(Preprint) AAS 25-776

TAYLOR PROPAGATION FOR ENSEMBLE BASED FILTERING

Felipe Giraldo-Grueso[*], **Alberto Fossà**[†], **Ryan J. Menges**[‡], **Daniel J. Scheeres**[§],
Renato Zanetti[¶]

The propagation step in spacecraft navigation can be computationally expensive when dealing with particles. This makes the implementation of nonlinear filters difficult for onboard applications. Studying fast and accurate propagation schemes is therefore crucial to enable the use of nonlinear filters onboard spacecraft. In this work, numerical propagation based on the Taylor method is compared with Runge-Kutta (RK)-type integrators in the context of the Earth-Moon system, modeled via the circular restricted three-body problem (CR3BP). The Taylor integrator is shown to be faster that the numerical propagators in this environment, without accuracy loss. In addition, the Taylor integrator is used for the propagation step in a relative navigation scenario, where the ensemble Gaussian mixture filter (EnGMF) is used to estimate the state of a chaser, in a quasi-periodic orbit (QPO), relative to Gateway, in the near rectilinear halo orbit (NRHO). In this example, the Taylor integrator shows to be faster than the RK-type integrators, and the runtime remains nearly constant when considering longer propagation horizons.

INTRODUCTION

In state estimation, the Bayesian recursive relations (BRRs) are solved to estimate the state probability density function (PDF).[1] These relations are composed of two steps. In the case of continuous dynamics, the first step deals with the propagation of the state PDF via the Fokker-Planck (FP) equation[2] (for discrete dynamics, the Chapman-Kolmogorov (CK) equation is used for the PDF propagation[1]). In this step, the state PDF is propagated to the next measurement epoch. After recording a measurement, the second step is to update the state PDF using Bayes' rule. If the dynamics and measurement models are linear and driven by zero-mean Gaussian additive noise, the BRRs can be analytically solved using the Kalman filter (KF).[1] When the dynamics become highly nonlinear and the state PDF starts to become non-Gaussian, the BRRs can become intractable. In this case, using linear filters, such as the KF, can result in loss of accuracy and consistency.

Most onboard navigation solutions use linear filters, such as the extended Kalman filter (EKF)[1] or the unscented Kalman filter (UKF),[3] to produce a fast approximation to the Bayesian estimation problem. Note that in this work, the linearity of a filter is defined with respect to the use of the

---

[*] Ph.D. Candidate, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712

[†] Postdoctoral Fellow, Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712.

[‡] Ph.D. Student, Department of Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO 80303.

[§] A. Richard Seebass Chair Professor, Department of Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO 80303.

[¶] Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712.

measurement in the update step. In space applications, these filters are typically coupled with numerical integrators, to propagate the previous state estimate to the current epoch. While this approach has been proven to be effective, the accuracy of the estimate could be improved by using nonlinear filters such as the particle filter (PF),[4] the ensemble Gaussian mixture filter (EnGMF),[5–7] or the point mass filter (PMF).[8–11] The challenge with nonlinear filters is that they rely on particles or grids, which can make the onboard numerical integration computationally unfeasible. Therefore, exploring different methodologies to accurately propagate the state in a rapid manner could enable the use of nonlinear filters for onboard spacecraft applications.

Sequential filters for onboard orbit determination (OD) typically rely on Runge-Kutta (RK) type numerical integrators for the propagation step.[12] These are explicit methods that come in two flavors: fixed or adaptive step size. The former are simpler to implement and require a known number of operations to propagate the state after selecting the step size. However, to preserve the accuracy, the step size must be tuned for the regions where the dynamics are faster, thus forcing the integrator to take redundant steps in the slower regions. This issue is addressed by adaptive step size integrators, which adjust the step size based on the local error, although at the expenses of a more complex algorithm. A different approach to numerical integration is the Taylor method.[13] In contrast to high-order RK integrators that evaluate the dynamics at multiple points within the same step, the Taylor method achieves the same accuracy by evaluating the higher-order derivatives of the dynamics at a single point. This is particularly advantageous at tighter tolerances, as the growth in the computational cost when moving to higher accuracies is slower compared to that of RK integrators.[13] These properties are demonstrated in this work where `heyoka`,[14] a highly optimized implementation of the Taylor method, is used to improve the performance of nonlinear sequential filters.

In cislunar space, where observations are sparse and the dynamics are highly nonlinear, using linear filters for onboard OD can lead to poor estimation performance.[6] Nonlinear filters can offer improved accuracy in such environments, though at the expense of increased computational demands due to the propagation of particles. Therefore, the adoption of these fast Taylor propagation techniques could yield benefits by allowing the use of nonlinear filters for onboard OD. In this work, the circular restricted three-body problem (CR3BP) is used to model the Earth-Moon system, focusing on spacecraft orbiting a $9:2$ resonant near rectilinear halo orbit (NRHO) and a nearby quasi-periodic orbit (QPO). This setup provides a challenging scenario for evaluating the performance of the Taylor integrator under highly nonlinear conditions. Additionally, its use within a nonlinear filter is validated through a relative navigation scenario, where linear filters are shown to diverge.

The remainder of this paper is organized as follows. First, the CR3BP is presented. Next, the Taylor method is discussed focusing on the characteristics of the chosen implementation, namely `heyoka`. The equations for the EnGMF are then presented. Subsequently, two numerical examples are presented. The first example compares the Taylor integrator with several state-of-the-art RK-type integrators. The second example validates the use of `heyoka` in a relative navigation scenario using the EnGMF. Finally, conclusions are drawn.

THE CIRCULAR RESTRICTED THREE BODY PROBLEM

The CR3BP is a dynamical model that describes the motion of an infinitesimally small body under the gravitational attraction of two massive bodies, known as primaries.[15] The two primaries have masses m_1 and m_2, respectively, where $m_1 \geq m_2$. These primaries are modeled as point masses, and assumed to move in a circular orbit about their common barycenter. The third body has mass $m_3 \ll m_1, m_2$, and does not affect the motion of the primaries. The system is completely described by the mass parameter μ, which is defined as the ratio between m_2 and the sum of the primaries' masses, i.e.

$$\mu = \frac{m_2}{m_1 + m_2}, \tag{1}$$

where $0 < \mu \leq 1/2$ since $m_1 \geq m_2$.

The motion of m_3 is studied in a rotating frame, the synodic frame, in which the two primaries remain stationary. The frame's origin O coincides with the barycenter of m_1 and m_2, while its x axis is along the $P_1 - P_2$ segment, pointing towards P_2, where P_1 and P_2 are the location of the primaries. The z axis is aligned with the primaries' angular momentum vector, $\boldsymbol{\omega}$, and the y axis completes the right-handed triad. The construction of the frame is shown in Fig. 1 together with the inertial frame O_{XYZ}. Capital letters denote the axes of the inertial frame, while lowercase letters denote those of the rotating frame. Both frames share the same origin O.

Units are defined such that the distance between the primaries is equal to one and the period of their orbit is equal to 2π. Using these units, the synodic frame rotates with respect to the inertial frame with constant angular velocity $\boldsymbol{\omega} = 1\hat{\boldsymbol{k}}$, where $\hat{\boldsymbol{k}}$ is the unit vector along the Z axis. The primaries are located at $P_1 = (-\mu, 0, 0)$ and $P_2 = (1 - \mu, 0, 0)$, respectively, while the third mass is located at P_3. The position of the latter is $\boldsymbol{r} = [x\ y\ z]^{\mathrm{T}}$, and its velocity is $\boldsymbol{v} = [\dot{x}\ \dot{y}\ \dot{z}]^{\mathrm{T}}$. The vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ denote the position of P_3 relative to P_1 and P_2, respectively.

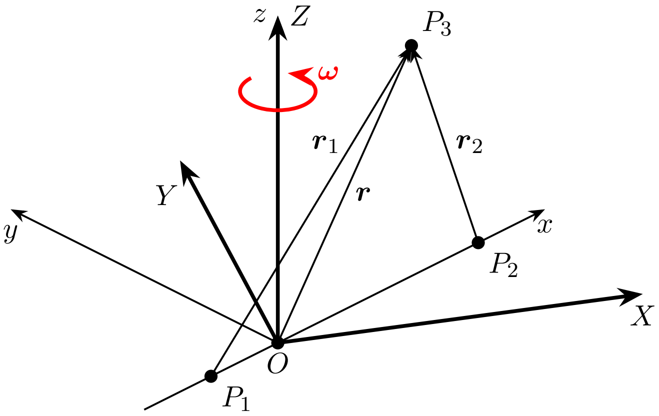


Figure 1: Definition of the CR3BP synodic frame.

The gravitational potential generated by the two primaries on P_3 is given by

$$U(x, y, z) = -\frac{1 - \mu}{r_1} - \frac{\mu}{r_2}, \tag{2}$$

where r_1 and r_2 are given by

$$\begin{aligned} r_1 &= \|\boldsymbol{r_1}\| = \sqrt{(x + \mu)^2 + y^2 + z^2}, \\ r_2 &= \|\boldsymbol{r_2}\| = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}. \end{aligned} \tag{3}$$

The equations of motion (EOMs) that govern the motion of m_3 are then expressed in the synodic frame as[15]

$$\begin{aligned} \ddot{x} - 2\dot{y} &= -\frac{\partial \bar{U}}{\partial x}, \\ \ddot{y} + 2\dot{x} &= -\frac{\partial \bar{U}}{\partial y}, \\ \ddot{z} &= -\frac{\partial \bar{U}}{\partial z}, \end{aligned} \tag{4}$$

with $\bar{U}\left(x, y, z\right)$ the *augmented* or *effective potential* given by

$$\bar{U}\left(x, y, z\right) = -\frac{1}{2}\left(x^2 + y^2\right) + U\left(x, y, z\right). \tag{5}$$

Equation (4) is a system of three second-order ordinary differential equations (ODEs) that must be numerically integrated with known initial conditions (ICs) to predict the motion of m_3 over time. Since the EOMs in Eq. (4) are Hamiltonian and independent of time, they admit an energy integral of motion E. However, the Jacobi constant $C = -2E$ is used more frequently.[15] It is computed as

$$C\left(x, y, z, \dot{x}, \dot{y}, \dot{z}\right) = -\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) - 2\bar{U}\left(x, y, z\right). \tag{6}$$

TAYLOR INTEGRATION

Numerical integration methods based on Taylor series expansions were first developed to study the dynamics of beams in particle accelerators.[16] These techniques have since been adopted to solve different initial value problems (IVPs) in astrodynamics, including the CR3BP,[13] the N-body problem,[14,17] and the dynamics around irregularly-shaped bodies such as asteroids and comets.[14] Similarly to RK methods, the Taylor integrator is an explicit method. It differs however in the way each step is computed, and, for variable step size integrators, in the way the error is controlled.

Consider the IVP

$$\begin{cases} \dot{\boldsymbol{x}}\left(t\right) & = \boldsymbol{f}\left(\boldsymbol{x}, t\right), \\ \boldsymbol{x}\left(t_0\right) & = \boldsymbol{x}_0, \end{cases} \tag{7}$$

with $\boldsymbol{x} \in \mathbb{R}^n$ the state vector, $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ the dynamics, $t \in \mathbb{R}$ the time, $\boldsymbol{x}_0$ the ICs, and t_0 the initial time. Given the state $\boldsymbol{x}(t_k)$ at time t_k, the Taylor integrator computes the state at time $t_{k+1} = t_k + h_k$ as[13]

$$\boldsymbol{x}\left(t_{k+1}\right) = \sum_{n=0}^{p} \frac{1}{n!}\boldsymbol{x}^{(n)}\left(t_k\right) h_k^n, \tag{8}$$

where $\boldsymbol{x}^{(n)}(t_k)$ is the n-th time derivative of the state evaluated at t_k, h_k is the current step size, and $p > 0$ is the order of the integrator. Note that for $p = 1$ Eq. (8) reduces to the Euler method. From Eq. (7) the following relation holds

$$\boldsymbol{x}^{(n)}\left(t\right) = \boldsymbol{f}^{(n-1)}\left(\boldsymbol{x}(t), t\right) \qquad n > 0, \tag{9}$$

i.e. the n-th time derivative of the state coincides with the $(n-1)$-th time derivative of the dynamics $\forall t \in \mathbb{R}$ and $n > 0$. From Eqs. (8) and (9) follows that the dynamics must be at least $p - 1$ times differentiable, and that these derivatives must be evaluated repeatedly to solve Eq. (7). If the first requirement is intrinsically related to the problem being solved, the second one is efficiently addressed by using automatic differentiation (AD) techniques. Contrarily to other numerical differentiation techniques such as finite differences methods, AD provides machine-precision accuracy with no computational overhead compared to the evaluation of the analytical derivatives. Given its advantages, this technique is at the core of most modern Taylor integrators,[13,14,17] including the integrator used in this work.

The Taylor integration method is a variable order integrator and there are two independent parameters that can be tuned to control its error: the order p and the step size h_k. If the same accuracy can be obtained for different combinations of these parameters, it is of interest to select the combination that minimizes the number of operations required to integrate the dynamics over a given time span. Denoting with ε_m the maximum allowed absolute truncation error, the optimal order p_m satisfies[13]

$$p_m = \left\lceil -\frac{1}{2}\log\varepsilon_m + 1 \right\rceil , \tag{10}$$

where $\lceil\cdot\rceil$ denotes the ceiling function. The corresponding step size h_k at time t_k is then computed as

$$h_k = \frac{\rho_{m,k}}{e^2}\exp\left(-\frac{0.7}{p_m - 1}\right), \tag{11}$$

with e denoting Euler's number and $\rho_{m,k}$ an estimation of the smallest radius of convergence among the Taylor series of all state variables at time t_k.[14] Although Eq. (11) is not an explicit function of ε_m, the step size h_k still depends on the error tolerance through the order p_m. One advantage of the Taylor integrator is that p_m can be much larger than the order of typical RK methods. This makes the Taylor method particularly suitable for high-accuracy applications, as its runtime is less affected by tighter tolerances where higher-order expansions are exploited to limit the number of steps.[13]

Motivated by the previous points, this work uses `heyoka`,[14] an adaptive Taylor integrator. This integrator is implemented in `C++` and includes the AD framework for the computation of the higher-order derivatives of the dynamics. The code is open-source[*] and exposed to Python through the `pybind11` library[†]. A Python package that depends on `heyoka`'s Python wrapper was developed and used in this work to build the adaptive integrator for the CR3BP dynamics. As the examples in this work are implemented in `MATLAB`, this integrator is called directly from the `MATLAB` environment using its native Python interface.

ENSEMBLE GAUSSIAN MIXTURE FILTER

As mentioned previously, the EnGMF has been shown to outperform linear filters, such as the UKF.[6] The EnGMF can be categorized as a hybrid PF coupled with ideas from the Gaussian sum filter (GSF).[18,19] Instead of propagating multiple Gaussian distributions, which can be expensive, the EnGMF propagates particles and uses kernel density estimation (KDE) techniques[20] to build a Gaussian mixture (GM) from the propagated particles. For this section, let the dynamics be expressed as

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k\left(\boldsymbol{x}_k\right) + \boldsymbol{\nu}_k, \tag{12}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the state at time step k, $\boldsymbol{F}_k : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is the flow of Eq. (7) obtained either by numerical integration or via the Taylor propagator, and $\boldsymbol{\nu}_k \in \mathbb{R}^n$ is additive white Gaussian process noise. In addition, let the measurement model be described as

$$\boldsymbol{y}_k = \boldsymbol{h}_k\left(\boldsymbol{y}_k\right) + \boldsymbol{w}_k, \tag{13}$$

---

[*] `https://github.com/bluescarni/heyoka`

[†] `https://github.com/bluescarni/heyoka.py`

with $\boldsymbol{y}_k \in \mathbb{R}^s$ the measurement at time step k, $\boldsymbol{h}_k : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^s$ the measurement model, and $\boldsymbol{w}_k \in \mathbb{R}^s$ additive white Gaussian measurement noise.

The EnGMF is composed of four steps.[6] The first step is to propagate the previous posterior particles, which represent the state PDF at time step k as a Dirac mixture (DM)

$$p(\boldsymbol{x}_k|\boldsymbol{y}_k) \approx \frac{1}{N}\sum_{i=1}^{N} \delta\left(\boldsymbol{x}_k - \boldsymbol{\mathcal{X}}_k^{+(i)}\right), \tag{14}$$

where $\boldsymbol{\mathcal{X}}_k^{+(i)}$ represents the i-th posterior particle at time step k and N is the total number of particles. The particles are propagated to the next time step $k+1$, such that

$$\boldsymbol{\mathcal{X}}_{k+1}^{-(i)} = \boldsymbol{F}_k\left(\boldsymbol{\mathcal{X}}_k^{+(i)}\right) + \boldsymbol{\nu}_k^{(i)}, \tag{15}$$

where $\boldsymbol{\mathcal{X}}_{k+1}^{-(i)}$ is the i-th prior particle at time step $k+1$ and $\boldsymbol{\nu}_k^{(i)}$ is the i-th process noise sample at time step k. With the propagated particles, the propagated PDF can be described as

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k) \approx \frac{1}{N}\sum_{i=1}^{N} \delta\left(\boldsymbol{x}_{k+1} - \boldsymbol{\mathcal{X}}_{k+1}^{-(i)}\right). \tag{16}$$

The second step is to build a GM from the DM in Eq. (16) using KDE techniques

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k) \approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}\left(\boldsymbol{x}_{k+1}; \boldsymbol{\mathcal{X}}_{k+1}^{-(i)}, \beta_{\text{sil}}^2 \hat{\boldsymbol{P}}_{k+1}\right), \tag{17}$$

where

$$\hat{\boldsymbol{P}}_{k+1} = \left[\boldsymbol{\mathcal{X}}_{k+1}^{-(i)} - \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\mathcal{X}}_{k+1}^{-(i)}\right][\cdot]^{\mathrm{T}}, \tag{18}$$

and β_{sil}^2 is given by Silverman's rule of thumb[20]

$$\beta_{\text{sil}}^2 = \left(\frac{4}{N\,(n+2)}\right)^{\frac{2}{n+4}}. \tag{19}$$

The third step is to update the GM in Eq. (17) once a measurement ($\boldsymbol{y}_{k+1}$) is obtained. To update the GM, a GSF update is performed[18,19] such that

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{k+1}) \approx \sum_{i=1}^{N} \widetilde{\boldsymbol{w}}_{k+1}^{(i)} \mathcal{N}\left(\boldsymbol{x}_{k+1}; \widetilde{\boldsymbol{\mathcal{X}}}_{k+1}^{+(i)}, \widetilde{\boldsymbol{P}}_{k+1}^{(i)}\right), \tag{20}$$

where $\widetilde{\boldsymbol{\mathcal{X}}}_{k+1}^{+(i)}$ and $\widetilde{\boldsymbol{P}}_{k+1}^{(i)}$ are obtained by performing a Kalman update on each component, and the weights, $\widetilde{\boldsymbol{w}}_{k+1}^{(i)}$, are proportional to the probability of the measurement, i.e.

$$\widetilde{\boldsymbol{w}}_{k+1}^{(i)} \propto \mathcal{N}\left(\boldsymbol{y}_{k+1}; \boldsymbol{h}_{k+1}\left(\boldsymbol{\mathcal{X}}_{k+1}^{-(i)}\right), \boldsymbol{W}_{k+1}^{(i)}\right), \tag{21}$$

where $\boldsymbol{W}_{k+1}^{(i)}$ is the i-th innovation covariance. Note that the first two moments of Eq. (20) can be obtained using the following formulation

$$\boldsymbol{x}_{k+1}^{+} = \sum_{i=1}^{N} \widetilde{\boldsymbol{w}}_{k+1}^{(i)} \widetilde{\boldsymbol{\mathcal{X}}}_{k+1}^{+(i)}, \tag{22}$$

$$\boldsymbol{P}_{k+1}^{+} = \sum_{i=1}^{N} \widetilde{\boldsymbol{w}}_{k+1}^{(i)} \left(\widetilde{\boldsymbol{P}}_{k+1}^{(i)} + \left[\widetilde{\boldsymbol{\mathcal{X}}}_{k+1}^{+(i)} - \boldsymbol{x}_{k+1}^{+} \right] [\cdot]^{\mathrm{T}} \right). \tag{23}$$

The fourth and final step is to resample particles ($\boldsymbol{\mathcal{X}}_{k+1}^{+}$) from the GM in Eq. (20), to return to a DM approximation as in Eq. (14). Sampling points from a GM is not a straightforward process. The canonical EnGMF samples points randomly.[5,6] Other strategies have been implemented to sample deterministic points, although with a higher algorithmic complexity.[21] This work adheres to the canonical formulation and samples points randomly. Figure 2 shows a flowchart of the four steps in the EnGMF.

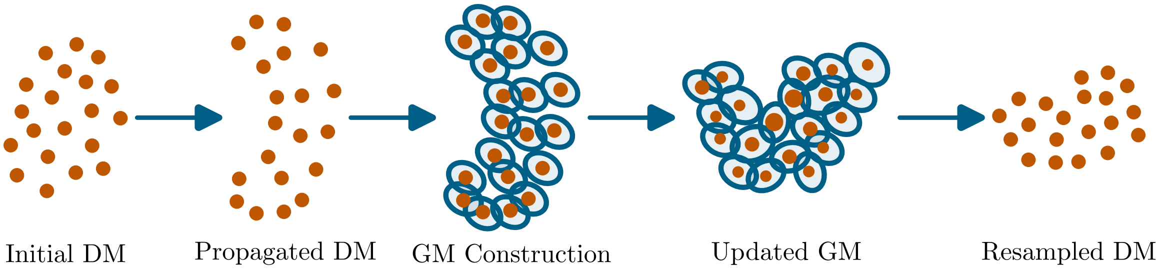


Figure 2: Flowchart of the four steps in the EnGMF.

Note that the most computationally expensive parts of this algorithm are the propagation of the particles in Eq. (15) and the Kalman update of each of the Gaussian components in Eq. (17). This work focuses on comparing different propagation methods, as presented in the previous section, in order to study their effect when using them within the EnGMF framework.

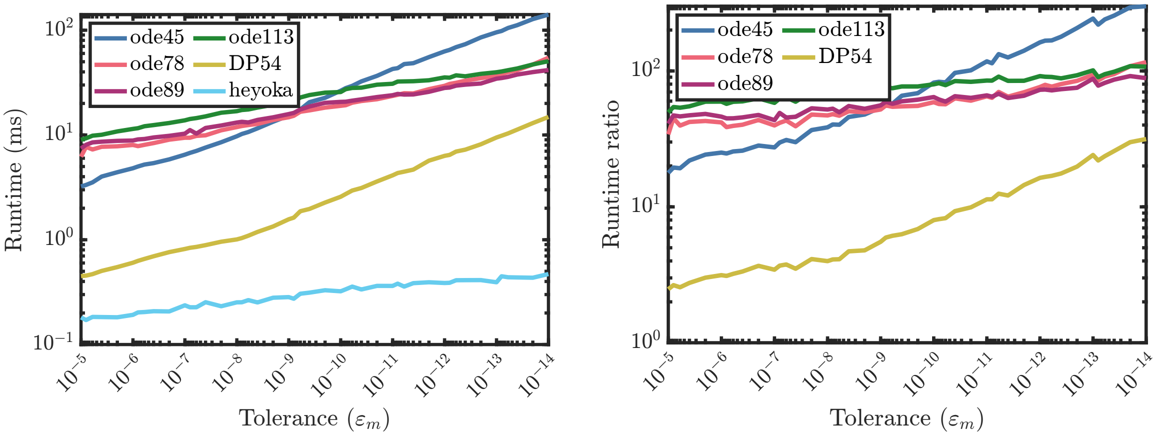
NUMERICAL EXAMPLES

This section presents two numerical experiments. The first example compares the performance of the Taylor integrator, `heyoka`, with other state-of-the-art numerical integrators. The second example compares the impact of different integrators when implemented in the EnGMF to perform relative navigation. Both consider spacecraft in the cislunar region and model the dynamics with the CR3BP.

Integrator Performance

To assess the performance of `heyoka`, the latter is compared against the numerical integrators for non-stiff ODEs available in `MATLAB`, namely `ode45`,[22,23] `ode78`,[24] `ode89`,[24] and `ode113`.[23,25] A `C++` implementation of the Dormand-Prince (DP)54 method, generated from `MATLAB` source code using the `MATLAB` Coder, is also included in the comparison. Note that these integrators are coded in different programming languages. The `MATLAB` integrators are expected to be the slowest since the code is interpreted. However, both `heyoka` and the DP54 method are implemented in `C++` and thus compiled ahead-of-time (AOT). Moreover, the expression needed to evaluate the right-hand side (RHS) of Eq. (8) are computed and compiled just-in-time (JIT) when the propagator is called for the first time, and then cached for subsequent use.*

These integrators are tested on the CR3BP dynamics to measure their runtime and their consistency for different tolerances. The ICs used in this test correspond to the apolune of the $9{:}2$ resonant L_2 southern NRHO in the Earth-Moon system. These ICs are propagated for 9 revolutions, or about 59 days, and the simulations are repeated 300 times to obtain statistically significant results. The average runtime for each method, as well as the runtime relative to `heyoka`, are shown in Fig. 3.



(a) Absolute runtime for all integrators.

(b) Runtime relative to `heyoka`.

Figure 3: Absolute runtime for all integrators (left) and runtime relative to `heyoka` for different tolerances (right).

Regarding the `MATLAB` integrators, Fig. 3a shows that `ode45` is the fastest method at looser tolerances. However, higher-order methods become more efficient for tolerances tighter than 10^{-10}, as their runtime is less affected by the error threshold. The `C++` implementation of the DP54 method shows the same trend as `ode45`, but it is also one order of magnitude faster than the latter across all tolerances due to being compiled. Finally, `heyoka` is the fastest among all tested methods, and its performance is the least affected by the tolerance due to its variable order implementation. Figure 3b shows that the Taylor integrator is 2.5 times faster than the `C++` DP54 method for $\varepsilon_m = 10^{-5}$, and over 30 times faster for $\varepsilon_m = 10^{-14}$. As expected, the gap between `heyoka` and the `MATLAB` integrators is even larger, with the former being 20 to 90 times faster than the best performing `MATLAB` alternative at each tolerance.

The accuracy of each method is assessed by comparing the Jacobi constant of the ICs with that of the final state output by the corresponding integrator. This constant, defined in Eq. (6), is the only integral of motion in the CR3BP. Figure 4 shows the Jacobi constant error for the different integrators with different tolerances. From this figure, it can be seen that `heyoka` is the most accurate across all tolerances, closely followed by `MATLAB`'s `ode89` and `ode78`. The `ode113` integrator performs relatively the worst, while `ode45` and the DP54 integrator perform equally in between.

---

*If anything, there is a clear overhead when calling the Python wrapper for `heyoka` from `MATLAB`.

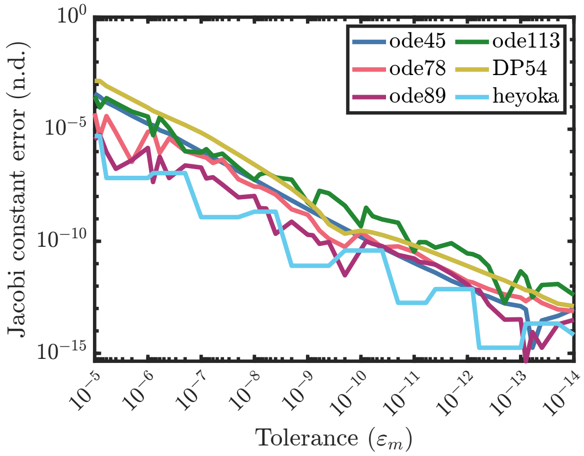


Figure 4: Jacobi constant error as a function of tolerance for all integrators.

Relative Navigation

To study the effect that different propagators can have on the propagation step for the EnGMF, a relative navigation example is presented.[26] In this example, a target spacecraft is modeled as NASA's Gateway on a $9:2$ resonant NRHO. The chaser spacecraft, whose orbit is unknown and estimated, resides in cislunar space near Gateway's NRHO. The objective of this relative navigation example is to estimate the absolute state of the chaser relative to the Earth-Moon barycenter using relative measurements. These relative measurements are simulated to be obtained onboard the chaser by observing the target.

For the specific scenario implemented here, Gateway is modeled on a $9:2$ NRHO and the chaser is on a QPO, both simulated using the CR3BP formulation. The QPO is two-dimensional (in frequency) and corresponds to bounded non-periodic motion occurring on the surface of an invariant torus near a periodic orbit possessing oscillatory modes.[26] A two-dimensional QPO is defined by two fundamental frequencies, and in this implementation, the chaser's QPO is bounded relative to Gateway's NRHO, with one of its fundamental frequencies matching the orbital period of the NRHO.[26] The QPO of the chaser and the NRHO of Gateway are shown in Fig. 5.

A geometric representation of the relative dynamical model is shown in Fig. 6. In this figure, the vectors $\boldsymbol{r}_G$ and $\boldsymbol{r}$ represent the position vectors of Gateway and the chaser, respectively, relative to the barycenter of the Earth-Moon system. The vector $\boldsymbol{\rho}$, which is defined as

$$\boldsymbol{\rho} = \boldsymbol{r}_G - \boldsymbol{r}, \tag{24}$$

represents the relative position between the two spacecraft. Using the notation from the geometric representation shown in Fig. 6, the state for the chaser, estimated using the different filtering methods, is defined as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^{\mathrm{T}}, \tag{25}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ are the position and velocity vectors, relative to the Earth-Moon barycenter, of the chaser. Within the estimation framework, the state for the target, Gateway, is defined as

$$\boldsymbol{x}_G = \begin{bmatrix} \boldsymbol{r}_G \\ \boldsymbol{v}_G \end{bmatrix} = \begin{bmatrix} x_G & y_G & z_G & \dot{x}_G & \dot{y}_G & \dot{z}_G \end{bmatrix}^{\mathrm{T}}, \tag{26}$$

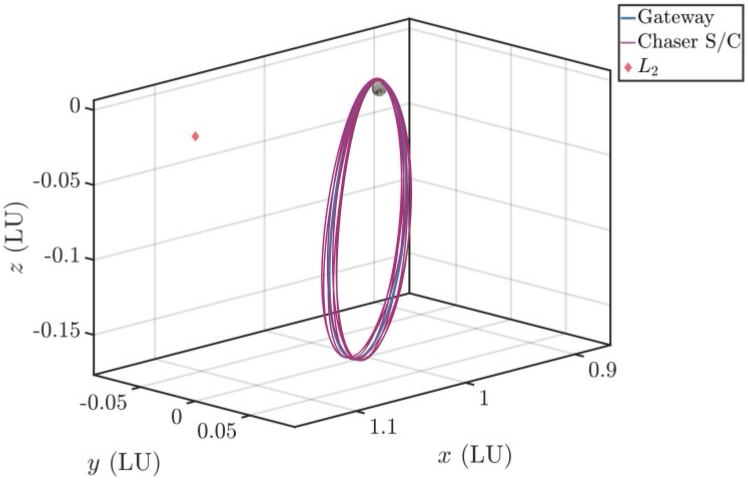


Figure 5: Chaser spacecraft's unknown QPO and Gateway's known $9:2$ NRHO used for the relative navigation example (adapted from Menges and Scheeres[26,27]).

where $\boldsymbol{r}_G$ and $\boldsymbol{v}_G$ are the position and velocity vectors, relative to the Earth-Moon barycenter. The state of Gateway is assumed to be known at all times onboard the chaser and is used within the measurement model for estimating its orbital trajectory.

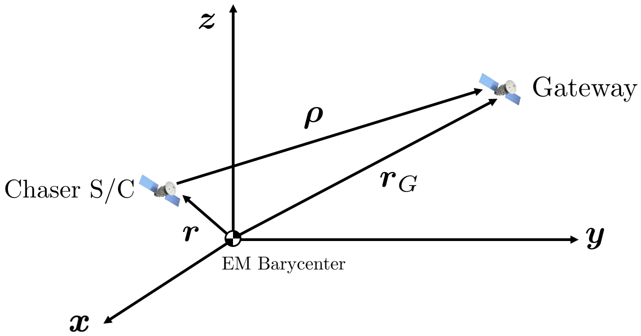


Figure 6: Geometric model for a relative navigation example involving NASA's Gateway on a $9:2$ NRHO and a chaser spacecraft on a nearby orbit (adapted from Menges and Scheeres[26]).

For this filtering scenario, the measurement model uses optical-only measurements consisting of azimuth and elevation to Gateway, θ_1 and θ_2, measured from the chaser. As mentioned previously, this model assumes that Gateway's state is known onboard the chaser. The geometry associated with these optical measurements is illustrated in Fig. 7, which shows the two angles relative to a rotating coordinate frame centered on the chaser. The equations defining the optical angle measurements are given by[28]

$$\theta_1 = \tan^{-1}\left(\rho_y\left(\rho_x\right)^{-1}\right), \tag{27}$$

$$\theta_2 = \tan^{-1}\left(\rho_z\left(\sqrt{\rho_x^2+\rho_y^2}\right)^{-1}\right), \tag{28}$$

where ρ_x, ρ_y, and ρ_z are the components of the relative position vector $\boldsymbol{\rho}$, defined in Eq. (24), in the rotating coordinate frame, such that

$$\rho = \sqrt{\boldsymbol{\rho} \cdot \boldsymbol{\rho}} = \sqrt{\rho_x^2 + \rho_y^2 + \rho_z^2}. \tag{29}$$

Note that, since attitude dynamics are not taken into account in this example, the rotating coordinate frame is aligned with the synodic frame but is centered at the chaser. Combining these expressions for the angles-only measurement model, the relative measurement model used here is defined as

$$\boldsymbol{h}_k\left(\boldsymbol{x}, \boldsymbol{x}_G\right) = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^{\mathrm{T}}. \tag{30}$$

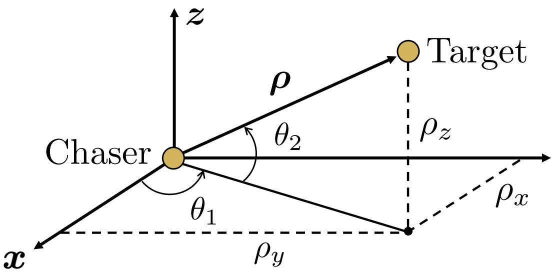


Figure 7: Relative navigation measurement model between NASA's Gateway on a $9\!:\!2$ NRHO and a chaser spacecraft on a nearby orbit (adapted from Menges and Scheeres[26]).

For this example, three different filters are implemented and compared as follows:

1. **UKF (DP54):** This filter uses the UKF formulation and uses the DP54 propagator presented in the *Integrator Performance* section to propagate the sigma points.
2. **EnGMF (DP54):** This filter follows the formulation presented in the previous section and uses the DP54 propagator presented in the *Integrator Performance* section to propagate the particles.
3. **EnGMF (`heyoka`):** This filter follows the formulation presented in the previous section and uses the `heyoka` propagator presented in the *Integrator Performance* section to propagate the particles.

The use of these three filters is made to compare the performance between linear and nonlinear filters, and to evaluate the runtime efficiency between using numerical propagators vs. Taylor integrators. For the UKF, $\alpha = 1$, $\beta = 2$, and $\kappa = 3 - n$, following the implementation described in Särkkä.[29] For both versions of the EnGMF, 500 particles are used. The true chaser trajectory and Gateway's location are obtained by using `MATLAB`'s `ode78`.

For each filter, 100 Monte Carlo (MC) runs are performed starting from an initial condition

$$\boldsymbol{x}_0 \sim \mathcal{N}\left(\begin{bmatrix}396717.65\ \mathrm{km} & 0\ \mathrm{km} & -67769.35\ \mathrm{km} & 0\ \frac{\mathrm{m}}{\mathrm{s}} & -97.58\ \frac{\mathrm{m}}{\mathrm{s}} & 0\ \frac{\mathrm{m}}{\mathrm{s}}\end{bmatrix}^{\mathrm{T}}, \boldsymbol{P}_0\right), \tag{31}$$

representing the approximate apolune of the QPO, with

$$\boldsymbol{P}_0 = \begin{bmatrix} \left(25\ \mathrm{km}\right)^2 \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \left(5\ \frac{\mathrm{m}}{\mathrm{s}}\right)^2 \boldsymbol{I}_{3\times3} \end{bmatrix}. \tag{32}$$

The location of Gateway is initialized at the apolune of the $9:2$ NRHO. The true chaser dynamics are propagated with additive zero-mean Gaussian process noise as described in Eq. (12), with covariance

$$\boldsymbol{Q}_k = \sigma_Q^2 \begin{bmatrix} \frac{\Delta t_k^2}{2}\boldsymbol{I}_{3\times3} & \Delta t_k\,\boldsymbol{I}_{3\times3} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{\Delta t_k^2}{2}\boldsymbol{I}_{3\times3} & \Delta t_k\,\boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{33}$$

with $\sigma_Q = 1 \times 10^{-12}\ \frac{\mathrm{km}}{\mathrm{s}^2}$, and $\Delta t_k = 0.5$ days. The angular measurements (obtained every Δt_k) are also corrupted with zero-mean Gaussian process noise as described in Eq. (13), with covariance[26]

$$\boldsymbol{R}_k = \sigma_R^2 \boldsymbol{I}_{2\times2}, \tag{34}$$

with $\sigma_R = 1 \times 10^{-5}$ rad.

To compare the performance of the three different filters, the standard deviation profiles are used.[30] The standard deviation profiles for position and velocity are defined as,

$$\sigma_{\boldsymbol{r}}(k) = \frac{1}{N_m}\sum_{j=1}^{N_m}\sqrt{\boldsymbol{P}_{k,j}^{+(1,1)} + \boldsymbol{P}_{k,j}^{+(2,2)} + \boldsymbol{P}_{k,j}^{+(2,2)}}, \tag{35}$$

$$\sigma_{\boldsymbol{v}}(k) = \frac{1}{N_m}\sum_{j=1}^{N_m}\sqrt{\boldsymbol{P}_{k,j}^{+(4,4)} + \boldsymbol{P}_{k,j}^{+(5,5)} + \boldsymbol{P}_{k,j}^{+(6,6)}}, \tag{36}$$

where N_m is the total number of MC runs. The standard deviation profiles obtained with the filter are compared with the sample standard deviation of the MC runs. The left panel in Fig. 8 shows the trajectory of Gateway (following the NRHO) and the chaser spacecraft in the QPO. The right panels in Fig. 8 shows the standard deviation profiles for each filter in solid lines. Their respective sample standard deviations are shown in dashed lines. For a filter to be consistent, its solid and dashed lines should match. If the filter is conservative, which can be desirable for aerospace applications, the solid line lies above the dashed line. If the filter is overconfident, suggesting potential divergence in the estimation error, the dashed line lies above the solid line.

From the standard deviation profiles, various conclusions can be drawn. First, it is clear that the UKF starts to diverge after the first perilune pass. The UKF completely diverges after the second perilune pass. In contrast, the EnGMF is able to accurately estimate the state throughout the entire trajectory. Despite perilune spikes, the EnGMF quickly recovers after gaining angular information. In addition, it can be seen that both the EnGMF (DP54) and the EnGMF (`heyoka`) provide the same exact performance. This figure shows the advantage of using nonlinear filters for highly nonlinear scenarios, and the equivalence of using numerical or Taylor integrators.

Figure 9 shows, on the left axis, the average position (Fig. 9a) and velocity (Fig. 9b) RMSE as a function of measurement frequency for the EnGMF (DP54) and the EnGMF (`heyoka`). The right axis of Fig. 9 shows the average runtime as a function of measurement frequency. Two main trends can be observed in this figure. First, as expected, the average estimation error grows with the sparsity of the measurements. As measurements become more sparse, the accuracy of the filter degrades equivalently for the EnGMF (DP54) and the EnGMF (`heyoka`). Second, the advantages of using the Taylor integrator can be noticed. As the propagation interval grows, the runtime for the EnGMF (DP54) grows linearly. In contrast, the runtime obtained with the EnGMF (`heyoka`) stays approximately constant. This shows that the main slowdown or computational bottleneck in the EnGMF is the propagation step, which can be alleviated by using Taylor integrators without losing accuracy.

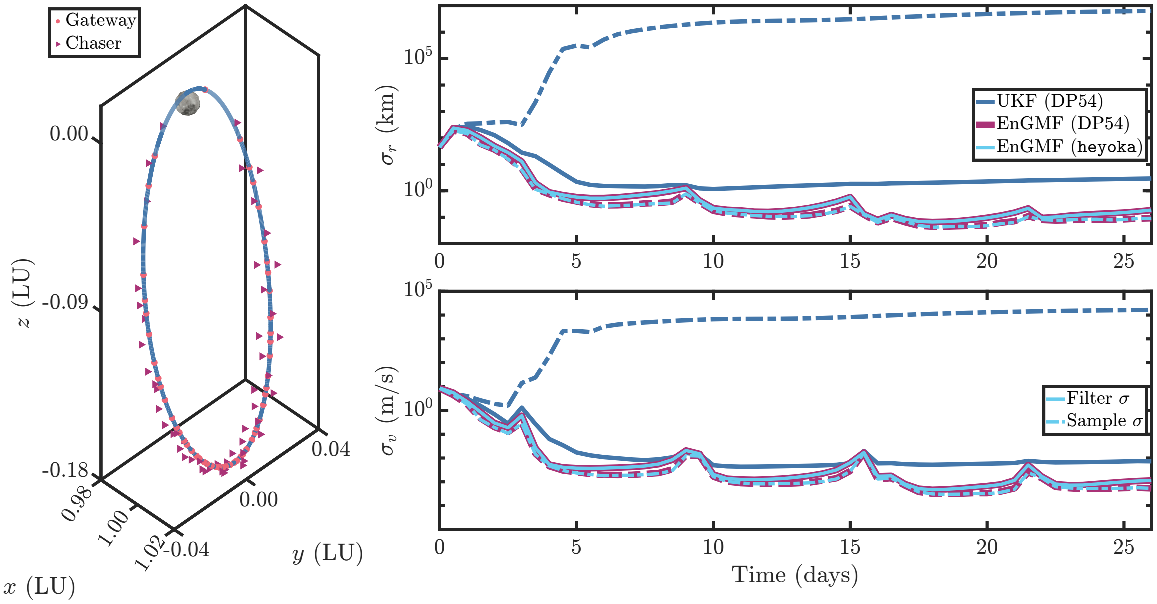
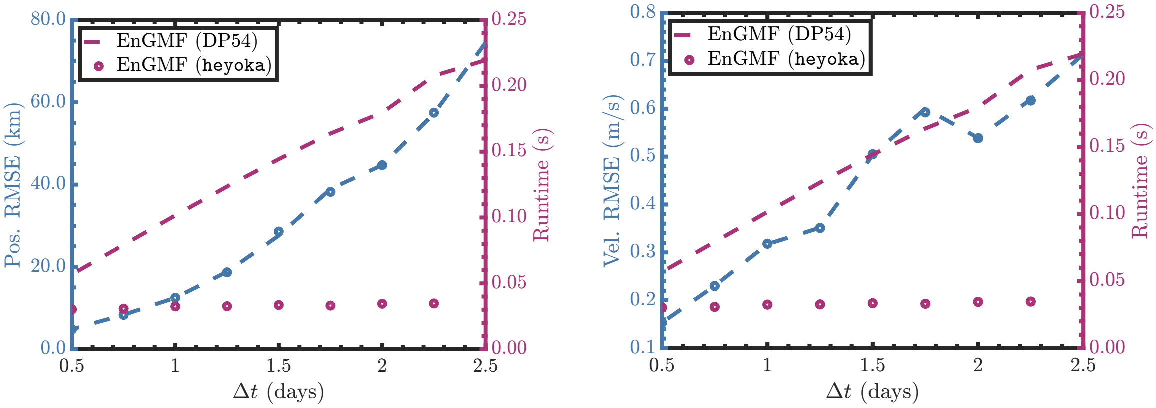


Figure 8: Trajectory of the chaser relative to Gateway in the NRHO, with each dot representing a measurement (left). Standard deviation profiles for the UKF (DP54), the EnGMF (DP54), and the EnGMF (`heyoka`) (right).



(a) Position RMSE

(b) Velocity RMSE

Figure 9: Position root mean square error (RMSE) and runtime as a function of Δt (left) for the two versions of the EnGMF. Velocity RMSE and runtime as a function of Δt (right) for the two versions of the EnGMF.

CONCLUSIONS

This work examined the implementation of the Taylor propagator, `heyoka`, with an EnGMF to support autonomous onboard OD under nonlinear dynamical conditions. Using the Earth-Moon system modeled by the CR3BP, the Taylor propagator was compared to existing numerical integrators, including `MATLAB`'s `ode45`, `ode78`, `ode89`, and `ode113`, and a compiled DP54 implementation. The results indicate that `heyoka` achieves low truncation errors and lower runtimes across a range of tolerances, making it suitable for high-accuracy propagations.

The Taylor propagator was then applied to the EnGMF to solve a relative navigation problem in cislunar space. In this scenario, a chaser spacecraft on a QPO estimates its state using relative angular measurements to NASA's Gateway on a $9:2$ resonant NRHO. The UKF diverged under these highly nonlinear dynamics, whereas both EnGMF implementations, using DP54 and `heyoka`, produced consistent state estimates. In addition, the EnGMF that uses the Taylor propagator maintained a nearly constant runtime for longer propagation intervals, reducing the overall computational costs associated with particle propagation.

Overall, combining Taylor integration and the EnGMF offers a practical solution for autonomous onboard OD in nonlinear scenarios such as the cislunar space. This integrated approach reduces the computational burden of propagating multiple particles and supports robust and accurate state estimation in challenging dynamical environments. Future work could explore extensions to different dynamical models and higher-fidelity onboard navigation systems to further enhance autonomous capabilities.

ACKNOWLEDGMENT

This material is based on research sponsored by the Air Force Office of Scientific Research (AFOSR) under award number: FA9550-23-1-0646.

REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc, 2001.

[2] D. Crouse, "Basic tracking using nonlinear continuous-time dynamic models [tutorial]," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 30, No. 2, 2015, pp. 4–41.

[3] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422.

[4] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, 2002, pp. 174–188.

[5] J. L. Anderson and S. L. Anderson, "A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts," *Monthly Weather Review*, Vol. 127, No. 12, 1999, pp. 2741–2758.

[6] S. Yun, R. Zanetti, and B. A. Jones, "Kernel-based ensemble gaussian mixture filtering for orbit determination with sparse data," *Advances in Space Research*, Vol. 69, No. 12, 2022, pp. 4179–4197.

[7] A. A. Popov and R. Zanetti, "Ensemble gaussian mixture filtering with particle-localized covariances," *Proceedings of the 2023 26th International Conference on Information Fusion (FUSION)*, Charleston, South Carolina, July 2023.

[8] R. Bucy and K. Senne., "Digital synthesis of non-linear filters," *Automatica*, Vol. 7, No. 3, 1971, pp. 287–298.

[9] M. Šimandl, J. Královec, and T. Söderström, "Advanced point-mass method for nonlinear state estimation," *Automatica*, Vol. 42, No. 7, 2006, pp. 1133–1145.

[10] F. Giraldo-Grueso, A. A. Popov, and R. Zanetti, "Gaussian Mixture-Based Point Mass Filtering," *Proceedings of the 2024 27th International Conference on Information Fusion (FUSION)*, Venice, Italy, July 2024.

[11] F. Giraldo-Grueso, A. A. Popov, and R. Zanetti, "Gaussian Mixture-Based Point Mass Filtering with Applications to Terrain Relative Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, 2025, pp. 1–15.

[12] J. R. Carpenter and C. N. D'Souza, *Navigation Filter Best Practices*. NASA, 2025.

[13] À. Jorba and M. Zou, "A Software Package for the Numerical Integration of ODEs by Means of High-Order Taylor Methods," *Experimental Mathematics*, Vol. 14, Jan. 2005, pp. 99–117, 10.1080/10586458.2005.10128904.

[14] F. Biscani and D. Izzo, "Revisiting High-Order Taylor Methods for Astrodynamics and Celestial Mechanics," *Monthly Notices of the Royal Astronomical Society*, Vol. 504, June 2021, pp. 2614–2628, 10.1093/mnras/stab1032.

[15] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross, *Dynamical Systems, the Three-Body Problem and Space Mission Design*. Marsden Books, 2022.

[16] M. Berz, *Modern Map Methods in Particle Beam Physics*. London, UK: Academic Press, 1999.

[17] J. A. Pérez-Hernández and L. Benet, "Non-Zero Yarkovsky Acceleration for near-Earth Asteroid (99942) Apophis," *Communications Earth & Environment*, Vol. 3, Jan. 2022, pp. 1–5, 10.1038/s43247-021-00337-x.

[18] H. Sorenson and D. Alspach, "Recursive bayesian estimation using gaussian sums," *Automatica*, Vol. 7, No. 4, 1971, pp. 465–479.

[19] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control*, Vol. 17, No. 4, 1972, pp. 439–448.

[20] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[21] F. Giraldo-Grueso, A. A. Popov, U. D. Hanebeck, and R. Zanetti, "Optimal Grid Point Sampling for Point Mass Filtering," *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, Kaua'i, Hawaii, Jan. 2025.

[22] J. R. Dormand and P. J. Prince, "A Family of Embedded Runge-Kutta Formulae," *Journal of Computational and Applied Mathematics*, Vol. 6, Mar. 1980, pp. 19–26, 10.1016/0771-050X(80)90013-3.

[23] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE Suite," *SIAM Journal on Scientific Computing*, Vol. 18, Jan. 1997, p. 22, 10.1137/S1064827594276424.

[24] J. H. Verner, "Numerically Optimal Runge–Kutta Pairs with Interpolants," *Numerical Algorithms*, Vol. 53, Mar. 2010, pp. 383–396, 10.1007/s11075-009-9290-3.

[25] L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*. San Francisco, CA: W. H. Freeman, 1975.

[26] R. J. Menges and D. J. Scheeres, "Orbit Determination Utilizing Taylor Methods: Applications to Relative Motion About Gateway," *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, Kaua'i, Hawaii, Jan. 2025.

[27] R. J. Menges and D. J. Scheeres, "Semi-Analytical F and G Series Solutions for the Circular Restricted Three-Body Problem," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Broomfield, Colorado, Aug. 2024.

[28] J. Greaves and D. Scheeres, "Spacecraft to Spacecraft Absolute Tracking for Autonomous Navigation of a Distributed Space System from Relative Sensors," *The Journal of the Astronautical Sciences*, Vol. 71, No. 5, 2024, p. 46.

[29] S. Särkkä, *Bayesian Filtering and Smoothing*. No. 3 in Institute of Mathematical Statistics Textbooks, Cambridge, UK: Cambridge University Press, first edition ed., 2013.

[30] S. Servadio and R. Zanetti, "Differential Algebra-Based Multiple Gaussian Particle Filter for Orbit Determination," *Journal of Optim. Theory Appl.*, Vol. 101, 2021, pp. 459–485.