteriorate [14,18], especially when the underlying noise distributions have nonzero finite higher-order moments. Although significant work has been done on addressing nonlinearity, with applications to filtering in weather prediction and other highly nonlinear processes like chaotic systems, less work has been done in addressing non-Gaussianity. While the models used in disease filtering are often nonlinear, the greater concern is often the non-Gaussian nature of their evolution and observations [13]. From the point of view of data assimilation for disease modeling, the statistics of the observations may be highly non-Gaussian. The goal of this paper is to address systems that exhibit strong non-Gaussianity.

For many engineered dynamic systems in practice, linearity is a reasonable assumption. For natural systems, nonlinearities cause methods based on linear models to perform poorly. Most nonlinear systems can behave approximately linearly over small operation ranges. The extended Kalman filter (EKF) is one of the most widely used Kalman filter for nonlinear dynamic systems. The EKF employs a linear approximation of the nonlinear system around a nominal state trajectory [14,15,19]. However, for highly nonlinear

*Contact author: dee5127@psu.edu

systems, linear approximations can introduce errors that can lead to divergence of the state estimate.

To address the drawbacks of the EKF, several well-known state estimators such as the ensemble Kalman filter (EnKF) [20–23], the unscented Kalman filter (UKF) [24,25], and the particle filter (PF) [14,26] have been developed. Although the particle filter can give better performance than the UKF, this comes at the cost of a higher computational effort. In some applications, the improved performance might not be worth the additional computational costs [14]. Although the EnKF can easily account for modeling errors, undersampling the ensembles can lead to filter divergence. This motivated the development of deterministic EnKFs that reduce the number of ensemble members to a strict minimum through deterministic ensembles, without causing any degradation in performance [27].

The UKF is a nonlinear filter that uses the unscented transform (UT) to propagate the mean and covariance of random variables [24,28]. The UT uses the intuition that "with a fixed number of parameters it should be easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation" [24]. Although the design of the UT was motivated by the Gaussian assumption, it is often used in contexts outside of those assumptions. It produces sets of vectors called *sigma points* that capture the higher-order moments of the standard Gaussian distribution. In comparison to the Monte Carlo methods used in the PF and EnKF, the UT uses much fewer points, which makes it computationally cost-effective. This is why the UT has been used to improve the performance of the EnKF [29], as well as to generate distributions to improve the performance of the PF [30,31]. Despite the several types of UTs that exist in the literature [32,33], a majority of them that were not developed using the Gaussian assumption do not try to match the skewness and kurtosis of non-Gaussian distributed random variables, thereby ensuring only second-order accuracy in propagating means and covariances.

The usage of the UT to assimilate data generated by nonlinear transformation of random variables revealed a fundamental mismatch in the application of the UT—the accuracy of the UT is reduced if the nonlinearities are high, especially if the Gaussian assumption with its second-order moment statistics is not satisfied. This led to the development of UTs that can account for some higher-order moment information such as the skewness and kurtosis [34–38]. The UTs can be grouped into two categories: the ones that employ $2n+1$ sigma points [34–36] and the ones that use more than $2n+1$ sigma points [37,38].

First, we consider those that use $2n+1$ sigma points. In Ref. [34], a UT was developed to match the average marginal skewness and kurtosis. The method however did not match the true skewness and true kurtosis for each element of the random vector. In Ref. [35], a randomized UT was used in the development of a filter for non-Gaussian systems. Although the method uses a stochastic integration rule to solve state and measurement statistics, the sigma points are generated under the Gaussian assumption. In Ref. [36], a UT was developed to capture the skewness of a random vector. However, the method assumes a closed skew normal distribution in its development. All preceding UTs that use $2n+1$ sigma points were either developed for some special distribution or can capture at most the average skewness and kurtosis.

Now we consider those that use more than $2n+1$ sigma points. In Ref. [37], a UT was developed to match the first four moments of Gaussian random variables. In Ref. [38], a higher-order UT was developed to match the skewness and kurtosis tensors with high accuracy. The method uses an approximate CANDECOMP-PARAFAC (CP) tensor decomposition to generate its sigma points. However, depending on the dimension of the problem and the error tolerance level in approximating the skewness and kurtosis tensors, this method can incur significant computational costs. This is because the sequence of vectors and constants used in the approximate CP method can significantly increase when the error tolerance level is made small. All preceding UTs that use more than $2n+1$ sigma points were either developed for some special distribution or had significantly higher complexity and computational cost.

For an n-dimensional random vector, $2n+1$ sigma points generally employ $2n^2+3n+1$ parameters ($2n+1$ weights and $2n^2+n$ constants that define the coordinates of the sigma points). Trying to match the mean, covariance, skewness, and kurtosis imposes n, $O(n^2)$, $O(n^3)$, and $O(n^4)$ constraints, respectively. In principle, it is impossible to match all these moments using only $2n+1$ sigma points. The zero skewness nature of the Gaussian distribution made it possible to use $2n+1$ sigma points to accurately match up to the skewness in Ref. [24]. The presence of the $O(n^3)$ skewness and $O(n^4)$ kurtosis constraints are what prompted researchers to look beyond $2n+1$ sigma points. However, we note that matching the mean and covariance constraints of any random vector using $2n+1$ sigma points still leaves n^2+2n+1 free parameters—we consider the $2n+1$ weights to be part of the free parameters (one of these free parameters will be lost by constraining the weights to sum to 1). These residual parameters have been underutilized in capturing as much information as possible about the components of the skewness and kurtosis tensors when the random variable is not Gaussian. One instance where the residual parameters were leveraged was in the capturing of the average marginal skewness and kurtosis, which only represents a total of two constraints [34].

In this paper, we develop the *generalized unscented transform* (GenUT) which is able to adapt to the unique statistics of most probability distributions. We use the intuition that employing sigma points more suitable to the inherent distributions of a random vector can lead to a more accurate propagation of means and covariances. Our method uses $2n+1$ sigma points that not only accurately matches the mean and covariance matrix but also takes advantage of the additional free parameters to accurately match the diagonal components of the skewness tensor and kurtosis tensor of most random vectors. We employ n^2+3n constraints in total; n for the mean, n^2 for the covariance (ignoring its symmetric property), n for the diagonal components of the skewness tensor, and n for the diagonal components of the kurtosis tensor. This total falls within the $2n^2+3n+1$ free parameters available. While more parameters remain, the diagonal components of the skewness and kurtosis tensors are the most significant. In comparison with Refs. [34–37], our method gives a general way to accurately match the diagonal components of the skewness

and kurtosis tensors of most random vectors. In comparison with Ref. [38], our method uses fewer sigma points, which is crucial for larger system dimensions. In comparison to the standard unscented transform, we acquire the most significant higher moment information of most probability distributions with the same number of sigma points. Unlike most unscented transforms, our method allows for constraints to be imposed on the sigma points while guaranteeing at least a second-order accuracy.

In Sec. II, we discuss the problems that arise when the Gaussian assumption is employed in the unscented transform. In Sec. III, we develop the GenUT sigma points that can capture certain properties of most probability distributions, such as its mean, covariance, skewness, and kurtosis. In Sec. IV, we show that our sigma points are accurate in approximating the mean, covariance, and diagonal components of the skewness and kurtosis tensors. In Sec. V, we address constraints and show that imposing constraints can at least maintain second-order accuracy. In Sec. VI, we evaluate the accuracy of the GenUT sigma points in propagating means and covariances of nonlinear transformations of arbitrarily distributed random vectors and we give several examples that demonstrate its effectiveness when compared against other unscented transforms. We discuss the conclusions in Sec. VII.

Notation. Vectors are lowercase and bold, while matrices and tensors are uppercase and bold. $\boldsymbol{P}_{[i]}$ represents the ith column of the matrix $\boldsymbol{P}$, $\boldsymbol{P}_{ij}$ represents the ith entry in the jth column of the matrix $\boldsymbol{P}$, and $\boldsymbol{x}_i$ represents the ith entry of the vector $\boldsymbol{x}$. We use $\boldsymbol{x} < \boldsymbol{y}$ to denote the component-wise inequality of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. $|\boldsymbol{x}|$ denotes the component-wise absolute value of the vector $\boldsymbol{x}$, while $\min|\boldsymbol{x}|$ denotes the smallest element of $|\boldsymbol{x}|$.

II. LIMITATIONS OF THE UNSCENTED TRANSFORM

We analyze the performance of unscented transforms that were motivated by Gaussian statistics [24,28]. We show how linearization approximations, via Taylor series expansion of a nonlinear transformation of a random vector $\boldsymbol{x}$ evaluated about its mean $\bar{\boldsymbol{x}}$, introduces errors in the propagation of means and covariances. Taylor series analysis is a standard method of deriving and justifying the unscented transform and its variants [24,28]. The basic assumption is that the distribution is concentrated near the mean and has fast decay away from the mean so that the higher-order terms in the Taylor series, which involve higher moments, are small compared with the leading-order terms. We consider alternative methods of analysis in the conclusion. We see that errors can be introduced in the propagation of means and covariances beyond the second order when used to approximate a nonlinear function $\boldsymbol{\lambda}(\boldsymbol{x})$ of a possibly non-Gaussian distributed random vector $\boldsymbol{x}$.

Definition 1. Let $\boldsymbol{x} \in \mathbb{R}^n$ be a random vector. We define the mean $\bar{\boldsymbol{x}} \in \mathbb{R}^n$, covariance $\boldsymbol{P} \in \mathbb{R}^{n\times n}$, skewness tensor $\boldsymbol{S} \in \mathbb{R}^{n\times n\times n}$, and kurtosis tensor $\boldsymbol{K} \in \mathbb{R}^{n\times n\times n\times n}$ as

$$\bar{\boldsymbol{x}} = \mathbb{E}[\boldsymbol{x}], \tag{1}$$

$$\boldsymbol{P} = \mathbb{E}[(\boldsymbol{x}-\bar{\boldsymbol{x}})(\boldsymbol{x}-\bar{\boldsymbol{x}})^T], \tag{2}$$

$$\boldsymbol{S}_{ijk} = \mathbb{E}[(\boldsymbol{x}-\bar{\boldsymbol{x}})_i(\boldsymbol{x}-\bar{\boldsymbol{x}})_j(\boldsymbol{x}-\bar{\boldsymbol{x}})_k], \tag{3}$$

$$\boldsymbol{K}_{ijkl} = \mathbb{E}[(\boldsymbol{x}-\bar{\boldsymbol{x}})_i(\boldsymbol{x}-\bar{\boldsymbol{x}})_j(\boldsymbol{x}-\bar{\boldsymbol{x}})_k(\boldsymbol{x}-\bar{\boldsymbol{x}})_l] \tag{4}$$

for $i, j, k, l \in \{1, \ldots, n\}$.

The sample mean and sample covariance of the nonlinear transformation $\boldsymbol{y} \in \mathbb{R}^n$ given by

$$\boldsymbol{y} = \boldsymbol{\lambda}(\boldsymbol{x}) \tag{5}$$

can be calculated as follows [24]:

(1) Calculate the $2n+1$ sigma points given by $\boldsymbol{\chi}_{[0]}=\bar{\boldsymbol{x}}$, $\boldsymbol{\chi}_{[i]}=\bar{\boldsymbol{x}}+(\sqrt{(n+\kappa)\boldsymbol{P}})_{[i]}$, and $\boldsymbol{\chi}_{[i+n]}=\bar{\boldsymbol{x}}-(\sqrt{(n+\kappa)\boldsymbol{P}})_{[i]}$. Their corresponding weights are $\boldsymbol{w}_0 = \frac{\kappa}{n+\kappa}$, $\boldsymbol{w}_i = \frac{1}{2(n+\kappa)}$, $\boldsymbol{w}_{i+n} = \frac{1}{2(n+\kappa)}$ for $i \in \{1, \ldots, n\}$, where $(\sqrt{(n+\kappa)\boldsymbol{P}})_{[i]}$ is the ith column of $\sqrt{(n+\kappa)\boldsymbol{P}}$, $\boldsymbol{w}_i$ is the weight associated with the ith sigma point, and κ is a free parameter. We typically set $\kappa = 3-n$ to minimize the fourth-order moment mismatch.

(2) Pass the sigma points through the known nonlinear function to get the transformed sigma points, then evaluate their sample mean and sample covariance using $\boldsymbol{\mathcal{Y}}_{[i]}=\boldsymbol{\lambda}(\boldsymbol{\chi}_{[i]})$, $\bar{\boldsymbol{y}}=\sum_{i=0}^{2n}\boldsymbol{w}_i\boldsymbol{\mathcal{Y}}_{[i]}$, $\boldsymbol{P}_y=\sum_{i=0}^{2n}\boldsymbol{w}_i(\boldsymbol{\mathcal{Y}}_{[i]}-\bar{\boldsymbol{y}})(\boldsymbol{\mathcal{Y}}_{[i]}-\bar{\boldsymbol{y}})^T$.

A. Accuracy in approximating the true mean

Applying a Taylor series expansion of $\boldsymbol{\lambda}(\boldsymbol{x})$ about its mean $\bar{\boldsymbol{x}}$, we show in Appendix A 1 that the true mean of $\boldsymbol{y} = \boldsymbol{\lambda}(\boldsymbol{x})$ is given as

$$\bar{\boldsymbol{y}} = \boldsymbol{\lambda}(\bar{\boldsymbol{x}}) + \left[\sum_{i,j=1}^{n}\frac{\boldsymbol{P}_{ij}}{2!}\frac{\partial^2\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j} + \sum_{i,j,k=1}^{n}\frac{\boldsymbol{S}_{ijk}}{3!}\frac{\partial^3\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k} + \sum_{i,j,k,l=1}^{n}\frac{\boldsymbol{K}_{ijkl}}{4!}\frac{\partial^4\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k\partial\boldsymbol{x}_l}\right]_{\boldsymbol{x}=\bar{\boldsymbol{x}}} + \mathbb{E}\left[\frac{D_{\tilde{\boldsymbol{x}}}^5\boldsymbol{\lambda}}{5!} + \frac{D_{\tilde{\boldsymbol{x}}}^6\boldsymbol{\lambda}}{6!} + \cdots\right]. \tag{6}$$

The analytical expression for the approximated mean from Ref. [24] is given as

$$\bar{\boldsymbol{y}}_u = \boldsymbol{\lambda}(\bar{\boldsymbol{x}}) + \frac{1}{2}\sum_{i,j=1}^{n}\boldsymbol{P}_{ij}\left.\frac{\partial\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} + \frac{1}{2(n+\kappa)}\sum_{i=1}^{2n}\left[\frac{D_{\sigma_i}^4\boldsymbol{\lambda}}{4!} + \frac{D_{\sigma_i}^6\boldsymbol{\lambda}}{6!} + \cdots\right]. \tag{7}$$

Comparing the above equation with the true mean of (6), we notice the following problems about the sigma points developed using the Gaussian assumption:

(1) The odd-powered moments in the approximation of the true mean are always zero due to their symmetry. This introduces significant approximation errors in situations where the odd-powered moments of the distribution of $\boldsymbol{x}$ are nonzero and the transformation $\boldsymbol{y} = \boldsymbol{\lambda}(\boldsymbol{x})$ is highly nonlinear.

(2) The fourth-order term fails to capture a part of the true kurtosis even when the optimal value of $\kappa = n-3$ is selected because of the Gaussian assumption.

Errors in approximating the mean beyond the second order occur not only for sets of $2n+1$ sigma points existing in the literature, but also for sets of $n+1$ sigma points [25,39]—this is because they do not account for the skewness and kurtosis of $\boldsymbol{x}$ when it is not Gaussian distributed.

Note that (6) also gives an insight into the sensitivity of the outputs to perturbations in skewness and kurtosis. In this paper, we assume that the true skewness and kurtosis are

available as inputs. However, in practice, one may only have noisy estimates. We can see from (6) that the sensitivity of the output to perturbations of the skewness will be proportional to the third derivatives of $\boldsymbol{\lambda}$. Similarly, the sensitivity of the output to perturbations of the kurtosis will be proportional to the fourth derivatives of $\boldsymbol{\lambda}$.

B. Accuracy in approximating the true covariance matrix

The true covariance matrix, which was evaluated in Appendix A 2, is given as

$$\begin{aligned}\boldsymbol{P}_y = \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{\Lambda}^T &+ \left[\sum_{i,j,k=1}^{n} \frac{\boldsymbol{S}_{ijk}}{2!}\left[\frac{\partial^2\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j}\frac{\partial\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_k} + \frac{\partial\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i}\frac{\partial^2\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k}\right]\right.\\ &+ \sum_{i,j,k,l=1}^{n} \boldsymbol{K}_{ijkl}\left[\frac{1}{3!}\frac{\partial^3\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k}\frac{\partial\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_l}\right.\\ &+ \frac{1}{3!}\frac{\partial\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i}\frac{\partial^3\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k\partial\boldsymbol{x}_l} + \frac{1}{4}\frac{\partial^2\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j}\frac{\partial^2\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_k\partial\boldsymbol{x}_l}\left.\right]\\ &+ \left.\left[\sum_{i,j=1}^{n}\frac{\boldsymbol{P}_{ij}}{2}\frac{\partial^2\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j}\right][\cdots]^T\right]_{\boldsymbol{x}=\bar{\boldsymbol{x}}} + \cdots,\end{aligned} \tag{8}$$

where we have used the notation $\boldsymbol{x}\boldsymbol{x}^T = \boldsymbol{x}[\cdots]^T$. The analytical expression for the approximated covariance matrix from [24] is given as

$$\begin{aligned}\boldsymbol{P}_u = \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{\Lambda}^T &+ \frac{1}{2(n+\kappa)}\sum_{i}^{2n}\left[\frac{D_{\sigma_i}\boldsymbol{\lambda}(D^3_{\sigma_i}\boldsymbol{\lambda})^T}{3!}\right.\\ &+ \left.\frac{D^3_{\sigma_i}\boldsymbol{\lambda}\left(D_{\sigma_i}\boldsymbol{\lambda}\right)^T}{3!} + \frac{D^2_{\sigma_i}\boldsymbol{\lambda}(D^2_{\sigma_i}\boldsymbol{\lambda})^T}{2!\times 2!}\right]\\ &+ \left[\frac{1}{2}\sum_{i,j=1}^{n}\boldsymbol{P}_{ij}\left.\frac{\partial^2\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}\right][\cdots]^T + \cdots.\end{aligned} \tag{9}$$

Comparing the above equation with the true covariance matrix of (8), we notice similar issues that were pointed out in approximating the mean—the approximation is only accurate up to the second order when $\boldsymbol{x}$ is not Gaussian distributed. All the odd-powered moments are zero because of the symmetric nature of the sigma points, while the fourth-powered moment is also inaccurate because of the Gaussian nature of the sigma points. As with the mean approximation, errors in the covariance matrix approximation are introduced beyond the second order not only for sets of $2n+1$ sigma points existing in the literature, but also for sets of $n+1$ sigma points.

III. GENERALIZED UNSCENTED TRANSFORM

For a random vector $\boldsymbol{x} \in \mathbb{R}^n$, we develop sigma points that can accurately capture the mean, covariance matrix, and the diagonal components of both the skewness tensor and the kurtosis tensor. This is done by selecting sigma point distributions that have the flexibility to either be symmetric when $\boldsymbol{x}$ is symmetrically distributed or be asymmetric when $\boldsymbol{x}$ is asymmetrically distributed.

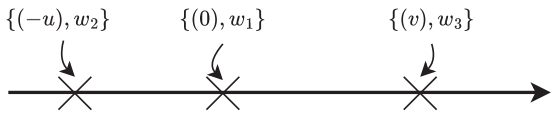


FIG. 1. Samples chosen for a one-dimensional distribution for the GenUT. The locations and weights of the sigma points are determined by the moments of the probability distribution.

Assumption 1. The random vector $\boldsymbol{x}$ follows a probability distribution with finite moments.

We reduce the problem of approximating $\boldsymbol{x}$ to the problem of approximating a user-specified arbitrarily distributed random vector $\boldsymbol{z} \in \mathbb{R}^n$ with zero mean and unit variance, whose higher-order moments are functions of the higher-order moments of $\boldsymbol{x}$. We write

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + (\sqrt{\boldsymbol{P}})\boldsymbol{z}, \tag{10}$$

where $\sqrt{\boldsymbol{P}}$ is the matrix square root of $\boldsymbol{P}$, $\sqrt{\boldsymbol{P}}\sqrt{\boldsymbol{P}}^T = \boldsymbol{P}$.

Definition 2. Let $\boldsymbol{x}$ be a vector, $\boldsymbol{P}$ be a square matrix, and k be some positive integer. We define the element-wise product (Hadamard product) as $\odot$, such that

$$\boldsymbol{x}^{\odot k} = \underbrace{\boldsymbol{x}\odot\boldsymbol{x}\odot\cdots\odot\boldsymbol{x}}_{\text{k times}},$$
$$(\boldsymbol{P}^{\odot k})^{-1} = (\underbrace{\boldsymbol{P}\odot\boldsymbol{P}\odot\cdots\odot\boldsymbol{P}}_{\text{k times}})^{-1}.$$

We also define the element-wise division (Hadamard division) as $\oslash$.

A. One-dimensional distribution

We develop sigma points that match the first three moments of $\boldsymbol{z}$ in a single dimension, and then constrain those points to match the fourth moment of $\boldsymbol{z}$. For a one-dimensional distribution, we show how to select sigma points such that the first four moments satisfy

$$\mathbb{E}[z_i] = 0, \qquad \mathbb{E}[(z-\bar{z})^2] = 1,$$
$$\mathbb{E}[(z-\bar{z})^3] = \frac{\boldsymbol{S}}{\sqrt{\boldsymbol{P}}^3}, \qquad \mathbb{E}[(z-\bar{z})^4] = \frac{\boldsymbol{K}}{\boldsymbol{P}^2}.$$

To capture the first three moments in a single dimension, three points are used: the first point lies at the origin with a weight of $\boldsymbol{w}_0$; the second point lies at a distance $-\boldsymbol{u}$ from the origin with a weight of $\boldsymbol{w}_1$; the third point lies at a distance $\boldsymbol{v}$ from the origin with a weight of $\boldsymbol{w}_2$. Therefore, in one-dimension, we use the following three sigma points:

$$\boldsymbol{\chi}_1 = 0, \boldsymbol{w}_1,$$
$$\boldsymbol{\chi}_2 = -\boldsymbol{u}, \boldsymbol{w}_2,$$
$$\boldsymbol{\chi}_3 = \boldsymbol{v}, \boldsymbol{w}_3,$$

where $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{w}_3$ are the weights for the respective sigma points. A visual representation of our sigma points in one dimension is shown in Fig. 1. Obeying the moments of $\boldsymbol{z}$ and

the fact that the sum of all weights should equal 1, we write

$$\boldsymbol{w_1} + \boldsymbol{w}_2 + \boldsymbol{w}_3 = 1, \tag{11}$$

$$-\boldsymbol{w}_2\boldsymbol{u} + \boldsymbol{w}_3\boldsymbol{v} = 0, \tag{12}$$

$$\boldsymbol{w_2}\boldsymbol{u}^2 + \boldsymbol{w}_3\boldsymbol{v}^2 = 1, \tag{13}$$

$$-\boldsymbol{w}_2\boldsymbol{u}^3 + \boldsymbol{w}_3\boldsymbol{v}^3 = \boldsymbol{S}\sqrt{\boldsymbol{P}}^{-3}. \tag{14}$$

From (12), we see that $\boldsymbol{w_2} = \frac{\boldsymbol{v}}{\boldsymbol{u}}\boldsymbol{w}_3$. Rewriting (13) using (14) gives

$$\boldsymbol{w}_3\boldsymbol{v}(\boldsymbol{u} + \boldsymbol{v}) = 1, \tag{15}$$

$$\boldsymbol{w}_3\boldsymbol{v}(\boldsymbol{v}^2 - \boldsymbol{u}^2) = \boldsymbol{S}\sqrt{\boldsymbol{P}}^{-3}. \tag{16}$$

We designate $\boldsymbol{u}$ as the free parameter while assuming that $\boldsymbol{u} > 0$. Using the fact that $\boldsymbol{v}^2 - \boldsymbol{u}^2 = (\boldsymbol{u} + \boldsymbol{v})(\boldsymbol{v} - \boldsymbol{u})$, substituting (15) into (16) gives

$$\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{S}\sqrt{\boldsymbol{P}}^{-3}. \tag{17}$$

From (11) and (15), we see that the weights are given as

$$\boldsymbol{w}_3 = \frac{1}{\boldsymbol{v}(\boldsymbol{u} + \boldsymbol{v})}, \quad \boldsymbol{w}_1 = 1 - \boldsymbol{w}_2 - \boldsymbol{w}_3. \tag{18}$$

We note that the free parameter $\boldsymbol{u}$ can be selected to match the fourth moment of $\boldsymbol{z}$. We now attempt to satisfy the fourth moment constraint given by

$$\boldsymbol{w}_2\boldsymbol{u}^4 + \boldsymbol{w}_3\boldsymbol{v}^4 = \boldsymbol{K}\boldsymbol{P}^{-2}. \tag{19}$$

Eliminating $\boldsymbol{w}_2$ using $\boldsymbol{w}_2 = \frac{\boldsymbol{v}}{\boldsymbol{u}}\boldsymbol{w}_3$ gives

$$\boldsymbol{w}_3\boldsymbol{v}(\boldsymbol{u}^3 + \boldsymbol{v}^3) = \boldsymbol{K}\boldsymbol{P}^{-2}. \tag{20}$$

Using the relationships $\boldsymbol{w}_3\boldsymbol{v}(\boldsymbol{u} + \boldsymbol{v}) = 1$, $\boldsymbol{u}^3 + \boldsymbol{v}^3 = (\boldsymbol{u} + \boldsymbol{v})(\boldsymbol{u}^2 + \boldsymbol{v}^2 - \boldsymbol{u}\boldsymbol{v})$, and $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{S}\sqrt{\boldsymbol{P}}^{-3}$, the above equation reduces to

$$\boldsymbol{u}^2 + \boldsymbol{S}\sqrt{\boldsymbol{P}}^{-3}\boldsymbol{u} + \boldsymbol{S}^2\sqrt{\boldsymbol{P}}^{-6} - \boldsymbol{K}\boldsymbol{P}^{-2} = 0.$$

The solution to the above quadratic equation is

$$\boldsymbol{u} = \tfrac{1}{2}\Big[-\boldsymbol{S}\sqrt{\boldsymbol{P}}^{-3} + \sqrt{4\boldsymbol{K}\boldsymbol{P}^{-2} - 3\boldsymbol{S}^2\sqrt{\boldsymbol{P}}^{-6}}\Big], \tag{21}$$

where $\boldsymbol{v}$ is given in (17). The equations for $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{w}_0$ remain unchanged.

Remark 1. We note that the sigma points described above, which accurately capture the kurtosis when constrained, were designed for when the state has a dimension of 1. This implies that $\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{S}, \boldsymbol{K} \in \mathbb{R}^1$.

In the next section, we extend this to multiple dimensions.

B. Multidimensional distribution

For an n-dimensional vector $\boldsymbol{z}$, we develop a set of sigma points that accurately matches its mean and covariance matrix, while accurately matching the diagonal components of the skewness tensor. Furthermore, by constraining the sigma points, we show that we can accurately match the diagonal components of the kurtosis tensor. We note that for an independent random vector, accurately matching the diagonal components of the skewness tensor implies an accurate matching of the entire skewness tensor.

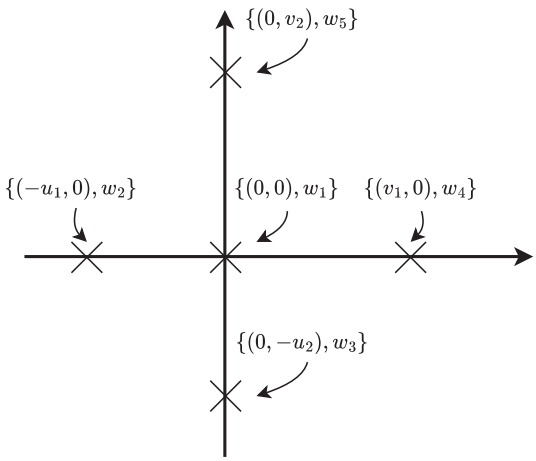


FIG. 2. Samples chosen for a two-dimensional distribution for the GenUT. The locations and weights of the sigma points are determined by the moments of the probability distribution.

Remark 2. For a multi-dimensional random vector $\boldsymbol{x}$, we do not place any assumption of independence on the elements of $\boldsymbol{x}$; they are allowed to be correlated. This agrees with the fact that, in practice, random vectors and their transformations are not always guaranteed to be uncorrelated.

Definition 3. We define the vectors $\check{\boldsymbol{S}} \in \mathbb{R}^n$ and $\check{\boldsymbol{K}} \in \mathbb{R}^n$ which contain the diagonal components of the skewness tensor and kurtosis tensor, respectively, such that

$$\check{\boldsymbol{S}} = [\boldsymbol{S}_{111}, \boldsymbol{S}_{222}, \ldots, \boldsymbol{S}_{nnn}]^T,$$
$$\check{\boldsymbol{K}} = [\boldsymbol{K}_{1111}, \boldsymbol{K}_{2222}, \ldots, \boldsymbol{K}_{nnnn}]^T.$$

For a multidimensional distribution, we show how to select the $2n + 1$ sigma points such that the first four moments satisfy

$$\mathbb{E}[\boldsymbol{z}] = \boldsymbol{0},$$
$$\mathbb{E}[(\boldsymbol{z} - \bar{\boldsymbol{z}})(\boldsymbol{z} - \bar{\boldsymbol{z}})^T] = \boldsymbol{I},$$
$$\mathbb{E}[(\boldsymbol{z} - \bar{\boldsymbol{z}})^{\odot 3}] = (\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\check{\boldsymbol{S}},$$
$$\mathbb{E}[(\boldsymbol{z} - \bar{\boldsymbol{z}})^{\odot 4}] = (\sqrt{\boldsymbol{P}}^{\odot 4})^{-1}\check{\boldsymbol{K}},$$

where $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ is the identity matrix.

Remark 3. Due to the positive definiteness of the covariance matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$, it is always invertible.

A visual representation of our sigma points for a two-dimensional distribution is shown in Fig. 2. Our first point lies at (0,0) with a weight of $\boldsymbol{w}_0$. Our second point lies on the coordinate axes a distance $-\boldsymbol{u}_1$ from the origin with a weight of $\boldsymbol{w}_1$. Our third point lies on the coordinate axes a distance $-\boldsymbol{u}_2$ from the origin with a weight of $\boldsymbol{w}_2$. Our fourth point lies on the coordinate axes a distance $\boldsymbol{v}_1$ from the origin with a weight of $\boldsymbol{w}_3$. Our fifth point lies on the coordinate axes a distance $\boldsymbol{v}_2$ from the origin with a weight of $\boldsymbol{w}_4$. Therefore, our unscented transform uses the following $2n + 1$ sigma points

$$\boldsymbol{\chi}_{[1]} = \boldsymbol{0}, \boldsymbol{w}_1,$$
$$\boldsymbol{\chi}_{[i+1]} = -\boldsymbol{u}_i\boldsymbol{I}_{[i]}, \boldsymbol{w}_{i+1}, \quad i = 1, \ldots, n,$$
$$\boldsymbol{\chi}_{[i+n+1]} = -\boldsymbol{v}_i\boldsymbol{I}_{[i]}, \boldsymbol{w}_{i+n+1}, \quad i = 1, \ldots, n,$$

where $\boldsymbol{I}_{[i]}$ is the ith column of the identity matrix. $\boldsymbol{0} \in \mathbb{R}^n$ is a vector of zeros. We note that $\boldsymbol{u} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]^T$ and $\boldsymbol{v} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]^T$.

Definition 4. We partition the weight vector as $\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}'^T, \boldsymbol{w}''^T]^T$, where $\boldsymbol{w}' = [\boldsymbol{w}_2, \boldsymbol{w}_3, \ldots, \boldsymbol{w}_{n+1}]^T$ and $\boldsymbol{w}'' = [\boldsymbol{w}_{n+2}, \boldsymbol{w}_{n+3}, \ldots, \boldsymbol{w}_{2n+1}]^T$.

Obeying the moments of $\boldsymbol{z}$, we write

$$\sum_{i=1}^{2n+1} \boldsymbol{w}_i = 1, \tag{22}$$

$$-\boldsymbol{w}' \odot \boldsymbol{u} + \boldsymbol{w}'' \odot \boldsymbol{v} = \boldsymbol{0}, \tag{23}$$

$$\boldsymbol{w}' \odot \boldsymbol{u}^{\odot 2} + \boldsymbol{w}'' \odot \boldsymbol{v}^{\odot 2} = \boldsymbol{1}, \tag{24}$$

$$-\boldsymbol{w}' \odot \boldsymbol{u}^{\odot 3} + \boldsymbol{w}'' \odot \boldsymbol{v}^{\odot 3} = (\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}}, \tag{25}$$

where $\boldsymbol{1} \in \mathbb{R}^n$ is a vector of ones. From (23), we see that $\boldsymbol{w}' = \boldsymbol{w}'' \odot \boldsymbol{v} \oslash \boldsymbol{u}$. Rewriting (24) and (25) gives

$$\boldsymbol{w}'' \odot \boldsymbol{v} \odot (\boldsymbol{u} + \boldsymbol{v}) = \boldsymbol{1}, \tag{26}$$

$$\boldsymbol{w}'' \odot \boldsymbol{v} \odot (\boldsymbol{u} + \boldsymbol{v}) \odot (\boldsymbol{v} - \boldsymbol{u}) = (\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}}. \tag{27}$$

Selecting $\boldsymbol{u} > \boldsymbol{0}$ as the free parameters, we get

$$\boldsymbol{v} = \boldsymbol{u} + (\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}}. \tag{28}$$

Therefore, from (22) and (26), we see that

$$\boldsymbol{w}'' = \boldsymbol{1} \oslash \boldsymbol{v} \oslash (\boldsymbol{u} + \boldsymbol{v}), \quad \boldsymbol{w}_1 = 1 - \sum_{i=2}^{2n+1} \boldsymbol{w}_i. \tag{29}$$

To match the diagonal components of the kurtosis tensor, we need to satisfy

$$\boldsymbol{w}' \odot \boldsymbol{u}^{\odot 4} + \boldsymbol{w}'' \odot \boldsymbol{v}^{\odot 4} = (\sqrt{\boldsymbol{P}}^{\odot 4})^{-1}\breve{\boldsymbol{K}}. \tag{30}$$

Solving the above equation results in constrained values for $\boldsymbol{u}$ such that

$$\boldsymbol{u} = -\tfrac{1}{2}(\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}} + \tfrac{1}{2}\sqrt{4(\sqrt{\boldsymbol{P}}^{\odot 4})^{-1}\breve{\boldsymbol{K}} - 3[(\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}}]^{\odot 2}}. \tag{31}$$

It can be shown from (10) that the algorithm for selecting the $2n+1$ sigma points for any random vector $\boldsymbol{x}$ is given in Algorithm 1.

We recall from (28) that the constraint $\boldsymbol{u} > \boldsymbol{0}$ exists. Applying this constraint on (31), we see that

$$\breve{\boldsymbol{K}} > \sqrt{\boldsymbol{P}}^{\odot 4}[(\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}}]^{\odot 2}. \tag{32}$$

The inequality in (32)—at least for a one-dimensional case—agrees with the findings by Pearson in Ref. [40] that for probability distributions, the standardized kurtosis always exceeds the square of the standardized skewness. If the inequality in (32) were violated, then (31) becomes infeasible, which in turn requires the free parameter $\boldsymbol{u} > \boldsymbol{0}$ in (28) to be selected such that $\boldsymbol{v} > \boldsymbol{0}$—although this eliminates the accuracy in matching the diagonal components of the kurtosis tensor, the sigma points are still able to accurately match the diagonal components of the skewness tensor.

There might be concerns that $\boldsymbol{v}$ in (28) might be negative whenever the term $(\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}}$ is negative. If (32) is satisfied, then selecting $\boldsymbol{u}$ using (31) leads to $\boldsymbol{v} > \boldsymbol{0}$. Alternatively, arbitrarily selecting $\boldsymbol{u}$ such that $\boldsymbol{u} > (\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}}$ ensures that $\boldsymbol{v} > \boldsymbol{0}$.

ALGORITHM 1. Sigma points for the generalized unscented transform.

1 *Prescribe the mean* $\bar{\boldsymbol{x}}$*, covariance* $\boldsymbol{P}$*, diagonal component of the skewness tensor* $\breve{\boldsymbol{S}} = [\boldsymbol{S}_{111}, \boldsymbol{S}_{222}, \ldots, \boldsymbol{S}_{nnn}]^T$*, and the diagonal components of the kurtosis tensor* $\breve{\boldsymbol{K}} = [\boldsymbol{S}_{1111}, \boldsymbol{K}_{2222}, \ldots, \boldsymbol{K}_{nnnn}]^T$;

2 *Choose the free parameter vector* $\boldsymbol{u} > \boldsymbol{0}$;

3 *Calculate the parameter vector* $\boldsymbol{v}$,
$$\boldsymbol{v} = \boldsymbol{u} + (\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\breve{\boldsymbol{S}};$$

4 *Calculate the* $2n+1$ *sigma points*,
$$\boldsymbol{\chi}_{[1]} = \bar{\boldsymbol{x}},$$
$$\boldsymbol{\chi}_{[i+1]} = \bar{\boldsymbol{x}} - \boldsymbol{u}_i\sqrt{\boldsymbol{P}}_{[i]},$$
$$\boldsymbol{\chi}_{[i+n+1]} = \bar{\boldsymbol{x}} + \boldsymbol{v}_i\sqrt{\boldsymbol{P}}_{[i]},$$
for $i \in \{1, \ldots, n\}$, where $\sqrt{\boldsymbol{P}}_{[i]}$ is the ith column of the matrix square root of $\boldsymbol{P}$;

5 *Calculate the weights*,
$$\boldsymbol{w}'' = \boldsymbol{1} \oslash \boldsymbol{v} \oslash (\boldsymbol{u} + \boldsymbol{v}),$$
$$\boldsymbol{w}' = \boldsymbol{w}'' \odot \boldsymbol{v} \oslash \boldsymbol{u}, \quad \boldsymbol{w}_1 = 1 - \textstyle\sum_{i=2}^{2n+1} \boldsymbol{w}_i,$$
where $\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}'^T, \boldsymbol{w}''^T]^T$.

Note: To match the diagonal components of the kurtosis tensor, select $\boldsymbol{u}$ using (31) in step 2.

Algorithm 1 can be used to create sigma points that can match up to the kurtosis if (32) is satisfied. For example, we want to prescribe some arbitrary mean, variance, skewness, and kurtosis for a random variable x that is not from any known probability distribution. Randomly selecting the mean $\bar{\boldsymbol{x}}$, variance $\boldsymbol{P}$, and skewness $\boldsymbol{S}$ as $\bar{\boldsymbol{x}} = 0.1$, $\boldsymbol{P} = 0.2$, and $\boldsymbol{S} = -0.5$, respectively, we can use Algorithm 1 to match them exactly. However, we can not randomly select a kurtosis $\boldsymbol{K}$ and expect to match it. The selection of the kurtosis $\boldsymbol{K}$ must satisfy (32), so for this example, we require $\boldsymbol{K} > (-0.5)^2/0.2 = 1.25$. Prescribing a kurtosis value of $\boldsymbol{K} = 1.3$ satisfies (32). Now using Algorithm 1, we see that $\boldsymbol{w}_1 = 0.2$, $\boldsymbol{w}_2 = 0.0286$, $\boldsymbol{w}_3 = 0.7714$, $\boldsymbol{u} = 5.8055$, and $\boldsymbol{v} = 0.2153$. The sample mean, sample covariance, sample skewness, and sample kurtosis exactly match their true prescribed values. We show how to calculate the sample statistics in Sec. IV.

C. Moments of a probability distribution

We use the moment generating function (MGF) $M(t)$ to evaluate the mean and higher-order central moments of a probability distribution. For any random variable x [41], its MGF and nth moment are given by

$$M(t) = \mathbb{E}[e^{tx}], \quad \mathbb{E}[x^n] = \left.\frac{\partial^n M}{\partial t^n}\right|_{t=0}. \tag{33}$$

We also use the gamma notation

$$\Gamma(k) = \int_0^\infty x^{k-1}e^{-x}dx. \tag{34}$$

The first four moments of ten different probability distributions can be found in Table I.

TABLE I. Probability distributions.

Random variable	Probability density function	Mean $\mathbb{E}[\boldsymbol{x}]$	Variance ($\boldsymbol{P}$) $\mathbb{E}[(\boldsymbol{x}-\bar{\boldsymbol{x}})^2]$	Skewness ($\boldsymbol{S}$) $\mathbb{E}[(\boldsymbol{x}-\bar{\boldsymbol{x}})^3]$	Kurtosis ($\boldsymbol{K}$) $\mathbb{E}[(\boldsymbol{x}-\bar{\boldsymbol{x}})^4]$
Gaussian $\mathcal{N}(\mu,\sigma^2)$	$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$, $x\in(-\infty,\infty)$	μ	σ^2	0	$3\sigma^4$
Exponential $E(\lambda)$	$\lambda e^{-\lambda x},\ x\geqslant 0,\lambda>0$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	$\frac{2}{\lambda^3}$	$\frac{9}{\lambda^4}$
Gamma $G(a,b)$	$\frac{x^{a-1}}{\Gamma(a)b^a}e^{\frac{-x}{b}}$, $x\geqslant 0,a>0,b>0$	ab	ab^2	$2ab^3$	$3ab^4(a+2)$
Weibull $W(a,b)$	$\frac{b}{a}\left(\frac{x}{a}\right)^{b-1}e^{-(\frac{x}{a})^b}$, $x\geqslant 0,\ a>0,\ b>0$, $\Gamma_{kb}=\Gamma\left(\frac{k}{b}+1\right)$	$a\Gamma_{1b}$	$a^2\left[\Gamma_{2b}-\Gamma_{1b}^2\right]$	$a^3\left(\Gamma_{3b}+2\Gamma_{1b}^3-3\Gamma_{1b}\Gamma_{2b}\right)$	$a^4\left(\Gamma_{4b}-3\Gamma_{1b}^4-4\Gamma_{1b}\Gamma_{3b}+6\Gamma_{1b}^2\Gamma_2\right)$
Rayleigh $R(\sigma)$	$\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}},\ x\geqslant 0$	$\sigma\sqrt{\frac{\pi}{2}}$	$\sigma^2\left(2-\frac{\pi}{2}\right)$	$\sigma^3(\pi-3)\sqrt{\frac{\pi}{2}}$	$\sigma^4\left(\frac{32-3\pi^2}{4}\right)$
Beta $BE(a,b)$	$\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}x^{a-1}(1-x)^{b-1}$, $x\in(0,1),\ a>0,\ b>0$, $\zeta_k=a+b+k$	$\frac{a}{\zeta_0}$	$\frac{ab}{\zeta_0^2\zeta_1}$	$\frac{2ab(b-a)}{\zeta_0^3\zeta_1\zeta_2}$	$\frac{3ab(2(b-a)^2+ab\zeta_2}{\zeta_0^4\zeta_1\zeta_2\zeta_3}$
Binomial $B(n,p)$	$\binom{n}{k}p^k(1-p)^{n-k}$, $p\in[0,1],\ k=0,1,2,\ldots,n$	np	$np(1-p)$	$np(1-p)(1-2p)$	$np(1-p)[1+p(1-p)(3n-6)]$
Poisson $P(\lambda)$	$\frac{\lambda^k}{k!}e^{-\lambda},\ \lambda>0$, $k=0,1,2,\ldots,\infty$	λ	λ	λ	$3\lambda^2+\lambda$
Geometric $GE(p)$	$p(1-p)^k,\ p\in(0,1]$, $k=0,1,2,\ldots,\infty$	$\frac{(1-p)}{p}$	$\frac{(1-p)}{p^2}$	$\frac{(p-1)(p-2)}{p^3}$	$\frac{(1-p)(p^2-9p+9)}{p^4}$
Negative binomial $NB(r,p)$	$\binom{r+k-1}{k}p^r(1-p)^r$, $k=0,1,2,\ldots,\infty$	$\frac{r(1-p)}{p}$	$\frac{r(1-p)}{p^2}$	$\frac{r(p-1)(p-2)}{p^3}$	$\frac{r(1-p)(p^2-6p-3pr+3r+6)}{p^4}$

IV. ACCURACY OF SIGMA-POINT SAMPLE STATISTICS

We demonstrate the accuracy of our sigma points in approximating any random vector $\boldsymbol{x}\in\mathbb{R}^n$.

Theorem 1. Let $\boldsymbol{x}\in\mathbb{R}^n$ be any random vector with mean $\bar{\boldsymbol{x}}$, covariance matrix $\boldsymbol{P}$, skewness tensor $\boldsymbol{S}$, and kurtosis tensor $\boldsymbol{K}$. The following statements are true for the $2n+1$ sigma points defined in Algorithm 1.

(1) The sample mean $\hat{\bar{\boldsymbol{x}}}=\sum_{i=1}^{2n+1}\boldsymbol{w}_i\boldsymbol{\chi}_{[i]}$ equals $\bar{\boldsymbol{x}}$.

(2) The sample covariance matrix $\hat{\boldsymbol{P}}=\sum_{i=1}^{2n+1}\boldsymbol{w}_i(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})^T$ equals $\boldsymbol{P}$.

(3) The sample skewness tensor $\hat{\boldsymbol{S}}_{jkl}=\sum_{i=1}^{2n+1}\boldsymbol{w}_i(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})_j(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})_k(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})_l$ equals $\boldsymbol{S}_{jkl}$ if $j=k=l$.

(4) The sample kurtosis tensor $\hat{\boldsymbol{K}}_{jklm}=\sum_{i=1}^{2n+1}\boldsymbol{w}_i(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})_j(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})_k(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})_l(\boldsymbol{\chi}_{[i]}-\hat{\bar{\boldsymbol{x}}})_m$, equals $\boldsymbol{K}_{jklm}$ if $j=k=l=m$ whenever $\boldsymbol{u}$ of (31) is used.

Proof. For our proof, we introduce diagonal matrices $\boldsymbol{U},\boldsymbol{V}\in\mathbb{R}^{n\times n}$ such that $\boldsymbol{U}=\mathrm{diag}(\boldsymbol{u})$ and $\boldsymbol{V}=\mathrm{diag}(\boldsymbol{v})$. In matrix form, we evaluate the sample mean as

$$\begin{aligned}\hat{\bar{\boldsymbol{x}}}&=[\bar{\boldsymbol{x}},\quad \bar{\boldsymbol{x}}-\sqrt{\boldsymbol{P}}\boldsymbol{U},\quad \bar{\boldsymbol{x}}+\sqrt{\boldsymbol{P}}\boldsymbol{V}][\boldsymbol{w}_1,\quad \boldsymbol{w}'^T,\quad \boldsymbol{w}''^T]^T\\&=\bar{\boldsymbol{x}}\sum_{i=1}^{2n+1}\boldsymbol{w}_i+\sqrt{\boldsymbol{P}}(\boldsymbol{V}\boldsymbol{w}''-\boldsymbol{U}\boldsymbol{w}')\\&=\bar{\boldsymbol{x}}+\sqrt{\boldsymbol{P}}(\boldsymbol{w}''\odot\boldsymbol{v}-\boldsymbol{w}'\odot\boldsymbol{u})=\bar{\boldsymbol{x}}\end{aligned}\tag{35}$$

because $\sum_{i=1}^{2n+1}\boldsymbol{w}_i=1$ and $\boldsymbol{w}''\odot\boldsymbol{v}=\boldsymbol{w}'\odot\boldsymbol{u}$. We see that the sample mean equals the actual mean. Evaluating the sample covariance matrix, we get

$$\begin{aligned}\hat{\boldsymbol{P}}&=\sqrt{\boldsymbol{P}}[\boldsymbol{U},\quad \boldsymbol{V}]\begin{bmatrix}\mathrm{diag}(\boldsymbol{w}'), & \boldsymbol{0}\\ \boldsymbol{0}, & \mathrm{diag}(\boldsymbol{w}'')\end{bmatrix}\begin{bmatrix}\boldsymbol{U}\sqrt{\boldsymbol{P}}\\ \boldsymbol{V}\sqrt{\boldsymbol{P}}\end{bmatrix}\\&=\sqrt{\boldsymbol{P}}[\mathrm{diag}(\boldsymbol{w}')\boldsymbol{U}^2+\mathrm{diag}(\boldsymbol{w}'')\boldsymbol{V}^2]\sqrt{\boldsymbol{P}}\\&=\sqrt{\boldsymbol{P}}\boldsymbol{I}\sqrt{\boldsymbol{P}}=\boldsymbol{P}\end{aligned}\tag{36}$$

because $\boldsymbol{w}' \odot \boldsymbol{u}^{\odot 2} + \boldsymbol{w}'' \odot \boldsymbol{v}^{\odot 2} = \mathbf{1}$ is the diagonal of $\text{diag}(\boldsymbol{w}')\boldsymbol{U}^2 + \text{diag}(\boldsymbol{w}'')\boldsymbol{V}^2$. We see that the sample covariance matrix equals the actual covariance matrix. Defining $\hat{\check{\boldsymbol{S}}} \in \mathbb{R}^n$ as a vector containing the diagonal components of the sample skewness tensor such that

$$\hat{\check{\boldsymbol{S}}} = [\hat{\boldsymbol{S}}_{111}, \hat{\boldsymbol{S}}_{222}, \ldots, \hat{\boldsymbol{S}}_{nnn}]^T,$$

we can evaluate the diagonal components of the sample skewness tensor as

$$\hat{\check{\boldsymbol{S}}} = ([-\sqrt{\boldsymbol{P}}\boldsymbol{U}, \quad \sqrt{\boldsymbol{P}}\boldsymbol{V}])^{\odot 3}[\boldsymbol{w}'^T, \quad \boldsymbol{w}''^T]^T \tag{37}$$

$$= [-\sqrt{\boldsymbol{P}}^{\odot 3}\boldsymbol{U}^{\odot 3}, \quad \sqrt{\boldsymbol{P}}^{\odot 3}\boldsymbol{V}^{\odot 3}][\boldsymbol{w}'^T, \quad \boldsymbol{w}''^T]^T \tag{38}$$

$$= \sqrt{\boldsymbol{P}}^{\odot 3}[-\boldsymbol{w}' \odot \boldsymbol{u}^{\odot 3} + \boldsymbol{w}'' \odot \boldsymbol{v}^{\odot 3}]$$

$$= \sqrt{\boldsymbol{P}}^{\odot 3}(\sqrt{\boldsymbol{P}}^{\odot 3})^{-1}\check{\boldsymbol{S}} = \check{\boldsymbol{S}}. \tag{39}$$

We see that our sigma points accurately match the diagonal components of the skewness tensor. Finally, defining $\hat{\check{\boldsymbol{K}}} \in \mathbb{R}^n$ as a vector containing the diagonal components of the sample kurtosis tensor such that

$$\hat{\check{\boldsymbol{K}}} = [\hat{\boldsymbol{K}}_{1111}, \hat{\boldsymbol{K}}_{2222}, \ldots, \hat{\boldsymbol{K}}_{nnnn}]^T,$$

we can evaluate the diagonal components of the sample kurtosis tensor as

$$\hat{\check{\boldsymbol{K}}} = ([-\sqrt{\boldsymbol{P}}\boldsymbol{U}, \quad \sqrt{\boldsymbol{P}}\boldsymbol{V}])^{\odot 4}[\boldsymbol{w}'^T \quad \boldsymbol{w}''^T]^T \tag{40}$$

$$= \sqrt{\boldsymbol{P}}^{\odot 4}[\boldsymbol{w}' \odot \boldsymbol{u}^{\odot 4} + \boldsymbol{w}'' \odot \boldsymbol{v}^{\odot 4}]$$

$$= \sqrt{\boldsymbol{P}}^{\odot 4}(\sqrt{\boldsymbol{P}}^{\odot 4})^{-1}\check{\boldsymbol{K}} = \check{\boldsymbol{K}}. \tag{41}$$

We see that our sigma points accurately match the diagonal components of the kurtosis tensor. ■

Theorem 1 shows that our sigma points in Algorithm 1 can accurately approximate the mean and covariance of any random vector, as well as the diagonal components of the skewness and kurtosis tensors—this makes it applicable to a wide variety of applications.

V. CONSTRAINED SIGMA POINTS

Noting that several physical systems require some constraints on their states or parameters, we show how our sigma points can be constrained while at least maintaining second-order accuracy.

We require the sigma points to be constrained such that

$$\boldsymbol{a} < \boldsymbol{\chi}_{[i]} < \boldsymbol{b} \quad \text{for } i \in \{1, \ldots, 2n+1\},$$

where $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^n$ are the lower bounds and upper bounds, respectively.

Assumption 2. The mean $\bar{\boldsymbol{x}}$ is within the bounds, such that $\boldsymbol{a} < \bar{\boldsymbol{x}} < \boldsymbol{b}$.

We note that our sigma points of Algorithm 1 can violate some state constraints despite being able to accurately capture the mean and covariance of a random vector, as well as the diagonal components of its skewness and kurtosis tensors. This might make them inapplicable in situations or models that only permit constrained values. For example, in applications that assume a Poisson distribution for the states, such as count data, the states are usually positive by default and can never be negative. When our sigma point of Algorithm 1 is applied, the positive constraint can be violated. We demonstrate this using the following example:

Example 1. We generate sigma points for an independent Poisson random vector x such that

$$\bar{\boldsymbol{x}} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1 \end{bmatrix}, \quad \check{\boldsymbol{S}} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}, \quad \check{\boldsymbol{K}} = \begin{bmatrix} 8.25 \\ 4 \end{bmatrix},$$

where $\bar{\boldsymbol{x}}$ is the mean, P is the covariance matrix, and $\check{\boldsymbol{S}}$ and $\check{\boldsymbol{K}}$ are vectors containing the diagonal components of the skewness tensor and kurtosis tensor, respectively. Using Algorithm 1, we see that $\boldsymbol{w}_1 = 0.3333$, $\boldsymbol{w}_2 = 0.2049$, $\boldsymbol{w}_3 = 0.2129$, $\boldsymbol{w}_4 = 0.1284$, $\boldsymbol{w}_5 = 0.1204$, $\boldsymbol{u}_1 = 1.3713$, $\boldsymbol{u}_2 = 1.3028$, $\boldsymbol{v}_1 = 2.1878$, and $\boldsymbol{v}_2 = 2.3028$. The $2n+1$ sigma points in matrix form is

$$\boldsymbol{\chi} = \begin{bmatrix} 1.5, & -0.1794 & 1.5 & 4.1794 & 1.5 \\ 1 & 1 & -0.3028 & 1 & 3.3028 \end{bmatrix}.$$

The sample statistics are

$$\hat{\bar{\boldsymbol{x}}} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}, \quad \hat{\boldsymbol{P}} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1 \end{bmatrix}, \quad \hat{\check{\boldsymbol{S}}} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}, \quad \hat{\check{\boldsymbol{K}}} = \begin{bmatrix} 8.25 \\ 4 \end{bmatrix}.$$

We see from Example 1 that despite the accuracy of the sample statistics, the sigma points $\boldsymbol{\chi}_{[2]}$ and $\boldsymbol{\chi}_{[3]}$ both had a negative value which do not satisfy the non-negativity of Poisson draws.

Corollary 1. Enforcing the constraint $\boldsymbol{a} < \boldsymbol{\chi}_{[i]} < \boldsymbol{b}$ for $i \in \{2, \ldots, 2n+1\}$ yields

(1) accurate capture of the mean, covariance matrix, and the diagonal components of the skewness tensor when $i \in \{2, \ldots, n+1\}$.

(2) accurate capture of the mean and covariance matrix when $i \in \{n+2, \ldots, 2n+1\}$.

Proof. Constraining $\boldsymbol{\chi}_{[i]}$ will require redefining $\boldsymbol{u}$ when $i \in \{2, \ldots, n+1\}$ and $\boldsymbol{v}$ when $i \in \{n+2, \ldots, 2n+1\}$. Theorem 1 establishes that violating (31) ensures inaccurate approximation of the diagonal components of the kurtosis tensor. Theorem 1 also establishes that violating (28) ensures inaccurate approximation of the diagonal components of the skewness and kurtosis tensors. ■

To enforce constraints on the sigma points, we introduce a *slack parameter* $\theta \in (0, \ldots, 1)$, which is a user selected constant. Using θ, we now redefine the free parameters $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ for $i \in \{1, \ldots, n\}$ as

$$\boldsymbol{u}_i = \theta\big[\min\big|(\bar{\boldsymbol{x}} - \boldsymbol{a}) \oslash \sqrt{P_{[i]}}\big|\big] \quad \text{if } \boldsymbol{\chi}_{[i+1]} < \boldsymbol{a},$$

$$\boldsymbol{v}_i = \theta\big[\min\big|(\boldsymbol{b} - \bar{\boldsymbol{x}}) \oslash \sqrt{P_{[i]}}\big|\big] \quad \text{if } \boldsymbol{\chi}_{[i+n+1]} > \boldsymbol{b},$$

where the sigma points get closer to their constraints as $\theta \to 1$. We note that the equations for $\boldsymbol{w}'$ and $\boldsymbol{w}''$ are unchanged.

We note that enforcing constrains on the sigma points results in a loss of accuracy in capturing the diagonal components of at least the kurtosis tensor. The constrained sigma point algorithm is given in Algorithm 2. We now show a benefit of Algorithm 2 in the following example:

Example 2. Using Algorithm 2 to generate positively constrained sigma points for the Poisson random vector, we select $\theta = 0.9$. We see that $\boldsymbol{w}_1 = -0.0576$, $\boldsymbol{w}_2 = 0.3003$, $\boldsymbol{w}_3 = 0.3968$, $\boldsymbol{w}_4 = 0.1725$, $\boldsymbol{w}_5 = 0.188$, $\boldsymbol{u}_1 = 1.1023$, $\boldsymbol{u}_2 = 0.9$,

ALGORITHM 2. Constrained sigma points for the generalized unscented transform.

Implement Algorithm 1
if $\boldsymbol{\chi}_{[i+1]} < \boldsymbol{a}$ *for* $i \in \{1, \ldots, 2n\}$ **then**
  if $i \leqslant n$ **then**
    $\boldsymbol{u}_i = \theta[\min\ |(\bar{\boldsymbol{x}} - \boldsymbol{a}) \oslash \sqrt{P_{[i]}}|]$
  end
  if $i > n$ **then**
    $\boldsymbol{v}_{i-n} = \theta[\min |(\boldsymbol{a} - \bar{\boldsymbol{x}}) \oslash \sqrt{P_{[i-n]}}|]$
  end
end
Repeat steps 3 and 4 of Algorithm 1 if $\boldsymbol{v}$ *was not redefined, otherwise repeat only step 4 of Algorithm 1*;
If $\boldsymbol{\chi}_{[i+1]} > \boldsymbol{b}$ for $i \in \{1, \ldots, 2n\}$ **then**
  if $i \leqslant n$ **then**
    $\boldsymbol{u}_i = \theta[\min |(\bar{\boldsymbol{x}} - \boldsymbol{b}) \oslash \sqrt{P_{[i]}}|]$
  end
  if $i > n$ **then**
    $\boldsymbol{v}_{i-n} = \theta[\min |(\boldsymbol{b} - \bar{\boldsymbol{x}}) \oslash \sqrt{P_{[i-n]}}|]$
  end
end
Repeat steps 3, 4, and 5 of Algorithm 1 if $\boldsymbol{v}$ *was not redefined, otherwise repeat only steps 4 and 5 of Algorithm 1*;
Note: $\theta \in (0, \ldots, 1)$ is a user-defined constant. The sigma points get closer to their constraints as $\theta \to 1$.

$\boldsymbol{v}_1 = 1.9188$, and $\boldsymbol{v}_2 = 1.9$. The $2n + 1$ positive sigma points in matrix form are

$$\boldsymbol{\chi} = \begin{bmatrix} 1.5 & 0.15 & 1.5 & 3.85 & 1.5 \\ 1 & 1 & 0.1 & 1 & 2.9 \end{bmatrix},$$

while the corresponding sample statistics are

$$\hat{\bar{\boldsymbol{x}}} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}, \quad \hat{\boldsymbol{P}} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1 \end{bmatrix}, \quad \hat{\check{\boldsymbol{S}}} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}, \quad \hat{\check{\boldsymbol{K}}} = \begin{bmatrix} 6.2587 \\ 2.7100 \end{bmatrix}.$$

We see from Example 2 that using Algorithm 2 ensures that the sigma points are always positive while ensuring accuracy in approximating the true mean and covariance of a random vector, as well as capturing the diagonal components of the skewness tensor. However, the ability to exactly capture the diagonal components of the kurtosis tensor is lost. A graphical representation of Examples 1 and 2 is shown in Fig. 3 where we plot the sigma points and the covariance.

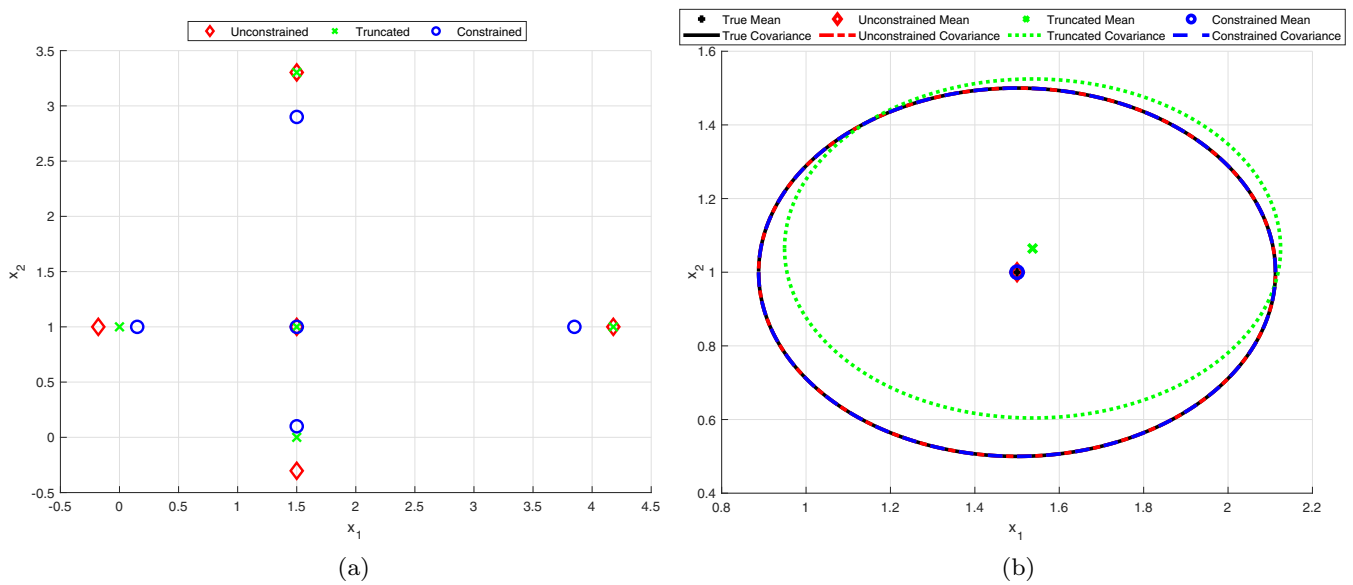
VI. PROPAGATION OF MEANS AND COVARIANCES OF NONLINEAR TRANSFORMATIONS

We analyze the performance of our new sigma point algorithm when they undergo nonlinear transformations. We show how linearization approximations, via Taylor series expansion of a nonlinear transformation of a random vector $\boldsymbol{x}$ evaluated about its mean $\bar{\boldsymbol{x}}$, introduce errors in the propagation of means and covariances. In Appendix A, we evaluated the true mean and true covariance of a random vector. In Appendix A, we evaluated the approximated mean and approximated covariance. We see that, although errors are introduced beyond the second order when approximating a nonlinear transformation of a random vector, these errors are minimized because of our ability to match the diagonal components of the skewness and kurtosis tensors. We also see that errors are introduced beyond the third order when the random vector is independent. We note that the nonlinear transformation $\boldsymbol{y} \in \mathbb{R}^n$ is given by

$$\boldsymbol{y} = \boldsymbol{\lambda}(\boldsymbol{x}), \tag{42}$$

where $\mathbb{E}[\boldsymbol{x}] = \bar{\boldsymbol{x}}$. Given $\bar{\boldsymbol{x}}$, $\boldsymbol{P}$, $\check{\boldsymbol{S}}$, and $\check{\boldsymbol{K}}$, we evaluate the sample mean and covariance of the nonlinear transformation of (42) using Algorithm 1.

For our comparison, we use the scaled unscented transform of Ref. [42], which is denoted as UT for the remainder of this paper, and the higher order sigma point unscented transform (HOSPUT) of [34]. The scaling of the UT was selected to



(a) (b)

FIG. 3. (a) Locations of sigma points for the unconstrained (Algorithm 1), truncated, and constrained (Algorithm 2) sigma points. (b) Mean and covariance of the unconstrained (Algorithm 1), truncated, and constrained (Algorithm 2) sigma points.

TABLE II. Percentage error in propagating $y = 3x + 2x^2$.

	x	GenUT	UT	MC	HOSPUT
Mean	$\mathcal{N}(1, 4)$	0	0	0.015	0
	$E(2)$	0	0	0.069	0
	$G(1, 2)$	0	0	0.452	0
	$W(1, 2)$	0	0	0.005	0
	$R(1)$	0	0	0.097	0
	$BE(3, 4)$	0	0	0.063	0
	$B(3, 0.3)$	0	0	0.457	0
	$P(2)$	0	0	0.270	0
	$GE(0.5)$	0	0	1.251	0
	$NB(4, 0.67)$	0	0	0.668	0
Covariance	$\mathcal{N}(1, 4)$	0	0	0.029	0
	$E(2)$	0	49.057	0.249	0
	$G(1, 2)$	0	64	1.889	0
	$W(1, 2)$	0	15.003	0.310	0
	$R(1)$	0	16.815	0.381	0
	$BE(3, 4)$	0	2.307	0.613	0
	$B(3, 0.3)$	0	16.380	0.359	0
	$P(2)$	0	25.946	1.061	0
	$GE(0.5)$	0	67.662	1.036	0
	$NB(4, 0.67)$	0	43.224	2.356	0

TABLE III. Percentage error in propagating $y = \sin(x)$.

	x	GenUT	UT	MC	HOSPUT
Mean	$\mathcal{N}(1.57, 0.1)$	0.001	0.001	0.012	0.001
	$E(2)$	0.219	5.788	0.110	0.219
	$G(0.5, 0.5)$	0.312	6.964	0.050	0.312
	$W(1, 2)$	0.017	0.831	0.029	0.017
	$R(1)$	0.049	0.912	0.007	0.049
	$BE(3, 4)$	0	0.038	0.037	0
	$B(3, 0.3)$	0.158	4.814	0.046	0.158
	$P(0.1)$	0.275	18.305	0.531	0.275
	$GE(0.7)$	2.416	32.906	0.138	2.416
	$NB(0.4, 0.67)$	0.176	44.172	0.383	0.176
Covariance	$\mathcal{N}(1.57, 0.1)$	5.026	5.026	0.444	5.026
	$E(2)$	23.499	72.557	0.213	23.499
	$G(0.5, 0.5)$	20.749	61.391	0.372	20.749
	$W(1, 2)$	4.862	31.760	0.043	4.862
	$R(1)$	12.158	50.678	0.531	12.158
	$BE(3, 4)$	0.031	0.940	0.225	0.031
	$B(3, 0.3)$	11.033	24.806	0.060	11.033
	$P(0.1)$	6.646	45.895	0.461	6.646
	$GE(0.7)$	12.074	87.637	0.070	12.074
	$NB(0.4, 0.67)$	39.068	135.783	0.366	39.068

match a Gaussian distribution. We do not compare against the sigma points in Refs. [35–38] because they either use a Gaussian assumption, a closed skew normal distribution, or more than $2n + 1$ sigma points.

A. Case study 1: Transformation of random variables

Defining x as a random variable that can follow any of the probability distributions given in the Table I, we evaluate the sample mean and covariance of two nonlinear transformations: a quadratic function of the random variable $y = 3x + 2x^2$, and a trigonometric function of the random variable $y = \sin(x)$. We also use 10^5 Monte Carlo draws from the different probability distributions. The true mean and covariance of the quadratic function can be easily evaluated using the raw moments of x up to its fourth order. The true mean and covariance of the trigonometric function can be evaluated using their characteristic functions. A comparison between the accuracy of the GenUT, UT, 10^5 Monte Carlo draws, and HOSPUT in approximating the true mean and true covariance of the nonlinear transformations for the different probability distributions is shown in Tables II and III (the abbreviations are defined in Table I).

For the quadratic function, we see that both the GenUT and HOSPUT gave an exact approximation of the true mean and true covariance for all the probability distributions. This is because the GenUT and HOSPUT are accurate up to the fourth-order moments when the random variable x has a dimension of 1. Although the 10^5 Monte Carlo draws gave relatively good approximations, they were not as accurate as the GenUT.

For the trigonometric function, we see that the GenUT, HOSPUT, and UT were unable to give exact approximations of the true mean and true covariance in most cases because the Taylor series expansion of $\boldsymbol{\lambda}(\boldsymbol{x})$ has terms beyond the fourth order. The GenUT and HOSPUT were more accurate than the UT for all the non-Gaussian probability distributions because they are both accurate up to the fourth order while the UT is accurate up to the second order. The 10^5 Monte Carlo draws sometimes gave better accuracy than the GenUT because of the random nature of its draws. A box plot of the accuracy of the GenUT, UT, and several Monte Carlo draws of different sizes is shown in Fig. 4 for the trigonometric function. We do not include the HOSPUT because it gives the same performance as the GenUT when a single random variable is transformed. We see that a significant number of Monte Carlo draws is needed to achieve the accuracy of the GenUT when approximating the mean. A significant number of Monte Carlo draws gives better accuracy in approximating the variance.

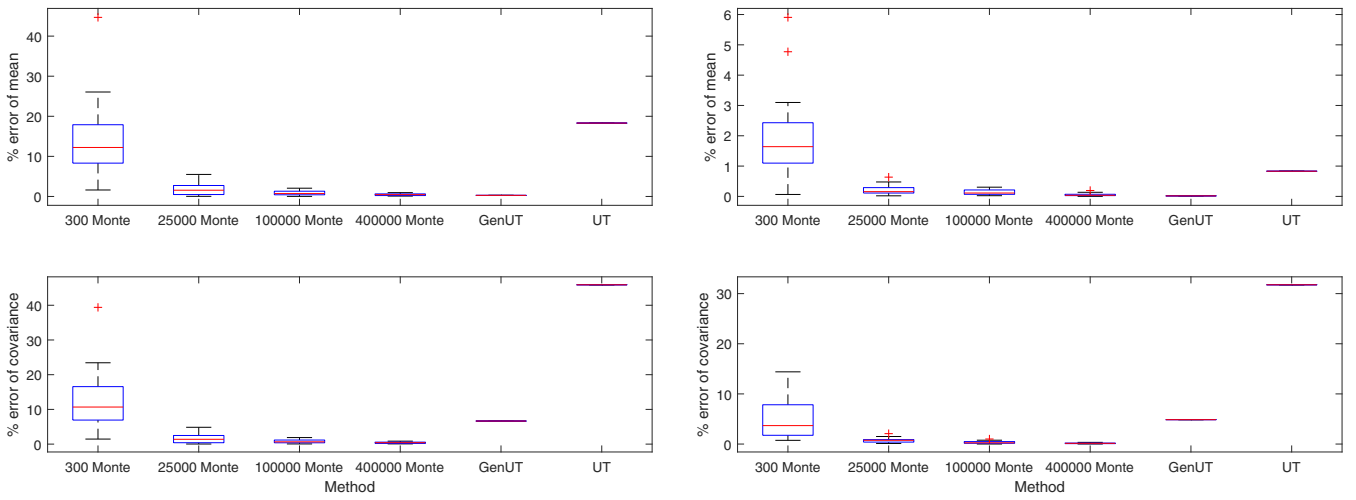
B. Case study 2: Transformation of a correlated random vector

We consider the nonlinear transformation of a correlated random vector from a bivariate Gamma distribution such that

$$\boldsymbol{y} = \begin{bmatrix} \sin(\boldsymbol{x}_1) \\ \cos(\boldsymbol{x}_2) \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} = \begin{bmatrix} \text{Gamma}(0.1, 0.3) \\ \text{Gamma}(2, 0.3) \end{bmatrix}, \tag{43}$$

where $\boldsymbol{x}_1 = \boldsymbol{\lambda}_1$ and $\boldsymbol{x}_2 = \boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2$ have a Pearson's correlation coefficient of $\rho = 0.22$ and cross covariance of $\mathbb{E}[(\boldsymbol{x}_1 - \bar{\boldsymbol{x}}_1)(\boldsymbol{x}_2 - \bar{\boldsymbol{x}}_2)] = 0.009$. We examine the performance of the GenUT, HOSPUT, and UT in approximating the true mean and covariance of $\boldsymbol{y}$. We calculate the true mean using their characteristic function, while we approximate the true covariance, skewness, and kurtosis of $\boldsymbol{y}$ using 10^7 Monte Carlo draws. The percentage approximation errors are

$$\underbrace{\begin{bmatrix} \mathbf{0.1} \\ \mathbf{0.11} \end{bmatrix}}_{\substack{\text{GenUT} \\ \% \text{ mean error}}}, \quad \underbrace{\begin{bmatrix} 2.86 \\ 1.67 \end{bmatrix}}_{\substack{\text{UT} \\ \% \text{ mean error}}}, \quad \underbrace{\begin{bmatrix} 2.19 \\ 1.18 \end{bmatrix}}_{\substack{\text{HOSPUT} \\ \% \text{ mean error}}},$$



(a) (b)

FIG. 4. (a) Moments of $y = \sin(x)$ when x is a Poisson random variable. (b) Moments of $y = \sin(x)$ when x is a Weibull random variable.

$$\underbrace{\begin{bmatrix} \mathbf{2.65} & \mathbf{10.1} \\ \mathbf{10.1} & \mathbf{8.19} \end{bmatrix}}_{\substack{\text{GenUT} \\ \%\text{ covariance error}}}, \quad \underbrace{\begin{bmatrix} 18.66 & 11.46 \\ 11.46 & 31.08 \end{bmatrix}}_{\substack{\text{UT} \\ \%\text{ covariance error}}}, \quad \underbrace{\begin{bmatrix} 17.2 & 10.84 \\ 10.84 & 9.42 \end{bmatrix}}_{\substack{\text{HOSPUT} \\ \%\text{ covariance error}}},$$

$$\underbrace{\begin{bmatrix} \mathbf{3.4} \\ \mathbf{37.33} \end{bmatrix}}_{\substack{\text{GenUT} \\ \%\text{ skewness error}}}, \quad \underbrace{\begin{bmatrix} 100.2 \\ \mathbf{37.33} \end{bmatrix}}_{\substack{\text{UT} \\ \%\text{ skewness error}}}, \quad \underbrace{\begin{bmatrix} 66.55 \\ 41.03 \end{bmatrix}}_{\substack{\text{HOSPUT} \\ \%\text{ skewness error}}},$$

$$\underbrace{\begin{bmatrix} \mathbf{47.63} \\ \mathbf{19.58} \end{bmatrix}}_{\substack{\text{GenUT} \\ \%\text{ kurtosis error}}}, \quad \underbrace{\begin{bmatrix} 88.92 \\ 79.6 \end{bmatrix}}_{\substack{\text{UT} \\ \%\text{ kurtosis error}}}, \quad \underbrace{\begin{bmatrix} 78.59 \\ 24.29 \end{bmatrix}}_{\substack{\text{HOSPUT} \\ \%\text{ kurtosis error}}},$$

where the lowest approximation errors are given in bold font. We see that the GenUT gave the least approximation error for the mean, covariance, skewness, and kutrosis. The UT gave the worst performance because it was unable to account for the non-Gaussian distributed nature of the random vector $\boldsymbol{x}$. The HOSPUT performed worse than the GenUT because, when the problem dimension exceeds 1, it is only able to match the average values of the diagonal elements of the skewness and kurtosis tensors.

C. Case study 3: Infectious disease models

We consider an SIR (susceptible-infectious-recovered) infectious disease model given by the difference equation [1]

$$\begin{aligned} S_{k+1} &= S_k - \frac{\beta S_k I_k}{N}, \\ I_{k+1} &= I_k + \frac{\beta S_k I_k}{N} - \gamma I_k, \\ R_{k+1} &= R_k + \gamma I_k, \end{aligned} \tag{44}$$

where β is the infection rate, γ is the recovery rate, and $N = S_k + I_k + R_k$. Using the conservation principle $S + I + R = N$, we reduce the model of (44) to

$$\begin{aligned} I_{k+1} &= I_k + \beta(N - I_k - R_k)\frac{I_k}{N} - \gamma I_k, \\ R_{k+1} &= R_k + \gamma I_k. \end{aligned}$$

We note that by defining a correlated random vector $\boldsymbol{x} = \text{Poisson}[I_k \quad R_k]^T$ from a bivariate Poisson distribution with Pearson's correlation coefficient $\rho = 0.045$ and cross covariance $\mathbb{E}[(\boldsymbol{x}_1 - \bar{\boldsymbol{x}}_1)(\boldsymbol{x}_2 - \bar{\boldsymbol{x}}_2)] = 0.2$, we can rewrite the above equation as

$$\begin{bmatrix} I_{k+1} \\ R_{k+1} \end{bmatrix} = \begin{bmatrix} I_k \\ R_k \end{bmatrix} + \begin{bmatrix} \beta(N - \boldsymbol{x}_1 - \boldsymbol{x}_2)\frac{\boldsymbol{x}_1}{N} \\ \gamma \boldsymbol{x}_1 \end{bmatrix}, \tag{45}$$

where $\boldsymbol{x}_i$ is the ith element of the vector $\boldsymbol{x}$.

We examine the performance of the GenUT, HOSPUT, and UT in approximating the true mean, covariance, skewness, and kurtosis of (45). We use the parameters $I_k = 10$, $R_k = 2$, $\beta = 1.5$, $\gamma = 0.3$, and $N = 100$. The percentage approximation errors are

$$\underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\substack{\text{GenUT} \\ \%\text{ mean error}}}, \quad \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\substack{\text{UT} \\ \%\text{ mean error}}}, \quad \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\substack{\text{HOSPUT} \\ \%\text{ mean error}}},$$

$$\underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0.03} \\ \mathbf{0.03} & \mathbf{0.02} \end{bmatrix}}_{\substack{\text{GenUT} \\ \%\text{ covariance error}}}, \quad \underbrace{\begin{bmatrix} 2.64 & 1.36 \\ 1.36 & \mathbf{0.02} \end{bmatrix}}_{\substack{\text{UT} \\ \%\text{ covariance error}}}, \quad \underbrace{\begin{bmatrix} 0.27 & 0.10 \\ 0.1 & \mathbf{0.02} \end{bmatrix}}_{\substack{\text{HOSPUT} \\ \%\text{ covariance error}}},$$

$$\underbrace{\begin{bmatrix} \mathbf{13.39} \\ \mathbf{0.01} \end{bmatrix}}_{\substack{\text{GenUT} \\ \%\text{ skewness error}}}, \quad \underbrace{\begin{bmatrix} 432.40 \\ 100 \end{bmatrix}}_{\substack{\text{UT} \\ \%\text{ skewness error}}}, \quad \underbrace{\begin{bmatrix} 28.35 \\ 10.15 \end{bmatrix}}_{\substack{\text{HOSPUT} \\ \%\text{ skewness error}}},$$

$$\underbrace{\begin{bmatrix} 4.42 \\ \mathbf{0.07} \end{bmatrix}}_{\substack{\text{GenUT} \\ \%\text{ kurtosis error}}}, \quad \underbrace{\begin{bmatrix} 10.15 \\ 3.19 \end{bmatrix}}_{\substack{\text{UT} \\ \%\text{ kurtosis error}}}, \quad \underbrace{\begin{bmatrix} \mathbf{4.03} \\ 0.57 \end{bmatrix}}_{\substack{\text{HOSPUT} \\ \%\text{ kurtosis error}}},$$

where the lowest approximation errors are given in bold font. We see that the GenUT gave the least approximation error for the mean, covariance, and skewness. The HOSPUT gave the lowest approximation error for the first component of the kurtosis. This can be explained by the fact that the skewness and kurtosis tensors depend on higher-order moments that are not accurately captured by the GenUT, UT, or HOSPUT.

A study on the accuracy of approximating the kurtosis of a nonlinear transformation of a random vector is beyond the scope of this paper.

VII. CONCLUSION

In this paper, we have developed the generalized unscented transform (GenUT) that is capable of adapting to the unique statistics of an arbitrarily distributed random variable. The GenUT forms a critical step towards realization of more accurate estimation and tracking of natural systems found in ecology, biology, weather, and physics, where nonlinearities tend to be omni-present and measurements of natural phenomena reveal non-Gaussian statistics.

As the scale of measured systems to track, predict, and control increases, the ability of methods based on linearization and simplified Gaussian statistics deteriorates, eventually becoming impractical to implement as the estimation error grows. Here we demonstrated the ability of GenUT to match the diagonal elements of the skewness and kurtosis tensors of most random vectors using $2n+1$ sigma points. This allows it to be more accurate than unscented transforms that were constrained to a specific probability distribution—especially where the physics are nonlinear. For example, in our case study in Sec. VI C we demonstrate that the GenUT extends the accuracy of a nonlinear infectious disease model of case count data utilizing minimal $2n+1$ sample points drawn from a multivariate distribution when compared to other transforms. However, we note that the presented tests are far from representing a realistic application of the GenUT's ability to properly propagate higher moments. Such applications, particularly for filtering problems, will necessitate the development of more advanced prediction and update equations, which is a direction for future research.

The error of the various transforms were analyzed using a Taylor series of the nonlinear function, a standard approach in the literature [24,28]. This analysis requires the assumption that the probability distribution is concentrated near its mean so that the higher-order moments have a negligible contribution to the Taylor series than the leading order moments. Of course, each additional moment of the input distribution that can be matched will improve this approximation. However, empirically we observe that matching additional moments improves performance even when the distribution is not concentrated near its mean. This case can be analyzed by considering the unscented transform (or GenUT) as a quadrature rule, and the error can be bounded in terms of polynomial approximation error bounds [38]. While such an approach is promising, it is particularly challenging in the case of the GenUT since matching the diagonal entries of the skewness and kurtosis tensors is equivalent to fitting a restricted class quartic polynomial. Error bounds for these restricted classes of polynomials are not readily available, so deriving these forms of error bounds is a direction for future research.

Although the GenUT gave better accuracy because it can capture the diagonal components of the skewness and kurtosis tensors with $2n+1$ sigma points, its inability to match the off-diagonal components implies that it is unable to match more terms in the Taylor series expansion, which can limit its accuracy. Trying to match the entire off-diagonal components of the skewness and kurtosis tensors will require looking beyond $2n+1$ sigma points, which is beyond the scope of this paper.

In terms of ease of implementation, we demonstrated that the GenUT uses $2n+1$ sigma points, like the unscented transform originally developed in Ref. [28]. When compared against unscented transforms that employ more than $2n+1$ sigma points, such as the one in Ref. [38], the GenUT is characterized by a lower computational cost (due to its lower number of sigma points) that scales linearly with the problem dimension. In scenarios where the problem dimension is relatively small and/or where the computational capacity is high, methods such as the one in Ref. [38] that uses more than $2n+1$ sigma points might be preferable due to its increased accuracy in approximating the entire skewness and kurtosis tensors.

In terms of performance, the GenUT and unscented transforms that use $2n+1$ sigma points developed under the Gaussian assumption give the same performance when the random variable is Gaussian distributed. However, when the random variable or random vector is not Gaussian distributed, the GenUT gives better accuracy in the propagation of means and covariances. Additionally, we also showed that the GenUT formulation makes it easy to analytically enforce constraints on the sigma points while still guaranteeing at least a second-order accuracy, which makes it appealing in models that permit only constrained values for random variables or parameters.

For uncertainty quantification, estimation, or prediction applications, when compared with existing unscented transforms, the GenUT gives the most accuracy that can be achieved by employing $2n+1$ sigma points. This accuracy will have increased significant consequences if the nonlinearities are strong and the problem dimension is large. The GenUT can be applied to any filter that uses linear or nonlinear transformations of random variables.

ACKNOWLEDGMENTS

This work was supported by NIH Director's Transformative Award No. 1R01AI145057, and from the National Science Foundation Grants No. DMS-1723175, No. DMS-1854204, and No. DMS-2006808.

DATA AVAILABILITY

The MATLAB® source code used to generate the results in this paper is available online [43].

APPENDIX A: TRUE MEAN AND COVARIANCE OF NONLINEAR TRANSFORMATIONS

We derive analytical expressions for the true mean and covariance when we take the Taylor series expansion of the nonlinear function $\boldsymbol{y} = \boldsymbol{\lambda}(\boldsymbol{x})$, where $\boldsymbol{x}$ is a random vector.

1. True mean of the nonlinear transformation

Applying a Taylor series expansion around $\bar{\boldsymbol{x}}$, where $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}}$, we write the true mean of $\boldsymbol{y}$ as

$$\begin{aligned}\bar{\boldsymbol{y}} &= \mathbb{E}[\boldsymbol{\lambda}(\boldsymbol{x})] \\ &= \boldsymbol{\lambda}(\bar{\boldsymbol{x}}) + \mathbb{E}\left[D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda} + \frac{D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{2!} + \frac{D^3_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{3!} + \frac{D^4_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{4!} + \cdots\right], \quad \text{(A1)}\end{aligned}$$

where $D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}$ is the total differential of $\boldsymbol{\lambda}(\boldsymbol{x})$ when perturbed around a nominal value $\bar{\boldsymbol{x}}$ by $\tilde{\boldsymbol{x}}$. We note that

$$D^k_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda} = \left(\sum_{i=1}^{n} \tilde{\boldsymbol{x}}_i \frac{\partial}{\partial \boldsymbol{x}_i}\right)^k \boldsymbol{\lambda}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}. \quad \text{(A2)}$$

Using (A2), we can evaluate the true mean of (A1) as

$$\begin{aligned}\bar{\boldsymbol{y}} = \boldsymbol{\lambda}(\bar{\boldsymbol{x}}) + \Bigg[&\sum_{i,j=1}^{n} \frac{\boldsymbol{P}_{ij}}{2!}\frac{\partial^2\boldsymbol{\lambda}}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j} + \sum_{i,j,k=1}^{n} \frac{\boldsymbol{S}_{ijk}}{3!}\frac{\partial^3\boldsymbol{\lambda}}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j \partial \boldsymbol{x}_k} \\ &+ \sum_{i,j,k,l=1}^{n} \frac{\boldsymbol{K}_{ijkl}}{4!}\frac{\partial^4\boldsymbol{\lambda}}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j \partial \boldsymbol{x}_k \partial \boldsymbol{x}_l}\Bigg]_{\boldsymbol{x}=\bar{\boldsymbol{x}}} + \cdots, \quad \text{(A3)}\end{aligned}$$

where $\boldsymbol{P}_{ij} = \mathbb{E}[\tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_j]$, $\boldsymbol{S}_{ijk} = \mathbb{E}[\tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_j\tilde{\boldsymbol{x}}_k]$, and $\boldsymbol{K}_{ijkl} = \mathbb{E}[\tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_j\tilde{\boldsymbol{x}}_k\tilde{\boldsymbol{x}}_l]$.

2. True covariance of the nonlinear transformation

The true covariance of y is

$$\boldsymbol{P}_y = \mathbb{E}[(\boldsymbol{y} - \bar{\boldsymbol{y}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^T]. \quad \text{(A4)}$$

Evaluating the expression $\boldsymbol{y} - \bar{\boldsymbol{y}}$, we write

$$\boldsymbol{y} - \bar{\boldsymbol{y}} = D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda} + \frac{D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{2!} + \frac{D^3_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{3!} - \mathbb{E}\left[\frac{D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{2!} + \frac{D^3_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{3!}\right] + \cdots. \quad \text{(A5)}$$

Substituting (A5) into (A4) gives

$$\begin{aligned}\boldsymbol{P}_y = \mathbb{E}\Bigg[&D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\left(D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\right)^T + \frac{D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\left(D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\right)^T}{2!} + \frac{D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\left(D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\right)^T}{2!} \\ &+ \frac{D^3_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\left(D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\right)^T}{3!} + \frac{D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\left(D^3_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\right)^T}{3!} + \frac{D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\left(D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}\right)^T}{2!\times 2!}\Bigg] \\ &+ \mathbb{E}\left[\frac{D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{2!}\right]\mathbb{E}\left[\frac{D^2_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}}{2!}\right]^T + \cdots. \quad \text{(A6)}\end{aligned}$$

We note that we can write the first term in the above equation as

$$\mathbb{E}[D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda}(D_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda})^T] = \frac{\partial\boldsymbol{\lambda}}{\partial\boldsymbol{x}}\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \mathbb{E}[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T]\frac{\partial\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}}\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} = \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{\Lambda}^T. \quad \text{(A7)}$$

Using (A2) and (A7), we can rewrite the true covariance matrix of (A6) as

$$\begin{aligned}\boldsymbol{P}_y = \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{\Lambda}^T + \Bigg[&\sum_{i,j,k=1}^{n} \frac{\boldsymbol{S}_{ijk}}{2!}\left[\frac{\partial^2\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j}\frac{\partial\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_k} + \frac{\partial\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i}\frac{\partial^2\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k}\right] \\ &+ \sum_{i,j,k,l=1}^{n} \boldsymbol{K}_{ijkl}\left[\frac{1}{3!}\frac{\partial^3\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k}\frac{\partial\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_l} + \frac{1}{3!}\frac{\partial\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i}\frac{\partial^3\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k\partial\boldsymbol{x}_l} + \frac{1}{4}\frac{\partial^2\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j}\frac{\partial^2\boldsymbol{\lambda}^T}{\partial\boldsymbol{x}_k\partial\boldsymbol{x}_l}\right] \\ &+ \left[\sum_{i,j=1}^{n} \frac{\boldsymbol{P}_{ij}}{2}\frac{\partial^2\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j}\right][\cdots]^T\Bigg]_{\boldsymbol{x}=\bar{\boldsymbol{x}}} + \cdots, \quad \text{(A8)}\end{aligned}$$

where we have used the notation $\boldsymbol{x}\boldsymbol{x}^T = \boldsymbol{x}[\cdots]^T$.

APPENDIX B: APPROXIMATION OF MEANS AND COVARIANCE USING THE GENERALIZED UNSCENTED TRANSFORM

We analytically show the accuracy in capturing the true mean and true covariance of $\boldsymbol{y} = \boldsymbol{\lambda}(\boldsymbol{x})$ when using our $2n+1$ sigma points. We also show that our sigma point transformations give improved accuracy by capturing the diagonal components of the skewness and kurtosis tensors. Recalling that $\boldsymbol{\chi}_{[i]}$ is the ith column of $\boldsymbol{\chi}$, we define $\tilde{\boldsymbol{\chi}}_{[i]} = \boldsymbol{\chi}_{[i]} - \bar{\boldsymbol{x}}$. We also define the jth entry in the ith column of $\tilde{\boldsymbol{\chi}}$ as $\tilde{\boldsymbol{\chi}}_{ji} = (\boldsymbol{\chi}_{[i]} - \bar{\boldsymbol{x}})_j$. We note that

$$\sum_{i=1}^{2n+1} \boldsymbol{w}_i D^k_{\tilde{\boldsymbol{x}}}\boldsymbol{\lambda} = \sum_{i=1}^{2n+1} \boldsymbol{w}_i \left(\sum_{j=1}^{n} \tilde{\boldsymbol{x}}_j \frac{\partial}{\partial\boldsymbol{x}_j}\right)^k \boldsymbol{\lambda}(\boldsymbol{x})\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}. \quad \text{(B1)}$$

1. Approximation of the mean

The approximated mean is given as

$$\begin{aligned}\hat{\bar{\boldsymbol{y}}} &= \sum_{i=1}^{2n+1} \boldsymbol{w}_i \boldsymbol{\lambda}(\boldsymbol{\chi}_{[i]}) \\ &= \sum_{i=1}^{2n+1} \boldsymbol{w}_i \left[\boldsymbol{\lambda}(\bar{\boldsymbol{x}}) + D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda} + \frac{D^2_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}}{2!} + \frac{D^3_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}}{3!} + \cdots\right] \\ &= \boldsymbol{\lambda}(\bar{\boldsymbol{x}}) + \sum_{i=1}^{2n+1} \boldsymbol{w}_i \left[D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda} + \frac{D^2_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}}{2!} + \frac{D^3_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}}{3!} + \cdots\right].\end{aligned}$$

Using (B1), we can evaluate the above equation as

$$\begin{aligned}\hat{\bar{\boldsymbol{y}}} = \boldsymbol{\lambda}(\bar{\boldsymbol{x}}) + \Bigg[&\sum_{i,j=1}^{n} \frac{\boldsymbol{P}_{ij}}{2}\frac{\partial\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j} + \sum_{i,j,k=1}^{n} \frac{\hat{\boldsymbol{S}}_{ijk}}{3!}\frac{\partial^3\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k} \\ &+ \sum_{i,j,k,l=1}^{n} \frac{\hat{\boldsymbol{K}}_{ijkl}}{4!}\frac{\partial^4\boldsymbol{\lambda}}{\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j\partial\boldsymbol{x}_k\partial\boldsymbol{x}_l}\Bigg]_{\boldsymbol{x}=\bar{\boldsymbol{x}}} + \cdots, \quad \text{(B2)}\end{aligned}$$

where $\sum_{i=1}^{2n+1} \boldsymbol{w}_i\tilde{\boldsymbol{\chi}}_{ji}\tilde{\boldsymbol{\chi}}_{ki} = \boldsymbol{P}_{jk}$, $\sum_{i=1}^{2n+1} \boldsymbol{w}_i\tilde{\boldsymbol{\chi}}_{ji}\tilde{\boldsymbol{\chi}}_{ki}\tilde{\boldsymbol{\chi}}_{li} = \hat{\boldsymbol{S}}_{jkl}$, and $\sum_{i=1}^{2n+1} \boldsymbol{w}_i\tilde{\boldsymbol{\chi}}_{ji}\tilde{\boldsymbol{\chi}}_{ki}\tilde{\boldsymbol{\chi}}_{li}\tilde{\boldsymbol{\chi}}_{mi} = \hat{\boldsymbol{K}}_{jklm}$.

In the Sec. IV, we already showed that we can accurately capture the diagonal components of the skewness and kurtosis tensors because $\hat{\boldsymbol{S}}_{jkl} = \boldsymbol{S}_{jkl}$ whenever $j = k = l$ and $\hat{\boldsymbol{K}}_{jklm} = \boldsymbol{K}_{jklm}$ whenever $j = k = l = m$. Therefore, by comparing (B2) with the true mean of (A3), we can see that our sigma points improves on the accuracy of propagating the mean of a nonlinear transformation.

2. Approximation of the covariance

The approximated covariance can be evaluated using the expression

$$\boldsymbol{P}_u = \sum_{i=1}^{2n+1} \boldsymbol{w}_i [\boldsymbol{\mathcal{Y}}_{[i]} - \hat{\boldsymbol{y}}][\boldsymbol{\mathcal{Y}}_{[i]} - \hat{\boldsymbol{y}}]^T. \tag{B3}$$

Evaluating the expression $\boldsymbol{\mathcal{Y}}_{[i]} - \hat{\boldsymbol{y}}$, we write

$$\boldsymbol{\mathcal{Y}}_{[i]} - \hat{\boldsymbol{y}} = D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda} + \frac{D^2_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}}{2!} + \frac{D^3_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}}{3!} - \sum_{j=1}^{2n+1} \boldsymbol{w}_j \left[\frac{D^2_{\boldsymbol{\chi}_{[j]}}\boldsymbol{\lambda}}{2!} + \frac{D^3_{\boldsymbol{\chi}_{[j]}}\boldsymbol{\lambda}}{3!} \right] + \cdots. \tag{B4}$$

We can write

$$\begin{aligned} \sum_{i=1}^{2n+1} \boldsymbol{w}_i D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda} (D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda})^T &= \sum_{j,k=1}^{n} \sum_{i=1}^{2n+1} \boldsymbol{w}_i \tilde{\boldsymbol{\chi}}_{ji} \tilde{\boldsymbol{\chi}}_{ki} \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{x}_j} \frac{\partial \boldsymbol{\lambda}^T}{\partial \boldsymbol{x}_k}\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} = \sum_{j,k=1}^{n} \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{x}_j}\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \boldsymbol{P}_{jk} \frac{\partial \boldsymbol{\lambda}^T}{\partial \boldsymbol{x}_k}\bigg|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \\ &= \boldsymbol{\Lambda} \boldsymbol{P} \boldsymbol{\Lambda}^T. \end{aligned} \tag{B5}$$

Substituting (B4) into (B3) and multiplying out gives

$$\begin{aligned} \boldsymbol{P}_u = \sum_{i=1}^{2n+1} \boldsymbol{w}_i \Bigg[& D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda} \big(D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big)^T + \frac{D^2_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big(D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big)^T}{2!} + \frac{D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big(D^2_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big)^T}{2!} + \frac{D^3_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big(D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big)^T}{3!} \\ & + \frac{D_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big(D^3_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big)^T}{3!} + \frac{D^2_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big(D^2_{\tilde{\boldsymbol{\chi}}_{[i]}}\boldsymbol{\lambda}\big)^T}{2! \times 2!} \Bigg] + \left[\sum_{j=1}^{2n+1} w_j \frac{D^2_{\boldsymbol{\chi}_{[j]}}\boldsymbol{\lambda}}{2!} \right] [\cdots]^T + \cdots. \end{aligned} \tag{B6}$$

Using (B1) and (B5), we can rewrite the approximated covariance matrix of (B6) as

$$\begin{aligned} \boldsymbol{P}_u = \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{\Lambda}^T + \Bigg[& \sum_{i,j,k=1}^{n} \frac{\hat{\boldsymbol{S}}_{ijk}}{2!} \left[\frac{\partial^2 \boldsymbol{\lambda}}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j} \frac{\partial \boldsymbol{\lambda}^T}{\partial \boldsymbol{x}_k} + \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{x}_i} \frac{\partial^2 \boldsymbol{\lambda}^T}{\partial \boldsymbol{x}_j \boldsymbol{x}_k} \right] + \sum_{i,j,k,l=1}^{n} \hat{\boldsymbol{K}}_{ijkl} \left[\frac{1}{3!} \frac{\partial^3 \boldsymbol{\lambda}}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j \partial \boldsymbol{x}_k} \frac{\partial \boldsymbol{\lambda}^T}{\partial \boldsymbol{x}_l} \right. \\ & \left. + \frac{1}{3!} \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{x}_i} \frac{\partial^3 \boldsymbol{\lambda}^T}{\partial \boldsymbol{x}_j \partial \boldsymbol{x}_k \partial \boldsymbol{x}_l} + \frac{1}{4} \frac{\partial^2 \boldsymbol{\lambda}}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j} \frac{\partial^2 \boldsymbol{\lambda}^T}{\partial \boldsymbol{x}_k \partial \boldsymbol{x}_l} \right] + \left[\sum_{i,j=1}^{n} \frac{\boldsymbol{P}_{ij}}{2} \frac{\partial^2 \boldsymbol{\lambda}}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j} \right] [\cdots]^T \Bigg]_{\boldsymbol{x}=\bar{\boldsymbol{x}}} + \cdots. \end{aligned} \tag{B7}$$

Comparing (B7) with the true covariance of (A8), we can see that our sigma points improve on the accuracy of propagating the covariance of a nonlinear transformation because we are able to accurately capture the diagonal components of the skewness and kurtosis tensors.

[1] M. J. Keeling and P. Rohani, *Modeling Infectious Diseases in Humans and Animals* (Princeton University Press, Princeton, NJ, 2011).

[2] S. J. Schiff, *Neural Control Engineering: The Emerging Intersection Between Control Theory and Neuroscience* (MIT Press, Cambridge, 2012), p. 361.

[3] E. Kalnay, *Atmospheric Modeling, Data Assimilation and Predictability* (Cambridge University Press, Cambridge, 2002).

[4] J. A. Knauss and N. Garfield, *Introduction to Physical Oceanography*, 3rd ed. (Waveland Press Inc., Long Grove, IL, 2016).

[5] R. M. Anderson and R. M. May, *Infectious Diseases of Humans* (Oxford University Press, New York, NY, 1991).

[6] E. Simons, M. Ferrari, J. Fricks, K. Wannemuehler, A. Anand, A. Burton, and P. Strebel, Lancet **379**, 2173 (2012).

[7] S. Chen, J. Fricks, and M. J. Ferrari, J. R. Stat. Soc. C: Appl. Stat. **61**, 117 (2012).

[8] C. Bretó, D. He, E. L. Ionides, and A. A. King, Ann. Appl. Stat. **3**, 319 (2009).

[9] J. Shaman and A. Karspeck, Proc. Natl. Acad. Sci. USA **109**, 20425 (2012).

[10] T. K. Yamana, S. Kandula, and J. Shaman, J. R. Soc. Interface **13**, 20160410 (2016).

[11] D. Ndanguza, I. S. Mbalawata, H. Haario, and J. M. Tchuenche, Math. Comput. Simul. **142**, 113 (2017).

[12] D. Ebeigbe, T. Berry, S. J. Schiff, and T. Sauer, Phys. Rev. Res. **2**, 043028 (2020).

[13] R. Li, S. Pei, B. Chen, Y. Song, T. Zhang, W. Yang, and J. Shaman, Science **368**, 489 (2020).

[14] D. Simon, *Optimal State Estimation* (John Wiley & Sons Inc., Hoboken, NJ, 2006).

[15] R. Kandepu, L. Imsland, and B. A. Foss, *2008 16th Mediterranean Conference on Control and Automation* (IEEE, New York, 2008), pp. 1453–1458.

[16] R. E. Kalman, J. Basic Eng. **82**, 35 (1960).

[17] J. Uhlmann and S. J. Julier, J. Cienc. Ing. **14**, 21 (2022).

[18] R. Izanloo, S. A. Fakoorian, H. S. Yazdi, and D. J. Simon, *2016 Annual Conference on Information Science and Systems (CISS)*, Vol. 16 (IEEE, New York, 2016), pp. 500–505.

[19] F. Gustafsson and G. Hendeby, IEEE Trans. Signal Process. **60**, 545 (2011).

[20] G. Evensen, J. Geophys. Res. **99**, 10143 (1994).

[21] P. L. Houtekamer and H. L. Mitchell, Mon. Weather Rev. **126**, 796 (1998).

[22] J. L. Anderson, Mon. Weather Rev. **129**, 2884 (2001).

[23] T. Berry and T. Sauer, Tellus **65**, 20331 (2013).
[24] S. J. Julier and J. K. Uhlmann, *A General Method for Approximating Nonlinear Transformations of Probability Distributions*, Tech. Rep. (Robotics Research Group, University of Oxford, 1996).
[25] S. J. Julier and J. K. Uhlmann, in *Proceedings of the 2002 American Control Conference* (IEEE Cat. No.CH37301) (IEEE, New York, 2002).
[26] G. Kitagawa, J. Comput. Graphical Stat. **5**, 1 (1996).
[27] D. T. Pham, Mon. Weather Rev. **129**, 1194 (2001).
[28] S. J. Julier and J. K. Uhlmann, in *Acquisition, Tracking, and Pointing XI*, edited by M. K. Masten and L. A. Stockum, Vol. 3086 (SPIE Proc., 1997), pp. 110–121.
[29] X. Luo and I. M. Moroz, Phys. D **238**, 549 (2009).
[30] Y. Rui and Y. Chen, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001* (IEEE Comput. Soc, 2001), pp. II-786–II-793.
[31] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan, Advances in Neural Information Processing Systems 584 (2001).
[32] D. Simon, IET Control Theory Appl. **4**, 1303 (2010).
[33] Y. Cheng and Z. Liu, in *2011 International Conference on Electrical and Control Engineering* (IEEE, New York, 2011), pp. 3073–3075.
[34] K. Ponomareva, P. Date, and Z. Wang, in *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems* (Eötvös Loránd Univ., Budapest, 2010), pp. 1609–1613.
[35] O. Straka, J. Duník, M. Šimandl, and E. Blasch, *2012 Proceedings of the 15th International Conference on Information Fusion* (IEEE, Singapore, 2012), pp. 2004–2011.
[36] J. Rezaie and J. Eidsvik, Int. J. Control **89**, 2572 (2016).
[37] J. Hou, W. Zhou, W.-A. Zhang, C. Zhang, C. Chen, and C. Shan, in *2019 15th International Conference on Computational Intelligence and Security (CIS)* (IEEE, 2019), pp. 26–30.
[38] D. C. Easley and T. Berry, SIAM/ASA J. Uncert. Quantif. **9**, 1094 (2021).
[39] H. M. Menegaz, J. Y. Ishihara, G. A. Borges, and A. N. Vargas, IEEE Trans. Autom. Control **60**, 2583 (2015).
[40] K. Pearson, Philos. Trans. R. Soc. London, Ser. A **216**, 429 (1916).
[41] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. (Tata McGraw-Hill Education, 2002).
[42] S. J. Julier and J. K. Uhlmann, Proc. IEEE **92**, 401 (2004).
[43] Generalized unscented transform MATLAB® source code, https://github.com/Schiff-Lab/Generalized-Unscented-Transform.