ORBIT UNCERTAINTY PROPAGATION OF A TUMBLING HIGH AREA-TO-MASS-RATIO SPACE OBJECT

Yang Cheng*

Space surveillance tracking problems, that are characterized by sparse observation data, necessitate long-term orbit propagation. For high area-to-mass-ratio space objects, long-term orbit uncertainty propagation needs to account for both conservative and non-conservative effect, such as the effect of solar radiation pressure, which results in non-negligible coupling between the orbital motion and the attitude motion, and a high-dimensional uncertainty propagation problem. The sparse grid method and the Monte Carlo methods are presented for orbit uncertainty propagation and are compared in the numerical simulation of a tumbling, high area-to-mass-ratio space object, whose attitude and angular velocity are unknown. The advantage of these methods is that they are non-intrusive and straightforward to implement and can accommodate Gaussian and non-Gaussian parametric uncertainties in the orbital and attitude motion models.

INTRODUCTION

Long-term orbit uncertainty propagation of space objects has gained much attention in recent years. It is especially necessary and important in the presence of infrequent observations of the objects. The orbital parameters of a space object are generally modeled as random variables and the orbit uncertainty is generally characterized by the probability density function or the moments of the orbit parameters. Even if the probability distribution of the orbital parameters is Gaussian initially, it will eventually become non-Gaussian without the conditioning of the distribution with measurements (i.e. after a long period of propagation through the nonlinear orbital dynamics), rendering the representation and propagation of the orbit uncertainty as a Gaussian distribution physically inaccurate.

Solar radiation pressure effects, which result from the momentum exchange between the surfaces of a space object and the photons of sunlight absorbed and reflected by the surfaces, significantly affect the behavior of the orbital dynamical system.[1] The simplest solar radiation pressure model is the cannonball model. It assumes that the object is a sphere with constant thermal-optical properties.[1] According to the cannonball model, the solar radiation pressure is independent of the attitude and thus the orbital motion and the attitude motion are completely decoupled. This model is oversimplified for objects which are either non-spherical and/or have tumbling dynamics that are slowly varying with respect to the orbital period, for example, high area-to-mass-ratio objects.[2] In more realistic models, the object is characterized by an assembly of simple geometric shapes such as flat plates.[3,1] Because the solar radiation pressure in these models depends on the orientation of the object, the orbital and attitude motions need to be considered simultaneously. The resultant uncertainty propagation problem is twelve-dimensional or higher, depending on whether the uncertainty of additional model parameters is accounted for.

*Assistant Professor, Department of Aerospace Engineering, Mississippi State University, Mississippi State, MS 39762.

Gaussian mixture and moment-based methods have been proposed to capture the non-Gaussian uncertainty resulting from the nonlinear dynamics. The former approximates the probability density function by a Gaussian mixture or Gaussian sum;[4,3,5,2] the latter explicitly or implicitly propagates the moments of the orbital parameters based on the Monte Carlo method, the state transition tensor method,[6,7] generalized Polynomial Chaos,[8] or the sparse grid method.[9]

Moment propagation is important in and of itself and also plays a role in the Kalman filtering based Gaussian mixture method,[10,11,12,5] which in the simplest form uses a moment propagation method to determine the means and covariances (the first two central moments) of the components of the Gaussian mixture model. While in the Gaussian mixture model the moments are determined under the Gaussian assumption, the general moment propagation method of this article is not limited to Gaussian or Gaussian mixture distributions.

The Kalman filtering based Gaussian mixture method is the state of the art for obtaining the Gaussian mixture model. It essentially reduces the probability density propagation problem to a parallel set of mean and covariance propagation problems that are handled by Kalman filtering. The method is one of the most efficient uncertainty propagation methods, when the assumptions for the structure of the Gaussian mixture model and the validity of using Kalman filtering for nonlinear systems hold. For uncertainty propagation problems with large uncertainty, high nonlinearity, and/or high non-Gaussianity, however, the Kalman filtering based Gaussian mixture method may be inefficient. The number of Gaussian components may need to be adaptively updated by splitting and merging the Gaussian components. Frequent splitting and merging may significantly increase the computational complexity of the method; moreover, efficient splitting and merging in multi-dimensional space is a challenging problem in itself. For the Gaussian assumption of Kalman filtering to be valid in nonlinear orbit propagation, the Kalman filtering based Gaussian mixture method tends to use a large number of small Gaussian components. For systems with large uncertainties, such a representation may become inaccurate or inefficient because even a large number of small Gaussian components may still fail to cover all the regions of significance effectively. In addition, the Kalman filtering based Gaussian mixture method cannot handle non-Gaussian random variables in the propagation model directly, which need to be approximated by a Gaussian mixture in advance. The approximation is a nontrivial task when multi-dimensional non-Gaussian random variables need to be approximated. That the Kalman filtering based Gaussian mixture method may be inefficient for certain complex systems motivates the study of the alternative, simulation based method for obtaining a Gaussian mixture model for the non-Gaussian probability density of the orbital parameters. The basic idea of the method is to reconstruct the Gaussian mixture model from random samples simulated by orbital propagation. The simulation based method consists of two steps. The first step involves generating random samples via Monte Carlo simulation of orbit propagation. The second step involves probability density estimation from the propagated random samples using the Expectation-Maximization method. Both steps are straightforward to implement and based on mature techniques. The simulation based method is complementary to the Kalman filtering based Gaussian mixture method and is particularly suited for complex orbit uncertainty propagation problems with large uncertainty, high nonlinearity and/or high non-Gaussianity.

This article presents the Monte Carlo methods and the sparse grid method for moment propagation and the Monte Carlo/Expectation Maximization method for obtaining a Gaussian mixture model of the non-Gaussian probability density.

DYNAMICS MODEL

Orbital and Attitude Motion

Under the assumption that the space object is a rigid body and that solar radiation pressure is the only non-conservative perturbation, the dynamic model consists of both the translational and rotational motion and is of the form[3]

$$\dot{\mathbf{r}}^i = \mathbf{v}^i \tag{1a}$$

$$\dot{\mathbf{v}}^i = \mathbf{a}_g^i(\mathbf{r}^i) + \mathbf{a}_{3rd}^i(\mathbf{r}^i) + \mathbf{a}_{srp}^i(\mathbf{r}^i,\, \mathbf{q}_{b/i}) \tag{1b}$$

$$\dot{\mathbf{q}}_{b/i} = \frac{1}{2}\Omega(\boldsymbol{\omega}_{b/i}^b)\, \mathbf{q}_{b/i} \tag{1c}$$

$$\dot{\boldsymbol{\omega}}_{b/i}^b = J^{-1}\left(\mathbf{m}_{srp}^b(\mathbf{r}^i,\, \mathbf{q}_{b/i}) - \boldsymbol{\omega}_{b/i}^b \times J\boldsymbol{\omega}_{b/i}^b\right) \tag{1d}$$

where $\mathbf{r}^i$ and $\mathbf{v}^i$ are, respectively, the position and velocity expressed in the Earth-centered inertial frame, $\mathbf{a}_g^i$ is the gravitational acceleration due to the Earth expressed in the Earth-centered inertial frame, $\mathbf{a}_{3rd}^i$ is the gravitational acceleration due to other gravitational bodies such as the Sun and the Moon expressed in the Earth-centered inertial frame, $\mathbf{a}_{srp}^i$ is the solar radiation pressure acceleration expressed in the Earth-centered inertial frame, $\mathbf{q}_{b/i}$ is the quaternion representation of the inertial attitude of the object or the rotation from the Earth-centered inertial frame to the body frame of the object, $\boldsymbol{\omega}_{b/i}^b$ is the the absolute angular velocity expressed in the object body frame, J is the moment-of-inertia matrix with respect to the body frame, $\mathbf{m}_{srp}^b$ is the solar radiation pressure moment expressed in the object body frame, and

$$\Omega(\boldsymbol{\omega}_{b/i}^b) = \begin{bmatrix} -[\boldsymbol{\omega}_{b/i}^b\times] & \boldsymbol{\omega}_{b/i}^b \\ -(\boldsymbol{\omega}_{b/i}^b)^T & 0 \end{bmatrix} \tag{2}$$

The attitude quaternion is subject to the unity norm constraint, that is, $\mathbf{q}_{b/i}^T\mathbf{q}_{b/i} = 1$. The transformation matrix $T_{b/i}$ from the Earth-centered inertial frame to the object body frame is related to the attitude quaternion $\mathbf{q}_{b/i}$ by

$$T_{b/i} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2q_1q_2 + 2q_3q_4 & 2q_1q_3 - 2q_2q_4 \\ 2q_1q_2 - 2q_3q_4 & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2q_1q_4 + 2q_2q_3 \\ 2q_1q_3 + 2q_2q_4 & 2q_2q_3 - 2q_1q_4 & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{3}$$

with

$$\mathbf{q}_{b/i} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{4}$$

The general expression for the Earth gravitational acceleration $\mathbf{a}_g^i$ is

$$\mathbf{a}_g^i = T_{i/f}\ \left[\nabla \cdot U\left(T_{f/i}\ \mathbf{r}^i\right)\right] \tag{5}$$

where ∇ is the gradient operator, U is the Earth gravitational potential, $T_{f/i}$ is the transformation matrix from the Earth-centered inertial frame to the Earth-centered Earth-fixed frame, and $T_{i/f}$ is the transformation matrix from the Earth-centered Earth-fixed frame to the Earth-centered inertial

frame. The two transformation matrices are related by $T_{i/f} = T_{f/i}^T$, where the superscript T denotes the matrix transpose. When the Earth is treated as a point mass, the corresponding acceleration reduces to

$$\mathbf{a}_g^i = -\frac{\mu_{earth}}{r^3}\mathbf{r}^i \tag{6}$$

where μ_{earth} is the Earth gravitational constant and $r = \|\mathbf{r}^i\|$ is the magnitude of $\mathbf{r}^i$, i.e. the geocentric distance of the object from the Earth.

The general expression for the third-body gravitational acceleration is

$$\mathbf{a}_{3rd}^i = \sum_{j=1}^{L} \mu_j \left(\frac{\mathbf{d}_{ij}}{d_{ij}^3} - \frac{\mathbf{r}_j^i}{r_j^3} \right) \tag{7}$$

where μ_j is the gravitational constant of the j^{th} body, $\mathbf{r}_j^i$ is the position of the j^{th} body expressed in the Earth-centered inertial frame, $\mathbf{d}_{ij} = \mathbf{r}_j^i - \mathbf{r}^i$, $r_j = \|\mathbf{r}_j^i\|$, and $d_{ij} = \|\mathbf{d}_{ij}\|$.

In Eqs. (1), it is the solar radiation pressure acceleration $\mathbf{a}_{srp}^i$ and moment $\mathbf{m}_{srp}^b$ that cause the coupling of the translational motion (Eqs. (1a) and (1b)) and the rotational motion (Eqs. (1c) and (1d)).

Solar Radiation Pressure

Solar radiation pressure is a function of the configuration of the space object, which is in general modeled as an assembly of flat plates, cylinders, or spheres. In this article, the flat plate model is used. The acceleration due to solar radiation pressure acting on an object consisting of K flat plates can be expressed as

$$\mathbf{a}_{srp}^i = f\, s_f \left(\frac{r_{au}}{r_{so}} \right)^2 \sum_{k=1}^{K} \frac{A_k}{m} \cos\phi_k \left[(1-\rho_k)\hat{\mathbf{u}}_{sun}^i + 2\left(\frac{\delta_k}{3} + \rho_k \cos\phi_k \right) \hat{\mathbf{u}}_k^i \right] \tag{8}$$

where $f \in [0,1]$ is the Earth shadow factor, s_f is the solar flux constant, r_{au} is the distance of one astronomical unit, $r_{so} = \|\mathbf{r}_{sun}^i - \mathbf{r}^i\|$ is the distance between the object and the Sun, A_k is the area of the k^{th} plate, m is the total mass of the object, and ρ_k and δ_k are the specular and diffuse reflection coefficients of the k^{th} plate, respectively. The unit vector $\hat{\mathbf{u}}_{sun}^i$ is the direction from the object to the Sun expressed in the Earth-centered inertial frame, the unit vector $\hat{\mathbf{u}}_k^i$ is the normal of the k^{th} flat plate expressed in the Earth-centered inertial frame, and ϕ_k is the angle between $\hat{\mathbf{u}}_{sun}^i$ and $\hat{\mathbf{u}}_k^i$. They are calculated as

$$\hat{\mathbf{u}}_{sun}^i = \frac{\mathbf{r}_{sun}^i - \mathbf{r}^i}{\|\mathbf{r}_{sun}^i - \mathbf{r}^i\|} \tag{9}$$

$$\hat{\mathbf{u}}_k^i = \left(T_{b/i}\right)^T \hat{\mathbf{u}}_k^b \tag{10}$$

$$\cos\phi_k = \left(\hat{\mathbf{u}}_{sun}^i\right)^T \hat{\mathbf{u}}_k^i \tag{11}$$

with $\hat{\mathbf{u}}_k^b$ the normal of the k^{th} flat plate expressed in the object body frame, which is constant if the object is a rigid body.

The moment $\mathbf{m}_{srp}^b$ due to solar radiation pressure is

$$\mathbf{m}_{srp}^b = \sum_{k=1}^{K} \mathbf{r}_k^b \times \mathbf{f}_{srp}^b \tag{12}$$

where $\times$ denotes the vector product, $\mathbf{r}_k^b$ is the position of the center of the k^{th} plate with respect to the center of mass of the object expressed in the object body frame and

$$\mathbf{f}_{srp}^b = m\mathbf{a}_{srp}^b = m\, T_{b/i}\, \mathbf{a}_{srp}^i \tag{13}$$

The cylindrical model is used for Earth shadow modeling. The cylindrical model assumes that the Sun is infinitely far away from the Earth, thus causing the light rays to be completely parallel which yields a cylindrical shadow extending behind the Earth with respect to the Sun. According to the cylindrical model, the Earth shadow factor f is binary, taking on 0 when the object is in the Earth shadow and 1 otherwise. A refined model is the conic model, which does not assume that the rays of light emitted by the Sun are parallel, and which models the geometric umbra/penumbra shadow of the Sun due to the presence of the Earth. According to the conic model, the Earth shadow factor f may take on any value in $[0, 1]$. The simpler cylindrical model is used in this article. The cylindrical Earth shadow factor f is calculated as

$$f = \begin{cases} 0 & \cos\psi < 0, \text{ and } \|\mathbf{r}^i\|^2(1-\cos^2\psi) < R_e^2 \\ 1 & \text{elsewise} \end{cases} \tag{14}$$

where R_e is the Earth radius and

$$\cos\psi = \frac{(\mathbf{r}^i)^T\mathbf{r}_{sun}^i}{\|\mathbf{r}^i\| \cdot \|\mathbf{r}_{sun}^i\|} \tag{15}$$

MOMENT PROPAGATION

For the coupled dynamic model given by Eq. (1), the state is defined as

$$\mathbf{x}^T = \begin{bmatrix} (\mathbf{r}^i)^T & (\mathbf{v}^i)^T & (\mathbf{q}_{b/i})^T & (\boldsymbol{\omega}_{b/i}^b)^T \end{bmatrix} \tag{16}$$

The dynamic model can be written in compact form as

$$\dot{\mathbf{x}}(t) = \mathbf{g}(t, \mathbf{x}(t), \mathbf{p}), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{17}$$

where $\mathbf{x}(t)$ denotes the state at time t, $\mathbf{g}$ the deterministic state dynamics, $\mathbf{x}_0$ the initial state, and $\mathbf{p}$ the model parameter vector. Examples of model parameters include the moment-of-inertia matrix and the reflection coefficients. The formal solution $\mathbf{x}_t \equiv \mathbf{x}(t)$ of the dynamics is denoted by

$$\mathbf{x}_t = \boldsymbol{\phi}(t, \mathbf{X}_0) = \boldsymbol{\phi}(t, \mathbf{x}_0, \mathbf{p}) \tag{18}$$

with

$$\mathbf{X}_0^T = \begin{bmatrix} \mathbf{x}_0^T & \mathbf{p}^T \end{bmatrix} \tag{19}$$

The probability distribution $p_0(\mathbf{X}_0)$ of $\mathbf{X}_0$ is assumed to be known, but not necessarily Gaussian.

The objective of moment propagation is to calculate the moments of $\mathbf{x}_t$. The first two moments of $\mathbf{x}_t$ are defined by

$$\hat{\mathbf{x}}_t \equiv E[\mathbf{x}_t] = \int \mathbf{x}_t \cdot p_t(\mathbf{x}_t) \cdot d\mathbf{x}_t \tag{20a}$$

$$M_t \equiv E[\mathbf{x}_t\, \mathbf{x}_t^T] = \int \mathbf{x}_t\, \mathbf{x}_t^T \cdot p_t(\mathbf{x}_t) \cdot d\mathbf{x}_t \tag{20b}$$

where E, $\hat{\mathbf{x}}_t$, M_t denote the expectation operator, the mean, and the mean square matrix, respectively. Higher moments are defined similarly. The moment definitions in Eqs. (20) cannot be directly used to calculate the moments because $p_t(\mathbf{x}_t)$ is unknown. However, since

$$\int \mathbf{x}_t \cdot p_t(\mathbf{x}_t) \cdot d\mathbf{x}_t = \int [\boldsymbol{\phi}(t, \mathbf{X}_0)] \cdot p_0(\mathbf{X}_0) \cdot d\mathbf{X}_0 \tag{21a}$$

$$\int \mathbf{x}_t \, \mathbf{x}_t^T \cdot p_t(\mathbf{x}_t) \cdot d\mathbf{x}_t = \int \left[\boldsymbol{\phi}(t, \mathbf{X}_0) \, \boldsymbol{\phi}^T(t, \mathbf{X}_0)\right] \cdot p_0(\mathbf{X}_0) \cdot d\mathbf{X}_0 \tag{21b}$$

the moments can be calculated as

$$\hat{\mathbf{x}}_t = \int [\boldsymbol{\phi}(t, \mathbf{X}_0)] \cdot p_0(\mathbf{X}_0) \cdot d\mathbf{X}_0 \tag{22a}$$

$$M_t = \int \left[\boldsymbol{\phi}(t, \mathbf{X}_0) \, \boldsymbol{\phi}^T(t, \mathbf{X}_0)\right] \cdot p_0(\mathbf{X}_0) \cdot d\mathbf{X}_0 \tag{22b}$$

The covariance matrix can be computed from the mean and the mean square matrix using

$$P_t \equiv E[(\mathbf{x}_t - \hat{\mathbf{x}}_t) \, (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T] = M_t - \hat{\mathbf{x}}_t \, \hat{\mathbf{x}}_t^T \tag{23}$$

Methods for Moment Propagation

To determine the mean and covariance, the Monte Carlo method, the quasi-Monte Carlo method, and the sparse grid method are used to approximate the integrations in Eqs. (22) and (23). These quadrature methods approximate the integration of the form $\int \mathbf{f}(\mathbf{X}_0) p_0(\mathbf{X}_0) \, d\mathbf{X}_0$, where $\mathbf{f}(\mathbf{X}_0)$ is an arbitrary function, as

$$\int \mathbf{f}(\mathbf{X}_0) p_0(\mathbf{X}_0) \, d\mathbf{X}_0 \approx \sum_{i=1}^{N} w^{(i)} \mathbf{f}(\mathbf{X}_0^{(i)}) \tag{24}$$

where $\mathbf{X}_0^{(i)}$ denotes the i^{th} point and $w^{(i)} > 0$ denotes its associated weight, with $\sum_{i=1}^{N} w^{(i)} = 1$. The methods differ only in the generation of $\mathbf{X}_0^{(i)}$ and $w^{(i)}$.

General Procedure Given a set of N points and weights $\{\mathbf{X}_0^{(i)}, w^{(i)}\}$, $i = 1, \ldots, N$, the three quadrature methods calculate the moments in Eq. (22) by the same procedure:

$$\mathbf{x}_t^{(i)} = \boldsymbol{\phi}(t, \mathbf{X}_0^{(i)}), \quad i = 1, \ldots, N \tag{25a}$$

$$\hat{\mathbf{x}}_t \approx \sum_{i=1}^{N} w^{(i)} \mathbf{x}_t^{(i)} \tag{25b}$$

$$M_t \approx \sum_{i=1}^{N} w^{(i)} \mathbf{x}_t^{(i)} \left(\mathbf{x}_t^{(i)}\right)^T \tag{25c}$$

$$P_t = M_t - \hat{\mathbf{x}}_t \, \hat{\mathbf{x}}_t^T \tag{25d}$$

The last equation is equivalent to

$$P_t \approx \sum_{i=1}^{N} w^{(i)} \left(\mathbf{x}_t^{(i)} - \hat{\mathbf{x}}_t\right) \left(\mathbf{x}_t^{(i)} - \hat{\mathbf{x}}_t\right)^T \tag{26}$$

Note that the weights $w^{(i)}$ do not vary with time. The complexity of the quadrature methods is approximately proportional to the number of quadrature points, N.

The quadrature methods are straightforward to implement in that they only need $\boldsymbol{\phi}(t, \mathbf{X}_0^{(i)})$, $i = 1, \ldots, N$, which are obtained by solving ordinary differential equations, but do not need derivatives of $\boldsymbol{\phi}(t, \mathbf{X}_0^{(i)})$. Additionally, they can be used with both Gaussian and non-Gaussian $p_0(\mathbf{X}_0)$.

In this article, $\mathbf{X}_0$ is assumed to be a combination of mutually independent Gaussian and uniform random variables. In the review of the Monte Carlo, quasi-Monte Carlo, and sparse grid methods that follows, it is further assumed that the Gaussian part $\mathbf{X}_0^{\mathcal{N}}$ of $\mathbf{X}_0$ obeys $\mathcal{N}(\mathbf{X}_0^{\mathcal{N}}; \mathbf{0}, I)$, the standard Gaussian distribution with zero mean and covariance matrix the identity matrix, and the uniform part $\mathbf{X}_0^{\mathcal{U}}$ of $\mathbf{X}_0$ obeys $\mathcal{U}[\mathbf{0}, \mathbf{1}]$, the standard uniform distribution on the interval $[0, 1]$ in each and every dimension. A linear transformation is all one needs to convert from the standard Gaussian and uniform distributions to the more general ones. If $\mathbf{X}_0^{\mathcal{N}}$ obeys $\mathcal{N}(\mathbf{X}_0^{\mathcal{N}}; \mathbf{0}, I)$, then $\sqrt{P_0}\mathbf{X}_0^{\mathcal{N}} + \hat{\mathbf{x}}_0$, with $\sqrt{P_0}\sqrt{P_0}^T = P_0$, obeys $\mathcal{N}(\mathbf{X}_0^{\mathcal{N}}; \hat{\mathbf{x}}_0, P_0)$. If $\mathbf{X}_0^{\mathcal{U}}$ obeys $\mathcal{U}[\mathbf{0}, \mathbf{1}]$, then $\mathrm{diag}(\mathbf{b}-\mathbf{a})\mathbf{X}_0^{\mathcal{U}}+\mathbf{b}$, where $\mathrm{diag}(\mathbf{b} - \mathbf{a})$ denotes the diagonal matrix with the main diagonal given by $\mathbf{b} - \mathbf{a}$, obeys $\mathcal{U}[\mathbf{a}, \mathbf{b}]$.

Monte Carlo and Quasi-Monte Carlo Methods In the Monte Carlo method, given $p_0(\mathbf{X}_0)$, the points $\mathbf{X}_0^{(i)}$ are generated as pseudorandom numbers. All weights are identical, given by $w^{(i)} = 1/N$.

In the quasi-Monte Carlo method, given $p_0(\mathbf{X}_0)$, the points $\mathbf{X}_0^{(i)}$ are generated as quasi-random numbers based on low-discrepancy sequences.[13] Several low-discrepancy sequences for the uniform distribution exist, such as Halton sequences and Sobol sequences. Distributions other than the uniform distribution can be generated by a transformation of the uniform distribution. For the standard Gaussian probability density function with zero mean and unity variance $\mathcal{N}(x; 0, 1) = 1/\sqrt{2\pi}\exp(-x^2/2)$, the points $\eta^{(i)}$ are generated by

$$\eta^{(i)} = \psi^{-1}(u^{(i)}) \tag{27}$$

where $\psi(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(x/\sqrt{2}\right)\right]$ is the Gaussian cumulative distribution function, erf is the error function, and $u^{(i)}$ are elements of the Halton or Sobol sequence. The weights of the points are $w^{(i)} = 1/N$.

Sparse Grid Method The multi-dimensional sparse grid quadrature is built upon univariate or one-dimensional quadratures. A general univariate quadrature rule $\mathcal{I}(f)$ is given by

$$\mathcal{I}(f) = \sum_{i=1}^{M} \varpi^{(i)} f(x^{(i)}) \approx \int f(x) w(x) dx \tag{28}$$

where $w(x)$ is a weighting function, $x^{(i)}$ and $\varpi^{(i)}$ are selected to achieve specified accuracy, and $\varpi^{(i)} > 0$, $\sum_{i=1}^{M} \varpi^{(i)} = 1$. The expectation with respect to the standard Gaussian distribution is related to Gauss-Hermite quadrature, with $w(x) = \exp(-x^2)$, $x \in (-\infty, \infty)$ in Eq. (28), by

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) \exp(-\frac{x^2}{2})\, dx = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} f(\sqrt{2}x) \exp(-x^2)\, dx \tag{29}$$

The left-hand side of the equation is the expectation and the right-hand side of the equation is the integration to be approximated by Gauss-Hermite quadrature. The expectation with respect to

the uniform distribution is related to Gauss-Legendre quadrature, with $w(x) = 1$, $x \in [-1, 1]$ in Eq. (28), by

$$\int_0^1 f(x)\,dx = \frac{1}{2}\int_{-1}^{-1} f(\frac{x+1}{2})\,dx \tag{30}$$

A univariate quadrature $\mathcal{I}(f)$ that is exact for polynomials f up to degree $2m-1$ is said to have accuracy level m. A level-m quadrature will be denoted by $\mathcal{I}_m$. The sets of the points and the weights corresponding to $\mathcal{I}_m$ will be denoted by $\mathcal{X}_m$ and $\mathcal{W}_m$, respectively. The minimum number of points of $\mathcal{I}_m$ is m, but $\mathcal{I}_m$ may use more than m points.

An n-dimensional sparse grid quadrature has accuracy level L, if it is exact for all polynomials $x_1^{j_1}x_2^{j_2}\cdots x_n^{j_n}$ with $\sum_{k=1}^n j_k \leq 2L-1$. Such a sparse grid quadrature is denoted by $\mathcal{I}_{n,L}^S$. It is generated from univariate quadratures $\mathcal{I}_l$ with accuracy level l, $1 \leq l \leq L$. The univariate Gauss-Hermite quadrature rule is used for the Gaussian random variables and the univariate Gauss-Legendre quadrature rule is used for the uniform random variables. The Smolyak quadrature rule[14] is used to generate an n-dimensional sparse grid as

$$\mathcal{I}_{n,L}^S(\mathbf{f}) = \sum_{\Xi\in\boldsymbol{\Upsilon}_{n,L}} (\Delta_{j_1}\otimes\Delta_{j_2}\otimes\cdots\otimes\Delta_{j_n})(\mathbf{f}) \tag{31}$$

with $\otimes$ representing the tensor product, $\Delta_{j_k} = \mathcal{I}_{j_k} - \mathcal{I}_{j_{k-1}}$, $\mathcal{I}_0 = 0$, $\Xi = (j_1, \cdots, j_n)$, and

$$\boldsymbol{\Upsilon}_{n,L} = \left\{\Xi = (j_1,\cdots,j_n) : j_k \geq 1, \sum_{k=1}^n (j_k - 1) \leq (L-1)\right\} \tag{32}$$

By introducing an auxiliary nonnegative integer q and defining[15]

$$\mathbf{N}_q^n = \begin{cases} \{\Xi \subset \mathbf{N}^n : \sum_{k=1}^n j_k = n+q\} & q \geq 0 \\ \varnothing & q < 0 \end{cases} \tag{33}$$

Eq. (31) can be rewritten as

$$\mathcal{I}_{n,L}^S(\mathbf{f}) = \sum_{q=L-n}^{L-1} (-1)^{L-1-q}\binom{n-1}{L-1-q} \sum_{\Xi\in\mathbf{N}_q^n} (\mathcal{I}_{j_1}\otimes\mathcal{I}_{j_2}\otimes\cdots\mathcal{I}_{j_n})(\mathbf{f}) \tag{34}$$

The set $\mathcal{X}_{n,L}^S$ of the (distinct) points of an n-dimensional, accuracy level L sparse grid quadrature is

$$\mathcal{X}_{n,L}^S = \bigcup_{q=L-n}^{L-1} \bigcup_{\Xi\in\mathbf{N}_q^n} (\mathcal{X}_{j_1}\otimes\mathcal{X}_{j_2}\otimes\cdots\mathcal{X}_{j_n}) \tag{35}$$

where $\mathcal{X}_{j_k}$, $1 \leq j_k \leq L$, is the point set of a univariate quadrature of at least accuracy level j_k. In general, the number of points of $\mathcal{X}_{j_k}$ is less than or equal to that of $\mathcal{X}_{j_k+1}$. An element of $(\mathcal{X}_{j_1}\otimes\mathcal{X}_{j_2}\otimes\cdots\mathcal{X}_{j_n})$ corresponds to an n-dimensional point $\begin{bmatrix} x_1^{(i_{j_1})} & x_2^{(i_{j_2})} & \dots & x_n^{(i_{j_n})}\end{bmatrix}^T$, where $x_k^{(i_{j_k})}$ is the k^{th} coordinate, of which the value is given by one of the elements of $\mathcal{X}_{j_k}$, i.e., $x_k^{(i_{j_k})} \in \mathcal{X}_{j_k}$, $k = 1, \ldots, n$.

The set of the weights of the elements of the point set $\mathcal{X}_{n,L}^S$ is denoted by $\mathcal{W}_{n,m}^S$. If a point $\begin{bmatrix} x_1^{(i_{j_1})} & x_2^{(i_{j_2})} & \dots & x_n^{(i_{j_n})} \end{bmatrix}^T$ is unique in all the combinations in Eq. (35), i.e., it belongs to one $(\mathcal{X}_{j_1} \otimes \mathcal{X}_{j_2} \otimes \cdots \mathcal{X}_{j_n})$ only, the weight associated with the point is $(-1)^{L-1-q}\binom{n-1}{L-1-q}\prod_{k=1}^n \varpi^{(i_{j_k})}$, with $\varpi^{(i_{j_k})}$ the weight associated with $x_k^{(i_{j_k})}$. All points of $\mathcal{X}_{n,L}^S$ other than the origin are unique if the univariate quadratures of different accuracy levels share no points or the origin only, i.e., $\mathcal{X}_{j_k} \bigcap \mathcal{X}_{j'_k} = \varnothing$ or $\mathcal{X}_{j_k} \bigcap \mathcal{X}_{j'_k} = \{\mathbf{0}\}$, for $j_k \neq j'_k$. When a point in $\mathcal{X}_{n,L}^S$ belongs to multiple $(\mathcal{X}_{j_1} \otimes \mathcal{X}_{j_2} \otimes \cdots \mathcal{X}_{j_n})$, the weight associated with the point is the sum of the weights in all occurrences. In general, the elements of $\mathcal{W}_{n,L}^S$ cannot be all positive, especially for large n.

GAUSSIAN MIXTURE MODEL OF NON-GAUSSIAN DENSITY

The moment propagation methods in the previous section provide partial information about the orbit uncertainty. The more difficult density estimation method is now presented in this section.

Recall that the formal solution of the dynamics is denoted by $\mathbf{x}_t = \boldsymbol{\phi}(t, \mathbf{x}_0, \mathbf{p})$. For sake of simplicity, the parameter vector $\mathbf{p}$ is assumed to be known in the density estimation problem. The dynamics is rewritten as $\mathbf{x}_t = \boldsymbol{\phi}(t; \mathbf{x}_0, t_0)$.

Suppose that the initial state $\mathbf{x}_0$ is a random vector with known probability density function $p_0(\mathbf{x}_0)$. The probability density function $p_t(\mathbf{x}_t)$ at t is given by[7]

$$p_t(\mathbf{x}_t) = \frac{p_0(\mathbf{x}_0)}{|\det\ \Phi(t, t_0)|} \tag{36}$$

where $|\cdot|$ denotes the absolute value and $\det$ denotes the matrix determinant. The initial state $\mathbf{x}_0$ is related to $\mathbf{x}_t$ by the inverse dynamics, given by

$$\mathbf{x}_0 = \mathbf{x}_0(\mathbf{x}_t) = \boldsymbol{\phi}^{-1}(t_0; \mathbf{x}_t, t) \tag{37}$$

and $\Phi(t, t_0)$ is the state transition matrix, given by

$$\Phi(t, t_0) = \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_0^T} \tag{38}$$

where the superscript T denotes the matrix or vector transpose. All Hamiltonian systems have $\det \Phi(t, t_0) = 1$.[7] In these systems, $p_t(\mathbf{x}_t)$ reduces to[7]

$$p_t(\mathbf{x}_t) = p(\mathbf{x}_0(\mathbf{x}_t)) \tag{39}$$

Here $\mathbf{x}_0(\mathbf{x}_t)$ is used to emphasize that $\mathbf{x}_0$ is a function of $\mathbf{x}_t$.

The objective of the problem is to approximate the non-Gaussian $p_t(\mathbf{x}_t)$ by a Gaussian mixture model, given by

$$p_t(\mathbf{x}_t) \approx \hat{p}_t(\mathbf{x}_t) = \sum_{k=1}^{K_t} w_{k,t}\, \mathcal{N}\left(\mathbf{x}_t; \hat{\mathbf{x}}_{k,t}, P_{k,t}\right) \tag{40}$$

where the n-variate Gaussian probability density with mean $\hat{\mathbf{x}}_{k,t}$ and covariance $P_{k,t}$ is defined by

$$\mathcal{N}\left(\mathbf{x}_t; \hat{\mathbf{x}}_{k,t}, P_{k,t}\right) = \frac{1}{\sqrt{\det(2\pi P_{k,t})}} \exp\left[-\frac{1}{2}(\mathbf{x}_t - \hat{\mathbf{x}}_{k,t})^T P_{k,t}^{-1}(\mathbf{x}_t - \hat{\mathbf{x}}_{k,t})\right] \tag{41}$$

The total number of Gaussian components is K_t and the weights on the Gaussian components satisfy

$$\sum_{k=1}^{K_t} w_{k,t} = 1, \quad w_{k,t} \geq 0 \tag{42}$$

It is important to note that K_t, $w_{k,t}$, $\hat{\mathbf{x}}_{k,t}$, and $P_{k,t}$ are all time-varying in general.

A simulation based method is presented for obtaining the Gaussian mixture model for $p_t(\mathbf{x}_t)$. It consists of two steps. The first step involves generating random samples via Monte Carlo simulation. The second step involves probability density estimation from random samples using the Expectation-Maximization method. The two steps are described in the two subsections that follow.

Monte Carlo Simulation

The Monte Carlo simulation involves generating a set of independent identically distributed random samples $\mathbf{x}_0^{(i)}$ and then propagate them to time t to obtain $\mathbf{x}_t^{(i)}$. It only requires that the random number generators and the orbit propagation code be available. The probability density $p_0(\mathbf{x}_0)$ may be Gaussian or non-Gaussian.

The sampling of $\mathbf{x}_0^{(i)}$ is denoted by

$$\mathbf{x}_0^{(i)} \sim p_0(\mathbf{x}_0), \quad i = 1, \ldots, N \tag{43}$$

where N is the number of random samples. Most random number generators are for independent random variables only, but the components of $\mathbf{x}_0$ may not be independent of each other. Without loss of generality, one may assume the existence of a relationship between the n-dimensional $\mathbf{x}_0$ and a d-dimensional ($d \leq n$) vector $\mathbf{z}$ of independent random variables, given by

$$\mathbf{x}_0 = \mathbf{g}(\mathbf{z}) \tag{44}$$

If both $\mathbf{x}_0$ and $\mathbf{z}$ are Gaussian random vectors, $\mathbf{g}$ corresponds to a linear transformation. The probability density of $\mathbf{z}$ can be derived from $p_0(\mathbf{x}_0)$, given by $p_z(\mathbf{z}) = \prod_{j=1}^{d} p_{z_j}(z_j)$.

So, the sampling step is carried out in two steps:

$$\mathbf{z}^{(i)} \sim p_z(\mathbf{z}), \quad i = 1, \ldots, N \tag{45}$$

and

$$\mathbf{x}_0^{(i)} = \mathbf{g}(\mathbf{z}^{(i)}), \quad i = 1, \ldots, N \tag{46}$$

The propagation step is given by

$$\mathbf{x}_t^{(i)} = \boldsymbol{\phi}(t; \mathbf{x}_0^{(i)}, t_0), \quad i = 1, \ldots, N \tag{47}$$

In the orbit uncertainty propagation problems in this article, $\mathbf{z}$ consists of independent Gaussian random variables and uniform random variables. They can be generated using standard random number generators.

Expectation-Maximization Algorithm

Given the independent, identically distributed random samples $\mathbf{x}_t^{(i)}$, $i = 1, \ldots, N$, from the Monte Carlo simulation, we can immediately have the empirical probability density in terms of Dirac's delta function, given by

$$\hat{p}_t(\mathbf{x}_t) = \sum_{i=1}^{N} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \tag{48}$$

This is, however, an inaccurate approximation to the probability density because $\hat{p}_t(\mathbf{x}_t)$ is zero at all points other than $\mathbf{x}_t^{(i)}$, $i = 1, \ldots, N$.

A better approximation of the probability density is the Gaussian mixture model. A maximum likelihood estimation problem is solved to reconstruct the Gaussian mixture model below from the random samples:

$$\hat{p}_t(\mathbf{x}_t) = \sum_{k=1}^{K_t} w_{k,t} \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{k,t}, P_{k,t}) \tag{49}$$

The Kalman filtering based Gaussian mixture method uses the same form for approximating the non-Gaussian probability density, but it usually uses small Gaussian components. As a result, it either needs to use a very large number of Gaussian components to cover all regions of significance or is inaccurate as a density estimation method.

Given the independent identically distributed random samples $\mathbf{x}_t^{(i)}$, $i = 1, \ldots, N$, the likelihood function is

$$L = \prod_{i=1}^{N} \hat{p}_t(\mathbf{x}_t^{(i)}) \tag{50}$$

Without constraint in K_t, large numbers of components are usually favored by the probability density method over smaller numbers and the random samples $\mathbf{x}_t^{(i)}$ are usually overfitted. Many information criteria are available to limit the number K_t of Gaussian components. The Bayesian information criterion is used here, defined by[16]

$$\text{BIC} = -2\log(L) + M\log(N) \tag{51}$$

where M is the number of independent parameters in the Gaussian mixture model. The best Gaussian mixture model is defined as the one minimizing the Bayesian information criterion.

Given K_t, the Expectation-Maximization algorithm updates $w_{k,t}$, $\hat{\mathbf{x}}_{k,t}$, and $P_{k,t}$ iteratively until it converges to a local minimum. Given the present best estimates $w_{k,t}$, $\hat{\mathbf{x}}_{k,t}$, and $P_{k,t}$, the key

operations in one iteration are summarized below:[16]

$$\pi_k^{(i)} = \frac{w_{k,t}\mathcal{N}\left(\mathbf{x}_t^{(i)}; \hat{\mathbf{x}}_{k,t}, P_{k,t}\right)}{\sum_{k=1}^{K_t} w_{k,t}\mathcal{N}\left(\mathbf{x}_t^{(i)}; \hat{\mathbf{x}}_{k,t}, P_{k,t}\right)}, \quad k = 1, \ldots, K_t,\ i = 1, \ldots, N \tag{52}$$

$$w_{k,t} \leftarrow \frac{\sum_{i=1}^{N} \pi_k^{(i)}}{N}, \quad k = 1, \ldots, K_t \tag{53}$$

$$\hat{\mathbf{x}}_{k,t} \leftarrow \frac{\sum_{i=1}^{N} \pi_k^{(i)} \mathbf{x}_t^{(i)}}{\sum_{s=1}^{N} \pi_k^{(i)}}, \quad k = 1, \ldots, K_t \tag{54}$$

$$\tilde{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(i)} - \hat{\mathbf{x}}_{k,t}, \quad k = 1, \ldots, K_t \tag{55}$$

$$P_{k,t} \leftarrow \frac{\sum_{i=1}^{N} \pi_k^{(i)} \left(\tilde{\mathbf{x}}_t^{(i)}\right)\left(\tilde{\mathbf{x}}_t^{(i)}\right)^T}{\sum_{i=1}^{N} \pi_k^{(i)}}, \quad k = 1, \ldots, K_t \tag{56}$$

where $\leftarrow$ denotes "assigned to."

Generalized Polynomial Chaos Based Simulation

An alternative simulation method for generating the random samples is presented. It is based on generalized Polynomial Chaos. The basic idea of generalized Polynomial Chaos is to approximate the n-dimensional $\mathbf{x}_t$ as a combination of orthogonal polynomials of the d-dimensional independent random vector $\mathbf{z}$, whose probability density is $p_z(\mathbf{z}) = \prod_{j=1}^{d} p_{z_j}(z_j)$, given by[17]

$$\mathbf{x}_t(\mathbf{z}) \approx \hat{\mathbf{x}}_t(\mathbf{z}) = \sum_{0 \le |\mathbf{i}| \le m} \mathbf{c}_{\mathbf{i}}(t)\, \psi_{\mathbf{i}}(\mathbf{z}) \tag{57}$$

where $\mathbf{i} = (i_1, \ldots, i_d)$ is a sequence of ordered indices, with $|\mathbf{i}|$ defined by

$$|\mathbf{i}| = \sum_{j=1}^{d} i_j \tag{58}$$

The inequality $0 \le |\mathbf{i}| m$ means that the highest total degree of the polynomials is m. The total number of terms in the summation in Eq. (57) is given by the binomial coefficient C_m^{d+m}. The orthogonal polynomial of $\mathbf{z}$ is given by

$$\psi_{\mathbf{i}}(\mathbf{z}) = \prod_{j=1}^{d} \psi_{i_j}(z_j) \tag{59}$$

where $\psi_{i_j}(\cdot)$ are the i_j^{th}-degree univariate orthogonal polynomials associated with $p_{z_j}(\cdot)$. They satisfy[17]

$$\mathbb{E}\left[\psi_{m_j}(z_j)\right] = \int \psi_{m_j}(z) p_{z_j}(z)\, dz = 0 \tag{60}$$

and

$$\mathbb{E}\left[\psi_{m_j}(z_j)\psi_{n_j}(z_j)\right] = \int \psi_{m_j}(z)\psi_{n_j}(z) p_{z_j}(z)\, dz = \gamma_{m_j}\delta_{m_j n_j} \tag{61}$$

where $\delta_{m_j n_j}$ is Kronecker's delta function and γ_{m_j} is the normalization constant. It follows that the orthogonal polynomials of $\mathbf{z}$ satisfy[17]

$$\mathbb{E}\left[\psi_{\mathbf{i}}(\mathbf{z})\psi_{\mathbf{j}}(\mathbf{z})\right] = \int \psi_{\mathbf{i}}(\mathbf{z})\psi_{\mathbf{j}}(\mathbf{z})p_z(\mathbf{z})\,d\mathbf{z} = \gamma_{\mathbf{i}}\delta_{\mathbf{ij}} \tag{62}$$

with

$$\gamma_{\mathbf{i}} = \prod_{k=1}^{d} \gamma_{i_k}, \quad \delta_{\mathbf{ij}} = \prod_{k=1}^{d} \delta_{i_k j_k} \tag{63}$$

The coefficient vector $\mathbf{c}_{\mathbf{i}}(t)$ in Eq. (57) determines the contribution of $\psi_{\mathbf{i}}(\mathbf{z})$ in the orthogonal polynomial approximation. The optimal coefficient vector is the projection of $\mathbf{x}_t$ on $\psi_{\mathbf{i}}(\mathbf{z})$, given by[17]

$$\mathbf{c}_{\mathbf{i}}(t) = \mathbb{E}\left[\mathbf{x}_t(\mathbf{z})\psi_{\mathbf{i}}(\mathbf{z})\right] = \int \mathbf{x}_t(\mathbf{z})\psi_{\mathbf{i}}(\mathbf{z})p_z(\mathbf{z})\,d\mathbf{z} \tag{64}$$

Because $\mathbf{x}_t$ in the integrand is unknown, the optimal $\mathbf{c}_{\mathbf{i}}(t)$ is unknown also. The approximate coefficient vector can be obtained by computing the integral using a sparse grid quadrature rule:[17]

$$\hat{\mathbf{c}}_{\mathbf{i}}(t) = \frac{1}{\gamma_{\mathbf{i}}} \sum_{j=1}^{N^S} w^{(j)} \mathbf{x}_t(\mathbf{z}^{(j)})\psi_{\mathbf{i}}(\mathbf{z}^{(j)}) \tag{65}$$

where $\mathbf{z}^{(j)}$ and $w^{(j)}$ are the quadrature points and their associated weights. This method for computing $\hat{\mathbf{c}}_{\mathbf{i}}(t)$ is known as the stochastic collocation method via direct projection. The number N^S of quadrature points is determined by the dimension d of $\mathbf{z}$ and the accuracy of the quadrature rule.[18,9] There is no simple formula for N^S, however. We require that the sparse grid quadrature be exact for all polynomials of degrees up to $2m$.

Once the coefficient vector $\hat{\mathbf{c}}_{\mathbf{i}}(t)$ is obtained, $\mathbf{x}_t(\mathbf{z}) = \boldsymbol{\phi}(t; \mathbf{x}_0(\mathbf{z}), t_0)$ is approximated by $\hat{\mathbf{x}}_t(\mathbf{z}) = \sum_{0\le|\mathbf{i}|\le m} \mathbf{c}_{\mathbf{i}}(t)\,\psi_{\mathbf{i}}(\mathbf{z})$. Then, the random samples $\mathbf{x}_t^{(i)}$ can be generated as follows:

1.
$$\mathbf{z}^{(i)} \sim p_z(\mathbf{z}), \quad i = 1, \ldots, N \tag{66}$$

2.
$$\mathbf{x}_0^{(i)} = \mathbf{g}(\mathbf{z}^{(i)}), \quad i = 1, \ldots, N \tag{67}$$

3.
$$\mathbf{x}_t^{(i)} = \sum_{0\le|\mathbf{i}|\le m} \hat{\mathbf{c}}_{\mathbf{i}}(t)\,\boldsymbol{\psi}_i(\mathbf{z}^{(i)}), \quad i = 1, \ldots, N \tag{68}$$

The main difference between the above procedure and the Monte Carlo simulation in the previous section is the last step, where orbit propagation is replaced by polynomial evaluations. The accuracy of the polynomial approximation depends on the nonlinearity of the orbital propagation model and the propagation time. The generalized Polynomial Chaos based method is more efficient than the Monte Carlo method when 1) $N^S \ll N$ and 2) the polynomials can be evaluated much faster than orbit propagation.

In the orbit uncertainty propagation problems in this article, $\mathbf{z}$ consists of Gaussian and uniform random variables. Their corresponding orthogonal polynomials are the Hermite and Legendre polynomials, respectively, which are given below.

The Hermite polynomials are orthogonal with respect to the standard Gaussian distribution:[17]

$$\int_{-\infty}^{\infty} H_m(x)H_n(x)\mathcal{N}(x;0,1)\,dx = n!\,\delta_{mn} \tag{69}$$

where

$$\mathcal{N}(x;0,1) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2} \tag{70}$$

They satisfy the following recursion:[17]

$$H_{n+1}(x) = xH_n(x) - nH_{n-1}(x) \tag{71}$$

with

$$H_0(x) = 1, \quad H_1(x) = x, \quad H_2(x) = x^2 - 1 \tag{72}$$

The Legendre polynomials are orthogonal with respect to the uniform distribution in $[-1,1]$:[17]

$$\int_{-1}^{1} P_n(x)P_m(x)\mathcal{U}_{[-1,1]}(x)\,dx = \frac{1}{2n+1}\delta_{mn} \tag{73}$$

$$\mathcal{U}_{[-1,1]}(x) = \frac{1}{2} \tag{74}$$

They satisfy the following recursion:[17]

$$P_{n+1}(x) = \frac{2n+1}{n+1}xP_n(x) - \frac{n}{n+1}P_{n-1}(x) \tag{75}$$

with

$$P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = \frac{3}{2}x^2 - \frac{1}{2} \tag{76}$$

SIMULATION SETUP

A flat-plate space object in a geosynchronous orbit is simulated from January 1, 2011, 00:00:00 to February 1, 2011, 00:00:00. The values of the physical constants are as follows: The Earth gravitational constant $\mu_{earth} = 398600\,\mathrm{km}^3/\mathrm{s}^2$, the Earth radius $R_e = 6378\,\mathrm{km}$, the astronomical unit $r_{au} = 1.49598 \times 10^8\,\mathrm{km}$, and the solar flux constant $s_f = 4.5605 \times 10^{-6}\,\mathrm{N/m}^2$. The position $\mathbf{r}^i_{sun}$ of the Sun is obtained from the JPL HORIZONS on-line solar system data and ephemeris computation service*.

The initial position and velocity expressed in the Earth-centered inertial frame are assumed to be Gaussian random vectors with mean

$$\hat{\mathbf{r}}^i_0 = [42000, 0, 0]^T\,\mathrm{km},\ \hat{\mathbf{v}}^i_0 = [0, 3.0807, 0]^T\,\mathrm{km/s} \tag{77}$$

and covariance

$$P^{rv}_0 = \mathrm{diag}([1,1,1]\,\mathrm{km}^2, [1,1,1]\,\mathrm{m}^2/\mathrm{s}^2) \tag{78}$$

The orbit, without perturbations, is in the X-Y plane of the Earth-centered inertial frame. The initial attitude quaternion and the initial angular velocity are assumed to be independent of each other and

*http://ssd.jpl.nasa.gov/?horizons

independent of the initial position and velocity. The initial attitude quaternion is assumed to obey the uniform distribution on the surface of the three-dimensional unit hypersphere. This corresponds to the case of no prior knowledge about the initial attitude.[19] The initial angular velocity expressed in the object body frame is assumed to be a Gaussian random vector with mean zero and covariance $\mathrm{diag}([1,1,1]) \times 10^{-6}\,\mathrm{rad}^2/\mathrm{s}^2$.

The uniform attitude quaternion is generated based on a nonlinear transformation of uniform random variables:[19]

$$\mathbf{q}_{b/i,0} = \begin{bmatrix} \sin e_2 \, \cos(e_1 - e_3) \\ \sin e_2 \, \sin(e_1 - e_3) \\ \cos e_2 \, \sin(e_1 + e_3) \\ \cos e_2 \, \cos(e_1 + e_3) \end{bmatrix} \tag{79}$$

with

$$e_1 = 2\pi u_1 \tag{80a}$$

$$e_2 = \cos^{-1}(1 - 2u_2) \tag{80b}$$

$$e_3 = 2\pi u_3 \tag{80c}$$

where u_1, u_2, and u_3 are independent uniform random variables on the interval $[0, 1]$ and e_1, e_2, and e_3 are a set of 3-1-3 Euler angles.

The space object is assumed to be a single thin rectangular flat plate with dimensions $2\,\mathrm{m}$ by $4\,\mathrm{m}$ by $1\,\mathrm{mm}$ and density $1500\,\mathrm{kg}/\mathrm{m}^3$. For this configuration, the moment of solar radiation pressure is zero. The area-to-mass ratio is $0.6667\,\mathrm{m}^2/\mathrm{kg}$. The diffuse reflection coefficient δ (subscript k dropped because there is only one plate) is assumed to be zero. In the moment propagation problem, the specular reflection coefficient ρ (subscript k dropped because there is only one plate) is assumed to be independent of the initial orbital and attitude parameters and uniformly distributed in $[0, 1]$. In the density estimation problem, $\rho = 0.8$.

The equations of motion are integrated using the explicit embedded Runge-Kutta Prince-Dormand (8, 9) method of the GNU Scientific Library[†] (version 1.15). The absolute and relative errors of the numerical integrator are set to 1×10^{-14}. The orbit and attitude propagator is implemented as a MATLAB C-MEX function to be used with MATLAB. The propagated attitude quaternion is renormalized every two minutes to ensure that the attitude quaternion obeys the unity norm constraint.

SIMULATION RESULTS

Moment Propagation

The vector $\mathbf{X}_0$ of random initial conditions and model parameters is given by

$$\mathbf{X}_0^T = \begin{bmatrix} (\mathbf{r}_0^i)^T & (\mathbf{v}_0^i)^T & (\mathbf{q}_{b/i,0})^T & (\boldsymbol{\omega}_{b/i,0})^T & \rho \end{bmatrix} \tag{81}$$

The apparent dimension of $\mathbf{X}_0$ is 14, but it has only 13 degrees of freedom because of the unity norm constraint of the attitude quaternion. Hence, the the sparse grids needed to generate the points of $\mathbf{X}_0$ are 13-dimensional instead of 14-dimensional.

The sparse grids are generated using SGMGA[‡], a sparse grid library developed by John Burkardt of Florida State University. The MATLAB version of the sparse grid library is used. SGMGA

[†] `http://www.gnu.org/software/gsl/`

[‡] `http://people.sc.fsu.edu/~jburkardt/m_src/sgmga/sgmga.html`

provides six growth rules of $\mathcal{I}_l$ or $\mathcal{X}_l$, $1 \leq l \leq L$, to generate sparse grids. The growth rule determines how fast the number of points of the univariate quadrature $\mathcal{I}_l$ increases with the accuracy level l. Recall that to generate a multi-dimensional sparse grid, the minimum number of points of $\mathcal{I}_l$ is l, but $\mathcal{I}_l$ may use more than l points. The numbers of points of $\mathcal{I}_l$ for different growth rules are given in Table 1. The numbers of points of the 13-dimensional sparse grids for the 14-dimensional

Table 1. Number of Points of $\mathcal{I}_l$ for Different Growth Rules

l	slow linear	slow linear odd	moderate linear	slow exponential	moderate exponential	full exponential
1	1	1	1	1	1	1
2	2	3	3	3	3	3
3	3	3	5	3	7	7
4	4	5	7	7	7	15
5	5	5	9	7	15	31

Table 2. Number of Points of 13-Dimensional Sparse Grids for $\mathbf{X}_0$

L	slow linear	slow linear odd	moderate linear	slow exponential	moderate exponential	full exponential
1	1	1	1	1	1	1
2	27	27	27	27	27	27
3	365	339	391	339	417	417
4	3329	2679	4005	2705	4577	4759
5	23245	15367	32397	16017	39599	44357

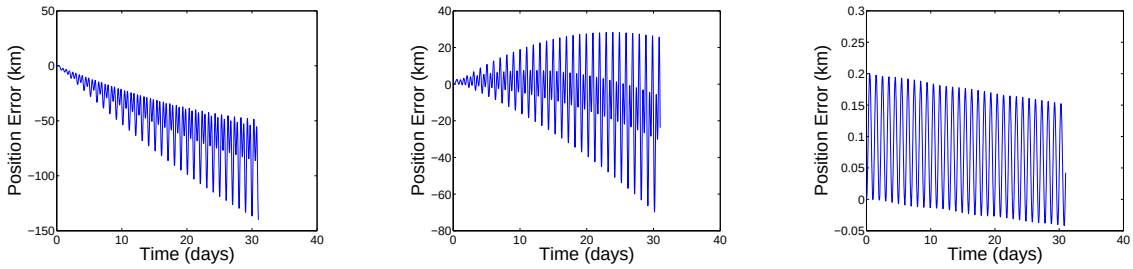
$\mathbf{X}_0$ with accuracy levels up to five are given in Table 2. The level-1 sparse grid is trivial and only has one point. The properties of the weights of the sparse grids are shown in Table 3, where SG(L, g) stands for the sparse grid method with accuracy level L and growth rule g. Growth Rules 1-6 correspond to "slow linear," "slow linear odd," "moderate linear," "slow exponential," "moderate exponential," and "full exponential," respectively, as indicated in Table 1. Note that SG(L, 1) use very large negative and/or positive weights. SG$(2, g), g = 2, \ldots, 6$, are identical to each other, because the sparse grids are constructed from the same univariate quadrature rules. For the same token, SG$(3, 2)$ and SG$(3, 4)$ are identical; and SG$(3, 5)$ and SG$(3, 6)$ are identical.

The difference between orbit propagation of a tumbling space object with and without solar radiation pressure is shown in Figure 1. In the former case, the solar radiation pressure is nonzero when the object is outside of the Earth shadow. The specular and diffuse reflection coefficients are set to 0.8 and 0, respectively. The initial attitude quaternion and the initial angular velocity are set to $[0.5, 0.5, 0.5, 0.5]^T$ and $[0.001, 0.001, 0.001]^T$ rad/s, respectively. In the latter case, the solar radiation pressure is always zero and the orbital motion is completely decoupled from the attitude motion, and the resultant orbit propagation problem is six-dimensional instead of 13-dimensional.

From Figure 1, it can be seen that the solar radiation pressure has a significant effect on the propagation of the orbit and that the dominant difference between the propagation with solar radiation

Table 3. Sparse Grid Weights

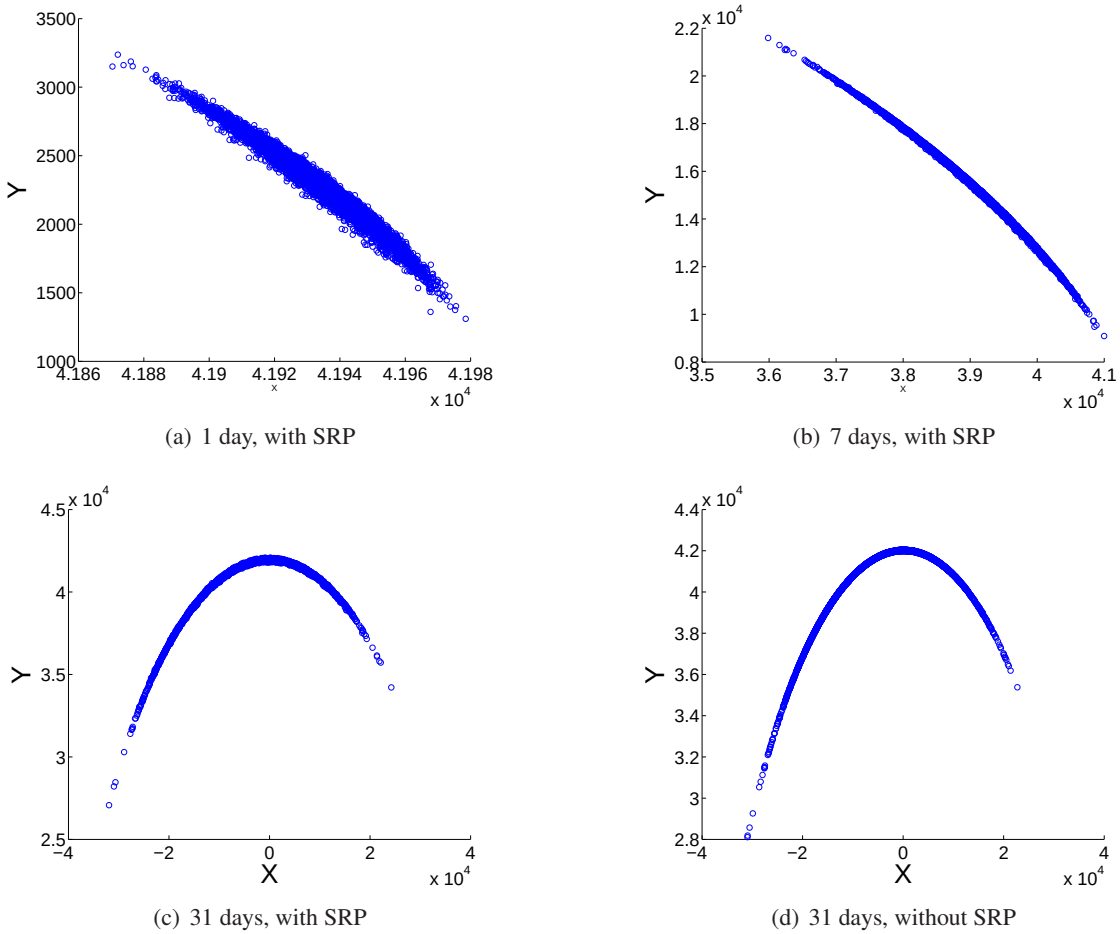
	mininum	maximum	median	standard deviation
SG(2,1)	-12	0.5	0.5	2.4056
SG(2,2)	-4.2222	0.27778	0.16667	0.85276
SG(2,3)	-4.2222	0.27778	0.16667	0.85276
SG(2,4)	-4.2222	0.27778	0.16667	0.85276
SG(2,5)	-4.2222	0.27778	0.16667	0.85276
SG(2,6)	-4.2222	0.27778	0.16667	0.85276
SG(3,1)	-6	73.778	0.25	4.1931
SG(3,2)	-1.0185	8.2963	0.027778	0.50221
SG(3,3)	-1.2963	6.4563	0.046296	0.42092
SG(3,4)	-1.0185	8.2963	0.027778	0.50221
SG(3,5)	-1.2963	5.4687	0.046296	0.37145
SG(3,6)	-1.2963	5.4687	0.046296	0.37145
SG(4,1)	-313.33	36.667	0.125	6.3927
SG(4,2)	-11.785	1.4609	0.007716	0.26056
SG(4,3)	-3.9194	2.2905	0.007716	0.19505
SG(4,4)	-12.773	1.4609	0.007716	0.27611
SG(4,5)	-0.93381	2.0372	0.007716	0.15379
SG(4,6)	-1.1739	2.0372	0.007716	0.16072
SG(5,1)	-154	1042.1	0.0625	8.8266
SG(5,2)	-1.2447	15.098	0.001286	0.13613
SG(5,3)	-1.7961	2.0392	0.001286	0.10112
SG(5,4)	-1.4981	18.871	0.001286	0.16294
SG(5,5)	-9.0026	0.81208	0.0014239	0.071551
SG(5,6)	-1.9419	1.986	0.0014239	0.07771



(a) X axis (b) Y axis (c) Z axis

Figure 1. Difference between the propagated positions with and without solar radiation pressure

pressure and that without solar radiation pressure is in the X-Y plane of the Earth-centered inertial frame. Although the Z component of the Sun position vector expressed in the Earth-centered inertial frame is negative throughout January 2011, the Z component of the propagated position of the object is not always positive. That is due to the attitude motion of the object.



(a) 1 day, with SRP

(b) 7 days, with SRP

(c) 31 days, with SRP

(d) 31 days, without SRP

Figure 2. Monte Carlo samples of position

Figure 2 shows the uncertainty distributions at Day 1, Day 7, and Day 31 approximated by 10,000-sample Monte Carlo simulations. In the captions, SRP stands for solar radiation pressure. The distribution becomes more non-convex, non-Gaussian, and long-tailed as time increases. For comparison purposes, Figure 2(c) shows the Monte Carlo samples at the end of the simulation (Day 31) propagated in the presence of solar radiation pressure. Figure 2(d) shows the Monte Carlo samples at the end of the simulation (Day 31) propagated in the absence of solar radiation pressure. Although Figure 2(c) and Figure 2(d) appear similar, the accuracies of the mean and covariances corresponding to them are noticeably different, as shown in Table 6 and 7, which will be discussed in more detail.

The simulation results are summarized in four tables. Only an important subset of the means, variances, and covariances is included. Tables 4, 5, and 6 give the position means $\hat{x}$, $\hat{y}$, $\hat{z}$, the position variances P_x, P_y, P_z, and the covariance of P_{xy} using the Monte Carlo method, the quasi-Monte Carlo method, and the sparse grid method after 1 day, 7 days, and 31 days of propagation in the presence of solar radiation pressure. The position means $\hat{x}$, $\hat{y}$, $\hat{z}$, the position variances P_x, P_y, P_z, and the covariance of P_{xy} using the three methods after 31 days of propagation in the absence of solar radiation pressure are given in Table 7. In all the tables, MC(N) and QMC(N) stands for the Monte Carlo and quasi-Monte Carlo methods with N random samples. SG(L, g) stands for the

sparse grid method with accuracy level L and growth rule g. The true values of $\hat{x}$, $\hat{y}$, $\hat{z}$, P_x, P_y, P_z, and P_{xy} are unknown.

Table 4. Mean and Covariance of Position (1 day, with solar radiation pressure)

	$\hat{x}$ (km)	$\hat{y}$ (km)	$\hat{z}$ (km)	P_x (km^2)	P_y (km^2)	P_{xy} (km^2)	P_z (km^2)
MC(1e2)	41932	2318.2	-0.16221	226.04	69429	-3933.3	1.8177
MC(1e3)	41935	2256.6	0.028427	211.54	69424	-3779.5	1.5747
MC(1e4)	41934	2275.9	-0.015969	200.27	65342	-3571.5	1.5151
MC(1e5)	41934	2272.6	0.00010383	201.16	65781	-3591.1	1.5675
QMC(1e2)	41934	2271.2	0.014888	204.83	66254	-3630.7	1.5399
QMC(1e3)	41934	2271.0	-0.0018363	199.97	65453	-3573.8	1.5510
QMC(1e4)	41934	2271.3	-5.3804e-05	201.62	65842	-3597.2	1.5533
QMC(1e5)	41934	2271.3	4.6371e-05	201.26	65854	-3594.1	1.5510
SG(2,1)	41949	2268.2	-0.0001637	-72.199	65865	-3527.0	1.5410
SG(2,2)	41937	2270.5	-0.00035145	186.83	65939	-3577.5	1.5409
SG(2,3)	41937	2270.5	-0.00035145	186.83	65939	-3577.5	1.5409
SG(2,4)	41937	2270.5	-0.00035145	186.83	65939	-3577.5	1.5409
SG(2,5)	41937	2270.5	-0.00035145	186.83	65939	-3577.5	1.5409
SG(2,6)	41937	2270.5	-0.00035145	186.83	65939	-3577.5	1.5409
SG(3,1)	41911	2274.2	0.016371	-247.48	65907	-3542.8	1.5486
SG(3,2)	41933	2271.8	0.0029447	204.12	65850	-3593.4	1.5472
SG(3,3)	41936	2272.1	0.0040376	202.17	65782	-3598.2	1.5472
SG(3,4)	41933	2271.8	0.0029447	204.12	65850	-3593.4	1.5472
SG(3,5)	41937	2270.9	0.0030722	188.12	65819	-3592.5	1.5472
SG(3,6)	41937	2270.9	0.0030722	188.12	65819	-3592.5	1.5472
SG(4,1)	41963	2270.9	-0.051071	-703.69	65699	-3568.7	1.5384
SG(4,2)	41937	2271.8	-0.00039400	196.99	65807	-3595.3	1.5483
SG(4,3)	41935	2270.3	-0.0073919	195.09	65879	-3592.4	1.5484
SG(4,4)	41938	2270.6	-0.0013594	181.42	65843	-3588.3	1.5483
SG(4,5)	41933	2271.7	-0.0058776	206.10	65838	-3595.5	1.5480
SG(4,6)	41935	2271.3	-0.0058149	196.83	65857	-3590.9	1.5480
SG(5,1)	41905	2268.6	0.055128	-494.90	66053	-3676.4	1.5714
SG(5,2)	41934	2271.6	-0.0049119	204.88	65839	-3595.9	1.5479
SG(5,3)	41934	2271.0	0.011090	200.35	65864	-3590.6	1.5463
SG(5,4)	41933	2271.8	-0.004363	205.84	65834	-3595.3	1.5474
SG(5,5)	41935	2270.6	0.0089674	192.02	65887	-3586.0	1.5473
SG(5,6)	41934	2270.8	0.012574	199.35	65849	-3592.6	1.5579

The comparison between the Monte Carlo and the quasi-Monte Carlo results shows that the quasi-Monte Carlo method has better repeatability and less random variation than the Monte Carlo method. For example, in Table 5, the maximum difference among QMC(N), $N =$ 1e2, 1e3, 1e4, 1e5, in $\hat{x}$, $\hat{y}$, and $\hat{z}$ is 3 km. The difference in $\hat{x}$, $\hat{y}$, and $\hat{z}$ between QMC(1e4) and QMC(1e5) is below

Table 5. Mean and Covariance of Position (7 days, with solar radiation pressure)

	$\hat{x}$ (km)	$\hat{y}$ (km)	$\hat{z}$ (km)	P_x (km^2)	P_y (km^2)	P_{xy} (km^2)	P_z (km^2)
MC(1e2)	39088	15194	0.41207	4.0176e+05	2.8075e+06	-1.0554e+06	17.395
MC(1e3)	38952	15536	0.14438	4.5133e+05	2.7971e+06	-1.1195e+06	24.882
MC(1e4)	38958	15522	0.047599	4.5554e+05	2.8251e+06	-1.1297e+06	26.818
MC(1e5)	38961	15517	0.024334	4.4866e+05	2.7961e+06	-1.1154e+06	26.336
QMC(1e2)	38960	15518	-0.013314	4.5507e+05	2.8062e+06	-1.1251e+06	26.229
QMC(1e3)	38961	15517	-0.0013972	4.4492e+05	2.7738e+06	-1.1066e+06	26.628
QMC(1e4)	38960	15520	0.0081169	4.4810e+05	2.7908e+06	-1.1137e+06	26.591
QMC(1e5)	38960	15520	0.0084614	4.4794e+05	2.7912e+06	-1.1136e+06	26.573
SG(2,1)	39078	15468	0.019931	4.4024e+05	2.7918e+06	-1.1105e+06	26.360
SG(2,2)	38979	15509	0.0081163	4.5255e+05	2.7959e+06	-1.1157e+06	26.359
SG(2,3)	38979	15509	0.0081163	4.5255e+05	2.7959e+06	-1.1157e+06	26.359
SG(2,4)	38979	15509	0.0081163	4.5255e+05	2.7959e+06	-1.1157e+06	26.359
SG(2,5)	38979	15509	0.0081163	4.5255e+05	2.7959e+06	-1.1157e+06	26.359
SG(2,6)	38979	15509	0.0081163	4.5255e+05	2.7959e+06	-1.1157e+06	26.359
SG(3,1)	38782	15587	0.0036808	3.9937e+05	2.7852e+06	-1.0980e+06	26.375
SG(3,2)	38946	15527	0.0096923	4.4526e+05	2.7901e+06	-1.1121e+06	26.503
SG(3,3)	38971	15522	0.016335	4.4662e+05	2.7873e+06	-1.1116e+06	26.503
SG(3,4)	38946	15527	0.0096923	4.4526e+05	2.7901e+06	-1.1121e+06	26.503
SG(3,5)	38980	15512	0.011989	4.4629e+05	2.7884e+06	-1.1115e+06	26.503
SG(3,6)	38980	15512	0.011989	4.4629e+05	2.7884e+06	-1.1115e+06	26.503
SG(4,1)	39170	15456	0.0032131	4.2629e+05	2.7884e+06	-1.1047e+06	26.744
SG(4,2)	38979	15517	0.012765	4.4998e+05	2.7893e+06	-1.1136e+06	26.560
SG(4,3)	38965	15511	0.0015761	4.4693e+05	2.7900e+06	-1.1133e+06	26.545
SG(4,4)	38988	15507	0.0084186	4.4952e+05	2.7903e+06	-1.1134e+06	26.560
SG(4,5)	38953	15522	0.007633	4.4583e+05	2.7890e+06	-1.1124e+06	26.514
SG(4,6)	38964	15517	0.0093389	4.4583e+05	2.7894e+06	-1.1121e+06	26.514
SG(5,1)	38749	15566	0.025408	3.8055e+05	2.7846e+06	-1.0990e+06	26.361
SG(5,2)	38955	15522	0.0068598	4.4613e+05	2.7887e+06	-1.1126e+06	26.529
SG(5,3)	38957	15520	0.015636	4.4685e+05	2.7916e+06	-1.1126e+06	26.507
SG(5,4)	38952	15524	0.0085700	4.4497e+05	2.7887e+06	-1.1118e+06	26.498
SG(5,5)	38966	15515	0.014334	4.4817e+05	2.7921e+06	-1.1134e+06	26.543
SG(5,6)	38959	15519	0.017133	4.4725e+05	2.7906e+06	-1.1130e+06	26.532

1 km. The maximum difference among MC in $\hat{x}$, $\hat{y}$, and $\hat{z}$ is over 300 km and the difference in $\hat{x}$, $\hat{y}$, and $\hat{z}$ between MC(1e4) and MC(1e5) is as large as 5 km. The small variation in the quasi-Monte Carlo results is probably because the quasi-Monte Carlo method is more "deterministic" than the Monte Carlo method.

In Tables 4, 5, and 6, the results yielded by the sparse grid methods with the same accuracy level are noticeably different. In particular, SG(L, 1) yields very different results from the others and

SG(2,1) yields negative variances in Table 4. Were the mapping from the sparse grid at the initial time to $\mathbf{x}_t$ a polynomial function, the sparse grids with the same sufficiently high accuracy level would yield identical results. When that is not the case, however, the weights of the sparse grids, among other factors, play a role. The sparse grid results suggest that large discrepancy among the weights tends to result in less accurate results. Growth Rules 3 ("moderate linear"), 5 ("moderate exponential"), and 6 ("full exponential") use more balanced weights and are better behaved than

Table 6. Mean and Covariance of Position (31 days, with solar radiation pressure)

	$\hat{x}$ (km)	$\hat{y}$ (km)	$\hat{z}$ (km)	P_x (km^2)	P_y (km^2)	P_{xy} (km^2)	P_z (km^2)
MC(1e2)	-5413.4	40881	-0.25431	5.7481e+07	2.3073e+06	7.1406e+06	180.81
MC(1e3)	-4614.8	40940	-0.59945	6.1033e+07	2.0801e+06	7.1477e+06	171.57
MC(1e4)	-4597.3	40923	-0.15281	6.2499e+07	1.9751e+06	7.0966e+06	177.07
MC(1e5)	-4606.6	40940	0.11458	6.1084e+07	1.9264e+06	6.8703e+06	177.02
QMC(1e2)	-4613.7	40929	-0.047136	6.1546e+07	2.1340e+06	7.4863e+06	182.19
QMC(1e3)	-4604.9	40943	0.061218	6.0960e+07	1.8574e+06	6.8131e+06	177.78
QMC(1e4)	-4615.3	40939	0.083320	6.1188e+07	1.9223e+06	6.8523e+06	177.20
QMC(1e5)	-4614.9	40939	0.084064	6.1151e+07	1.9227e+06	6.8371e+06	177
SG(2,1)	-3882.5	40908	0.15625	6.1471e+07	8.1460e+05	8.1592e+06	183.23
SG(2,2)	-4487.4	40938	0.086097	6.0797e+07	1.9596e+06	7.3463e+06	183.25
SG(2,3)	-4487.4	40938	0.086097	6.0797e+07	1.9596e+06	7.3463e+06	183.25
SG(2,4)	-4487.4	40938	0.086097	6.0797e+07	1.9596e+06	7.3463e+06	183.25
SG(2,5)	-4487.4	40938	0.086097	6.0797e+07	1.9596e+06	7.3463e+06	183.25
SG(2,6)	-4487.4	40938	0.086097	6.0797e+07	1.9596e+06	7.3463e+06	183.25
SG(3,1)	-5707.5	40969	0.083257	6.1055e+07	1.9076e+06	4.9906e+06	176.53
SG(3,2)	-4700.2	40944	0.06294	6.1218e+07	1.9371e+06	6.5922e+06	176.86
SG(3,3)	-4548.9	40941	0.084136	6.1194e+07	1.9100e+06	6.7099e+06	176.86
SG(3,4)	-4700.2	40944	0.06294	6.1218e+07	1.9371e+06	6.5922e+06	176.86
SG(3,5)	-4492.5	40935	0.090372	6.1146e+07	1.9042e+06	6.7545e+06	176.86
SG(3,6)	-4492.5	40935	0.090372	6.1146e+07	1.9042e+06	6.7545e+06	176.86
SG(4,1)	-3292.9	40947	-0.0079472	5.7750e+07	2.0655e+06	9.3655e+06	176.98
SG(4,2)	-4499.0	40940	0.088469	6.0985e+07	1.9217e+06	7.0333e+06	176.90
SG(4,3)	-4579.0	40934	0.099255	6.1125e+07	1.9065e+06	6.7821e+06	176.88
SG(4,4)	-4442.7	40934	0.094705	6.0932e+07	1.9160e+06	7.0782e+06	176.90
SG(4,5)	-4655.7	40939	0.091833	6.1231e+07	1.9107e+06	6.6358e+06	176.83
SG(4,6)	-4590.3	40937	0.10785	6.1190e+07	1.9075e+06	6.6866e+06	176.83
SG(5,1)	-6030.4	40867	0.31084	6.0946e+07	1.5743e+06	3.7945e+06	177.09
SG(5,2)	-4644.6	40940	0.091378	6.1209e+07	1.9108e+06	6.6644e+06	176.87
SG(5,3)	-4636.8	40938	0.074089	6.1152e+07	1.9207e+06	6.7519e+06	176.81
SG(5,4)	-4664.9	40940	0.090191	6.1277e+07	1.9094e+06	6.5623e+06	176.82
SG(5,5)	-4573.6	40936	0.086774	6.1039e+07	1.9217e+06	6.9160e+06	176.85
SG(5,6)	-4620.1	40939	0.083463	6.1132e+07	1.9192e+06	6.7899e+06	176.84

the other rules. SG(6, 3) and SG(6, 6), SG(4,5) and SG(4,6), and SG(3,5) and SG(3,6) are in good agreement with each other and are considered the best sparse grid results. They agree with the best Monte Carlo and quasi-Monte Carlo results in around three significant decimal digits, but use fewer points. The current data are not sufficient to establish exactly which quadrature method is the most accurate, if there is one.

Table 7. Mean and Covariance of Position (31 days, without solar radiation pressure)

	$\hat{x}$ (km)	$\hat{y}$ (km)	$\hat{z}$ (km)	P_x (km^2)	P_y (km^2)	P_{xy} (km^2)	P_z (km^2)
MC(1e2)	-5075.4	41062	0.88145	4.9820e+07	1.6710e+06	6.9990e+06	165.41
MC(1e3)	-4053.4	41064	-0.36452	5.9219e+07	1.8624e+06	6.2715e+06	185.10
MC(1e4)	-4432.2	40996	0.18652	6.1337e+07	1.9534e+06	6.9119e+06	181.85
MC(1e5)	-4423.9	40999	0.073999	6.1220e+07	1.8661e+06	6.6414e+06	177.02
QMC(1e2)	-4466.5	41004	-0.088371	6.0392e+07	1.9456e+06	7.1170e+06	181.61
QMC(1e3)	-4462.0	41002	-0.0088917	6.0654e+07	1.8326e+06	6.6988e+06	177.96
QMC(1e4)	-4465.1	40997	0.00076831	6.1019e+07	1.8985e+06	6.7603e+06	177.38
QMC(1e5)	-4464.9	40997	-0.00019903	6.1010e+07	1.8981e+06	6.7500e+06	177.34
SG(2,1)	-4464.3	40992	0	6.2483e+07	7.9079e+05	7.0568e+06	183.70
SG(2,2)	-4464.8	40997	0	6.1013e+07	1.9017e+06	6.7556e+06	183.70
SG(2,3)	-4464.8	40997	0	6.1013e+07	1.9017e+06	6.7556e+06	183.70
SG(2,4)	-4464.8	40997	0	6.1013e+07	1.9017e+06	6.7556e+06	183.70
SG(2,5)	-4464.8	40997	0	6.1013e+07	1.9017e+06	6.7556e+06	183.70
SG(2,6)	-4464.8	40997	0	6.1013e+07	1.9017e+06	6.7556e+06	183.70
SG(3,1)	-4464.9	40997	0	6.0990e+07	1.9186e+06	6.7450e+06	177.19
SG(3,2)	-4464.9	40997	0	6.0990e+07	1.9181e+06	6.7453e+06	177.34
SG(3,3)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(3,4)	-4464.9	40997	0	6.0990e+07	1.9181e+06	6.7453e+06	177.34
SG(3,5)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(3,6)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(4,1)	-4464.9	40997	0	6.1011e+07	1.8987e+06	6.7500e+06	177.34
SG(4,2)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(4,3)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(4,4)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(4,5)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(4,6)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(5,1)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(5,2)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(5,3)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(5,4)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(5,5)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34
SG(5,6)	-4464.9	40997	0	6.1010e+07	1.8989e+06	6.7499e+06	177.34

Tables 4, 5, and 6 show that the results of the variances P_x, P_y, P_z, and covariance P_{xy} have more variations than those of the means $\hat{x}$, $\hat{y}$, and $\hat{z}$. That is because the variances and covariances involve higher degrees of nonlinearity than the means. Tables 4, 5, and 6 also show that as the propagation time increases, the variations of the means, variances, and covariances increases. That means that the difficulty and complexity of estimating the moments accurately increases with the propagation time.

In the absence of solar radiation pressure, all the methods perform very well and are in high agreement, as shown by Table 7. In Table 7, $\hat{z}$ yielded by the sparse grid methods are exactly zero, because of the symmetry of the sparse grid and the fact that the orbit is constrained in the X-Y plane. SG(3,4) and SG(3,5) (or equivalently SG(3,6)) with 319 and 417 points, respectively, achieve the same accuracy as QMC(1e5). The sharp contrast between Table 7 and Table 6 indicates that the uncertainty propagation problem in the presence of solar radiation pressure is much more difficult than that in the absence of solar radiation pressure, because the former involves high-dimension integration and large attitude motion uncertainty.

Gaussian Mixture Model

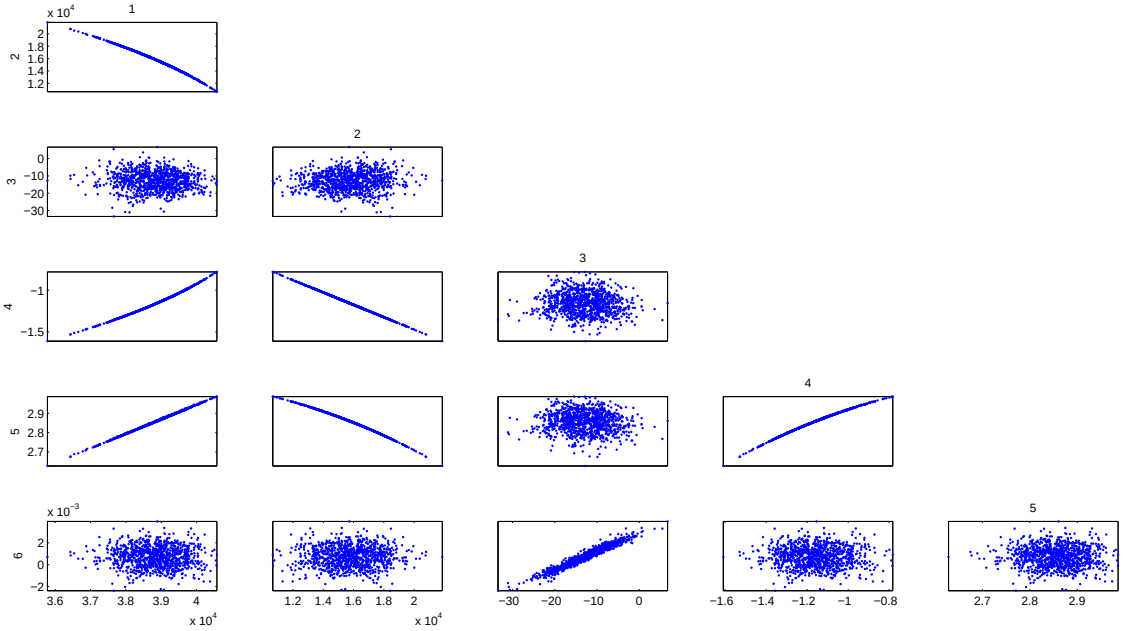


Figure 3. Scatter Plots of Position and Velocity: GEO Orbit

For the probability density estimation problem, the propagation time is limited to seven days. The result of a typical Monte Carlo run with 1000 random samples is shown in Fig. 3. Note that 1000 is a small sample size for the 12-dimensional problem. Only the position and velocity are shown and only the Gaussian mixture model for them is obtained. The Expectation-Maximization algorithm is used to get the Gaussian mixture model from the position and velocity parts of the random samples. The optimal number of Gaussian components in this run is 15 based the Bayesian

information criterion. The log-likelihood corresponding to the resultant Gaussian mixture model is 5536.96. The large value of the likelihood function indicates good agreement between the model and the random samples.

The result of the simulation based method is subject to Monte Carlo random variations. For example, with the same initial distribution, another run shows that the number of Gaussian components is 16 and the log-likelihood is 5698.47. Nevertheless, the Monte Carlo simulation based method is able to provide the correct "big picture" information of the probability distribution.

The generalized Polynomial Chaos based simulations with $3 \leq |\mathbf{i}| \leq 6$ are applied. They are computational more expensive (because the resultant number of sparse grid points $N^S > 1000$), but still much less accurate than the classic Monte Carlo simulation, because the polynomial approximation is poor for this 12-dimensional system with extremely large attitude uncertainty, and the attitude sparse grid used is insufficient for the uniform attitude distribution.

There are two main error sources for the simulation based method. The first is the random error in Monte Carlo simulation; the second is the estimation error in the Expectation-Maximization algorithm. The Expectation-Maximization algorithm has guaranteed convergence to the local optimum only. To reduce the error due to the Expectation-Maximization algorithm, one may run the Expectation-Maximization algorithm several times with different initial guesses and then choose the best result, which is more likely to be the global optimum. The Monte Carlo error is usually the larger error of the two, especially when a relative small number of random samples are used. Because of the Monte Carlo error, the simulation based method is more appropriate for providing the global picture or trend than providing accurate probability density estimation. The Monte Carlo error can be reduced by increasing the sample size, but quantizing the increased accuracy is not easy and problem dependent because little is not about the true $p_t(\mathbf{x}_t)$. The probability density estimation error can be characterized by the Kullback-Leibler divergence, given by

$$D = \int p_t(\mathbf{x}_t) \log \frac{p_t(\mathbf{x}_t)}{\hat{p}_t(\mathbf{x}_t)} \, d\mathbf{x}_t$$

which can be approximated by

$$\hat{D} = \frac{1}{N} \sum_{i=1}^{N} \log \frac{p_t(\mathbf{x}_t^{(i)})}{\hat{p}_t(\mathbf{x}_t^{(i)})}$$

where $\mathbf{x}_t^{(i)} \sim p_t(\mathbf{x}_t)$. The practical difficulty is that the numerical approximation to the Monte Carlo integration is subject to Monte Carlo random variations, too. It is estimated that over a million Monte Carlo samples are needed for evaluating the integral accurately.

The computational complexity of the simulation based method is mainly determined by that of Monte Carlo simulation, which is expensive especially when accurate probability density estimation is sought. When used to generate random samples, the generalized Polynomial Chaos may reduce the computational complexity by replacing orbit propagation with simpler polynomial evaluations. However, the polynomial approximation to the nonlinear mapping between the initial and final states is poor for long-term orbit propagation of the high area-to-mass-ratio space object in the absence of attitude knowledge.

CONCLUSIONS

Orbit uncertainty propagation of a tumbling, high area-to-mass-ratio space object under solar radiation pressure is challenging, because of the high dimensionality and large attitude motion un-

certainty. Moreover, the complexity of the problem increases with the propagation time. The Monte Carlo methods and sparse grid method were used to estimate the Gaussian mixture model, as well as the mean and covariance of the orbital parameters of the space object. These methods can be used with any orbit propagators, are straightforward to implement, and can handle both Gaussian and non-Gaussian uncertainty parameters of the orbit propagation model.

REFERENCES

[1] R. H. Lyon, *Geosynchronous Orbit Determination Using Space Surveillance Network Observations and Improved Radiactive Force Modeling*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[2] K. DeMars, M. Jah, D. Giza, and T. Kelecy, "Orbit Determination Performance Improvements for High Area-to-Mass Ratio Space Object Tracking Using an Adaptive Gaussian Mixtures Estimation Algorithm," *The 21st International Symposium on Space Flight Dynamics*, Toulouse, France, September 28 – October 2, 2009.

[3] K. J. DeMars, *Nonlinear Orbit Uncertainty Prediction and Rectification for Space Situational Awareness*. PhD thesis, The University of Texas at Austin, 2010.

[4] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Uncertainty Propagation for Nonlinear Dynamical Systems using Gaussian Mixture Models," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 6, 2008, pp. 1623–1633.

[5] K. J. DeMars, R. H. Bishop, and M. K. Jah, "A Splitting Gaussian Mixture Method for the Propagation of Uncertainty in Orbital Mechanics," *Advances in the Astronautical Sciences, AAS 11-201*, Vol. 140, 2011, pp. 1419–1438.

[6] R. S. Park and D. J. Scheeres, "Nonlinear Semi-Analytic Methods for Trajectory Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, 2007, pp. 1668–1676.

[7] R. S. Park and D. J. Scheeres, "Nonlinear Mapping of Gaussian Statistics: Theory and Applications to Spacecraft Trajectory Designs," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1367–1375.

[8] A. Prabhakar, J. Fisher, and R. Bhattacharya, "Polynomial Chaos-Based Analysis of Probabilistic Uncertainty in Hypersonic Flight Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, 2010, pp. 222–234.

[9] M. D. Nevels, B. Jia, M. R. Turnowicz, M. Xin, and Y. Cheng, "Sparse Grid-Based Orbit Uncertainty Propagation," *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Alaska, July 31-August 4, 2011.

[10] G. Terejanu, P. Singla, T. Singh, and P. Scott, "Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation," *IEEE Transactions on Automatic Control*, Vol. 56, No. 9, 2011, pp. 2151–2156.

[11] J. T. Horwood and A. B. Poore, "Adaptive Gaussian Sum Filters for Space Surveillance," *IEEE Transactions on Automatic Control*, Vol. 56, No. 8, 2011, pp. 1777 –1790.

[12] J. T. Horwood, N. D. Aragon, and A. B. Poore, "Gaussian Sum Filters for Space Surveillance: Theory and Simulations," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 6, 2011, pp. 1839–1851.

[13] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[14] F. Nobile, R. Tempone, and C. G. Webster, "A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data," *SIAM Journal of Numerical Analysis*, Vol. 46, 2008, pp. 2309–2345.

[15] F. Heiss and V. Winschel, "Likelihood Approximation by Numerical Integration on Sparse Grids," *Journal of Econometrics*, Vol. 144, No. 1, 2008, pp. 62–80.

[16] C. M. Bishop, *Pattern Recognication and Machine Learning*. New York, NY: Springer-Verlag, 2006.

[17] D. Xiu, *Numerical Methods for Stochastic Computations: Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton, NJ: Princeton University Press, 2010.

[18] B. Jia, M. Xin, and Y. Cheng, "Sparse Gauss-Hermite Quadrature Filter with an Application to Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp. 367–379.

[19] M. D. Shuster, "Uniform Attitude Probability Distributions," *The Journal of the Astronautical Sciences*, Vol. 51, No. 4, 2003, pp. 451–475.