Adaptive Relative Spacecraft Navigation in Proximity Operations with Unmodeled Target Behavior

Rachel Mamich[1*], Kristen Michaelson[1] and Renato Zanetti[1]

[1]Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, 2617 Wichita St North Office Building A, Austin, 78712, TX, USA.

*Corresponding author(s). E-mail(s): rachel.mamich@gmail.com; Contributing authors: kmichaelson@utexas.edu; renato@utexas.edu;

Keywords: Unmodeled Translational Maneuvers, Maneuver Detection, Adaptive Filtering, Crewed Spacecraft

An online method is presented for detecting and estimating unmodeled translational maneuvers of a target vehicle. Designed for relative navigation scenarios, the algorithm runs onboard the chaser vehicle and operates without communication from the target. By leveraging a buffer of observed pre-fit measurement innovations, the method eliminates the need for prior knowledge about the timing, magnitude, or direction of the target's maneuvers. This method integrates adaptive process noise estimation with a splitting algorithm designed specifically for this type of unmodeled dynamics. The algorithm demonstrates rapid detection and estimation of unmodeled target maneuvers in a low Earth orbit scenario, without relying on inflated uncertainty bounds. Its performance surpasses that of state-of-the-art adaptive filters designed with prior knowledge of the maneuver. The algorithm also demonstrates competitive performance in detecting and estimating target maneuvers in a cislunar scenario, matching or exceeding state-of-the-art filters configured with prior knowledge of the maneuver. In both scenarios, the adaptive estimation of process noise is key to enable both rapid transient response to detected maneuvers and accurate steady-state performance; all without any prior knowledge of the target's maneuver timing, magnitude and direction.

1 Introduction

The ability of a chaser vehicle to detect and estimate translational maneuvers (hereafter referred to as “burns”) performed by a target vehicle during spacecraft rendezvous and proximity operations is a rich and actively evolving area of research. The Kalman Filter (KF) [18, 17] and its extensions to nonlinear systems will perform poorly without prior knowledge of burn timing and magnitude . Thus, if a spacecraft being tracked performs a burn without the observer’s knowledge, it can lead to state divergence or loss of tracking by the observing sensors.

The most widely used modification of the KF for nonlinear systems is the Extended Kalman Filter (EKF). The EKF handles nonlinearities via first-order Taylor series approximations, centered on the filter estimated state [37]. Thus, if the filter estimated state strays too far from the true state, the EKF linearization approximation might be insufficient resulting in filter divergence. Filter divergence can also occur when there is a perturbation to the system that is unexpected or unmodeled. The rate at which the filter estimate diverges from the truth is greatly impacted by the nonlinearity of the system as well as the magnitude of the perturbation. When a perturbation or uncertainty is small enough, it is common to compensate for the lack of knowledge with process noise. However, larger perturbations with nonzero means, are not as easily remedied.

Unknown or unmodeled vehicle burns are an important class of perturbation in the aerospace industry. There has been research conducted into unmodeled target vehicle burns in ground based tracking applications as well as onboard relative navigation applications. These two applications have similar goals, but the computational power and availability of measurements have a large impact on the methods used to solve the problems. The work presented in this paper is focused on onboard rendezvous and docking applications. Spacecraft rendezvous and docking represent critical and complex flight phases, where failures can lead to extremely costly consequences.

There are scenarios in which a chaser vehicle may need to rendezvous with an uncooperative target. For example, communication may be lost with a malfunctioning spacecraft that must be intercepted for repair. In such cases, the ability to detect and estimate unexpected target burns becomes critical to avoid mission failure or collision, due to the increased accuracy of estimate provided to the collision avoidance system onboard.

In this work, an onboard relative navigation algorithm is proposed to detect and estimate an unmodeled target vehicle burn during proximity operations. To test the efficacy of the algorithm in the absence of a priori information, no explicit assumptions are made on the timing, objective, magnitude, or direction of the unmodeled target burn. The proposed algorithm is referred to as Burns Adaptively Learned On Orbit (BALOO). BALOO relies on a buffer of measurement innovations to simultaneously estimate the process noise in the system and detecting when a large unmodeled perturbation has taken place. When a large unmodeled target perturbation has been detected, a new hypothesis is created with an estimated unmodeled target acceleration. Low probability hypothesis are then discarded. The process noise adaptation portion of this work is based on the algorithm introduced in [24]. The contributions of this work are:

- A new rationale for determining if a new model is needed to represent the state based on the expected impacts of an unmodeled target vehicle translational maneuver
- A new method for defining the new model introduced to the system based again on the expected impacts of an unmodeled target vehicle translational maneuver
- An adaptive algorithm that allows for large dynamics model discrepancies, with no assumptions on the timing, magnitude, or direction of the perturbation
- A combination of small and large perturbation handling to reduce the number of additional models required to approximate the true dynamics of the system

The remainder of this paper is organized as follows. Section **??** reviews the state of the art and existing algorithms to address this problem. Then, Section **??** details how the BALOO algorithm functions. Next, Section **??** presents numerical examples in which the BALOO algorithm is compared against state of the art IMM algorithms that are tuned with prior information on the unmodeled target burns. Finally, Section **??** summarized the conclusions of this work.

2 Background and Existing Methodologies

Navigation algorithms typically rely on physical models of dynamics, sensor measurements, and on knowledge of both the initial state and noise covariance matrices. Knowledge of the physical models and covariance matrices are leveraged to construct the estimator. Uncertainty in the models results in degraded estimation performance. Compensation for model uncertainties can take many forms. In this section, methods for compensating for large and small model uncertainties are discussed.

2.1 Small Model Uncertainty

One commonly used technique to accommodate for unmodeled dynamics is State Noise Compensation (SNC) [37]. In SNC, an upper bound on the uncertainty in the system is determined and mitigated with process noise. It is common to estimate this upper bound by simulating trajectories with high fidelity models and comparing the results to those produced by the lower fidelity filter model [45]. Although SNC is widely used, it is limited by the assumption that the errors are a white process, i.e., not temporally correlated. To address this deficiency, Dynamic Model Compensation (DMC) is introduced [37]. DMC compensates for unknowns in the dynamics model by using Exponentially Correlated Random Variables (ECRVs, also known as first order Gauss-Markov processes). Although DMC addresses some of the shortcomings of SNC, both methods require a substantial knowledge about the system in order to be useful.

Uncertainties in the dynamics model can also be incorporated by including consider parameters in the state space [37]. Consider parameters allow for the inclusion of additional uncertain elements in the state space but without estimating them. This method is especially useful if the consider terms are not fully observable. Stauch and Jah demonstrate the use of consider parameters in an Unscented Kalman Filter (UKF) applied to an orbit determination scenario where the ballistic coefficient is not perfectly known [40]. In McCabe and DeMars, the usage of consider parameters is expanded to square-root filters in the context of ballistic reentry [26]. Consider parameters help to apply uncertainties in the state dynamics in more complex ways than SNC of DMC, but are still burdened with the need for a priori knowledge of parameter uncertainty.

Leveraging a priori estimates of uncertainties in a system via SNC, DMC, and consider parameters is commonplace in the aerospace industry and often provide reasonable upper bounds on the uncertainties in the systems during nominal flight conditions. However, there have been instances where it became necessary to adjust the noise parameters during the course of the mission [35, 11]. Stacy and D'Amico use Adaptive State Noise Compensation (ASNC) and Adaptive Dynamic Model Compensation (ADMC), which blend Covariance Matching (CM) techniques [36, 31, 34, 29] with SNC and DMC, respectively, to adjust the noise levels in a system orbiting an asteroid with a poorly known gravitational field [39]. Giraldo-Grueso et al. leverages neural networks and maximum likelihood methods in a reentry application [10, 9]. Mamich and Zanetti [24] provide an adaptive solution using correlation methods to adapt to poorly characterized venting disturbances caused by the presence of a crew on a target vehicle.

These strategies perform well when dealing with small-scale perturbations or those resembling white noise. However, their application with larger or more structured disturbances, can lead to subpar performance or even filter divergence.

2.2 Large Model Uncertainty

Larger, more structured model discrepancies require approaches beyond those discussed above. In this work, such "large discrepancies" refer specifically to unmodeled changes in the target vehicle's velocity, i.e., burns.

It is common to simplify the solution space for maneuver detection problems by assuming that burns are optimal in some sense; most often fuel-optimal.

This assumption is prevalent in many burn detection algorithms as spacecraft are typically designed to perform fuel-optimal maneuvers. Holzinger et al. leverage this idea using the minimum-fuel control distance and optimal control theory for burn detection and characterization [12]. This technique is expanded by Jaunzemis to non-Gaussian edge cases by applying a Gaussian mixture model; while the previous work assumed Gaussian distributions [14]. Lubey and Scheeres derive an optimal-control-based estimator using a Kalman filter algorithm in order to estimate both burns and mismodeled dynamics [23]. These methods are deterministic and solve an optimization problem to find the burn that best describes the observations.

An alternative approach that neither relies on optimality nor follows a deterministic framework is the use of Multiple Model (MM) filters. As the name suggests, MM filters are multiple model adaptive estimation (MMAE) algorithms that employ a bank of filters, each representing a different hypothesis about the system's dynamics, measurement models, and potentially burn detection criteria.

The earliest MMAE approaches assumed the existence of a single "true" model contained in the filter bank[2]. To extend the theory to more complex systems with switching dynamics and measurement models, Interacting Multiple Model (IMM) filters were proposed [2, 33, 25]. IMMs are a class of MM filters designed to handle instances where there may be more than one "true" model over the course of time designed to keep the solution computationally tractable. To do so, IMMs combine the state and covariance estimates of each node/filter in the bank in a weighted sum and provide each node the combined estimate. While the individual filters in the bank can be of different types, Kalman Filters are typically employed[25].

Several implementations of IMM have been successfully used to track vehicle performing maneuvers, ranging from standard IMMs with a wide array of models to encompass likely burns, to IMMs with variable sets of modes/filters enabled only when a burn is detected [21, 22]. These implementations require some prior knowledge of the potential models of the system in order to include them in the bank. Adaptive IMMs with companion bias filters [30, 20], and adaptive IMMs that include process noise adaptation [8] have been implemented to reduce reliance on a prioir information.

Gaussian Mixture Filters (GMFs) are a class of algorithms with many similarities to IMM. However, the different node in GMFs are used to better represent the probability density function (PDF) of the state rather than uncertainty in the underlying models. Although there have been some studies that have represented model uncertainties as distributions [46, 13].

Unlike IMMs, GMFs commonly split filter nodes. The decisions to split and the splitting direction are typically dictated by detection of nonlinearity in the system or non-Gaussianity in the node, while the splitting components are chosen to preserve the PDF. One state of the art GMF splitting algorithm is the AGEIS algorithm introduced by DeMars et al. which determines when to split based on the entropy of the system [7]. Tuggle and Zanetti's introduce new splitting methods for nonlinear measurement updates and for orbit uncertainty propagation [41, 42]. Kulik and LeGrand propose splitting methods informed by nonlinearities in the system [19]. The idea to split components has been employed in a broad range of applications, but has not yet been explored in the context of relative navigation in the presence of unmodeled target burns. This work investigates that possibility and develops a specialized splitting approach to solve this specific challenge.

3 Burns Adaptively Learned On Orbit

In [24], with the introduction of the MILTON (Measurement Innovation Leveraging To Obtain Noise) algorithm, it is shown that correlation methods function well in a system with unknown target venting perturbations and small gravitational model differences. Correlation methods were originally developed to adaptively estimate unbiased noise in a system [28]. The expected magnitudes of the venting perturbations for the Artemis missions are detailed in [5, 4, 15, 16, 27]. These perturbations are non-zero, but extremely small. While correlation methods were developed for linear systems with unbiased noise, the results show that correlation methods can still function well on nonlinear systems with low magnitude biases. Unfortunately, when an unmodeled perturbation is large, as is the case when the target vehicle performs an unmodeled burn, the algorithm MILTON is not capable of providing a quick adjustment.

Therefore, an MM filter with MILTON nodes and a tailored splitting algorithm is proposed to improve how quickly an unmodeled target vehicle burn can be detected and estimated in a relative navigation scenario. The proposed algorithm, BALOO, operates on the chaser vehicle. It is assumed that there is no communication between the two vehicles or with the ground. When the filter is initialized, it begins with a single MILTON node. If a split is determined to be required, a new node is generated based on the existing node and a buffer containing a time series of measurement innovations. The new node will also function as a MILTON filter. After a split occurs, each node operates independently from the other nodes in the system other than the normalization of the node weights which occurs during the measurement update. The algorithm's components are detailed below.

3.1 MILTON

In this work, each MILTON node operates in the same manner as the single MILTON filter is described in [24]. For completeness, a brief description of the MILTON algorithm from [24] is included.

A relative inertial state space is used in this work,

$$\mathbf{x} = \left[(\mathbf{r}_{t/c}^{N})^T \ (\mathbf{v}_{t/c}^{N})^T \ (\mathbf{a}_{t}^{N})^T \ (\mathbf{r}_{c}^{N})^T \ (\mathbf{v}_{c}^{N})^T \ (\mathbf{b}_{c}^{B_c})^T \right]^T, \tag{1}$$

where position vectors are indicated as $\mathbf{r}$, velocity vectors are indicated as $\mathbf{v}$, acceleration vectors are indicated as $\mathbf{a}$, and the chaser vehicle accelerometer bias vector is indicated as $\mathbf{b}$. The inertial frame is represented with the superscript ^{N} and the target vehicle and chaser vehicle body frames are represented with the superscripts ^{B_t} and ^{B_c} respectively. A relative vector subtracting the target states from the chaser states is shown with the subscript $_{t/c}$, while a vector with a single letter subscript, like $_c$, is taken to be defined as a standalone quantity. The chaser accelerometer bias terms are included in order to prevent attributing a bias in the onboard accelerometer with an unmodeled target vehicle acceleration. MILTON, consists of three primary components: propagation, measurement update, and an adaptive block.

3.1.1 Propagation

Each spacecraft is handled independently in the propagation phase. The states are propagated using a fixed step Runge-Kutta (RK4) propagator and the EKF propagation equations,

$$\begin{aligned} \dot{\mathbf{x}}_k &= \mathbf{f}\left(\mathbf{x}_k^+, t\right), \quad & \mathbf{P}_{k+1}^- &= \mathbf{F}\mathbf{P}_k^+\mathbf{F}^T + \hat{\mathbf{Q}}, \\ \mathbf{F} &\approx \mathrm{e}^{(\mathbf{A}\Delta t)}, & \mathbf{A} &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}_k^+}, \end{aligned} \tag{2}$$

where the state dynamics function, $\mathbf{f}()$, is a function of the state vector, $\mathbf{x}$ of size n, and time, t. The covariance estimate, $\mathbf{P}$, is propagated linearly using an approximated discrete time state transition matrix, $\mathbf{F}$, and the discrete time process noise matrix,

$\hat{\mathbf{Q}}$. Prior quantities are designated as, $()^-$. Posterior quantities are designated as $()^+$. Discrete time steps are shown as $()_k$ and $()_{k+1}$.

The original MILTON algorithm included a single filter only. In the BALOO algorithm, the filter is initialized with a single node, however the filter will create new nodes if the splitting criterion is met. The splitting algorithm will be discussed in the following section. The weight of each node, w is held constant over the propagation period,

$$w_{k+1}^- = w_k^+. \tag{3}$$

Prior to propagation, the relative target vehicle states are converted to inertial states using,

$$\mathbf{r}_t^N = \mathbf{r}_c^N + \mathbf{r}_{t/c}^N \quad \mathbf{v}_t^N = \mathbf{v}_c^N + \mathbf{v}_{t/c}^N. \tag{4}$$

The independent states are propagated with the dynamics,

$$\mathbf{f}(\mathbf{x}, t) = \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}}_t^N \\ \dot{\mathbf{v}}_t^N \\ \dot{\mathbf{a}}_t^N \\ \dot{\mathbf{r}}_c^N \\ \dot{\mathbf{v}}_c^N \\ \dot{\mathbf{b}}_c^{B_c} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_t^N \\ \mathbf{g}_t^N + \mathbf{a}_t^N \\ \boldsymbol{\nu}_t \\ \mathbf{v}_c^N \\ \mathbf{g}_t^N + \mathbf{NB_c}\left(\boldsymbol{\alpha}_c^{\mathbf{B_c}} - \mathbf{b}_c^{\mathbf{B_c}}\right) + \boldsymbol{\nu}_c \Delta t \\ -\beta \mathbf{b}^{\mathbf{B_c}} + \boldsymbol{\nu}_{b_c} \end{bmatrix}, \tag{5}$$

gravitational forces are shown as $\mathbf{g}$. Chaser accelerometer measurements, $\boldsymbol{\alpha}_c$, are taken in the chaser body frame. If the magnitude of the measured chaser acceleration exceeds a gating threshold, the measurement is used in propagation and applied as a constant over a timestep. To use this measurement, the estimated bias is removed, and then converted to the inertial frame using the direction cosine matrix (DCM), $\mathbf{NB_c}$, from the chaser body frame to the inertial frame. If the magnitude of the measured chaser acceleration is at or below the gating threshold, the accelerometer measurement is taken to be a direct measurement of the bias of the accelerometer and is not used in propagation [44]. The chaser accelerometer bias is modeled as a Markov process with an inverse of the time constant β. Process noise is represented as $\boldsymbol{\nu}$.

3.1.2 Measurement Update

In the measurement update, the EKF measurement update equations are used,

$$\begin{aligned} &\mathbf{x}_{k+1}^+ = \mathbf{x}_{k+1}^- + \mathbf{K}\left(\mathbf{r}_{y,k+1}^-\right), \quad \mathbf{r}_{y,k+1}^- = \mathbf{y}_{k+1} - \mathbf{h}\left(\mathbf{x}_{k+1}^-\right), \\ &\mathbf{K} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T \mathbf{W}_{k+1}^{-1}, \quad \mathbf{W}_{k+1} = \left(\mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R}\right), \\ &\mathbf{P}_{k+1}^+ = \left(\mathbf{I} - \mathbf{K}\mathbf{H}_{k+1}\right) \mathbf{P}_{k+1}^- \left(\mathbf{I} - \mathbf{K}\mathbf{H}_{k+1}\right)^T + \mathbf{K}\mathbf{R}\mathbf{K}^T, \quad \mathbf{H}_{k+1} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k+1}^-}. \end{aligned} \tag{6}$$

where the measurement function, $\mathbf{h}()$, is a function of the state. The measurement Jacobian, $\mathbf{H}$, is used to calculate the Kalman gain, $\mathbf{K}$, and the posterior covariance matrix. The measurement noise covariance matrix is $\mathbf{R}$. The quantity, $\mathbf{y}_{k+1}$, is the received measurement. The measurement is a vector of size m and is used to calculate the prefit measurement innovations, $\mathbf{r}^{-}_{y,k+1}$.

The weights in the BALOO algorithm are updated using the measurement likelihood and the previous weight. The weights are normalized to sum to one after the measurement update has been conducted on all of the existing nodes,

$$w^{+}_{k+1} \propto w^{-}_{k+1}\mathcal{N}\left(\mathbf{y}_{k+1}; \mathbf{h}\left(\mathbf{x}^{-}_{k+1}\right), \mathbf{W}_{k+1}\right). \tag{7}$$

The chaser vehicle is modeled as having an onboard flash LiDAR that provides relative range, ρ , and bearing, A, B, measurements in the LiDAR frame, $\mathbf{L}$,

$$\begin{aligned} &\boldsymbol{\rho}^{L} = \mathbf{LN}\mathbf{r}_{t/c}, \quad \rho = \left|\boldsymbol{\rho}^{\mathbf{L}}\right| + \eta_{\rho}, \\ &A = \arctan\left(\frac{\rho^{\mathbf{L}}_{x}}{\rho^{\mathbf{L}}_{z}}\right) + \eta_{A}, \quad B = \arctan\left(\frac{\rho^{\mathbf{L}}_{y}}{\rho^{\mathbf{L}}_{z}}\right) + \eta_{B}. \end{aligned} \tag{8}$$

where η_i is the measurement noise sample for the given measurement, i, $\mathbf{LN}$ is the DCM from the inertial frame to the chaser LiDAR frame and $\rho^{\mathbf{L}}_{x}$, $\rho^{\mathbf{L}}_{y}$, and$\rho^{\mathbf{L}}_{z}$ are the components of the relative range vector in the LiDAR frame.

In the event an accelerometer is below the set gating threshold, the measurement is applied as a direct measurement of the accelerometer bias [44],

$$\mathbf{b}^{\mathbf{B_c}}_{c} = \boldsymbol{\alpha}^{\mathbf{B_c}}_{c}, \tag{9}$$

The chaser vehicle position and velocity are treated as consider parameters and are not updated in the measurement update phase [37].

3.1.3 Process Noise Adaptation

Correlation methods are used to adaptively estimate the process noise used in each MILTON node. Correlation methods in nonlinear systems rely on a linear relationship in time of the prefit measurement innovations. Buffers are maintained for these prefit measurement innovations as well buffers of state transition matrices, measurement Jacobian matrices, and in the BALOO algorithm, a buffer of measurements themselves. For a single buffer of length m_b, this relationship is,

$$
\begin{aligned}
\underbrace{\begin{bmatrix} \mathbf{r}^-_{y,k} \\ \mathbf{r}^-_{y,k-1} \\ \vdots \\ \mathbf{r}^-_{y,k-m_b+1} \end{bmatrix}}_{=\mathcal{R}_{k,k-m_b+1}} &= \underbrace{\begin{bmatrix} \boldsymbol{\eta}_k \\ \boldsymbol{\eta}_{k-1} \\ \vdots \\ \boldsymbol{\eta}_{k-m_b+1} \end{bmatrix}}_{=\mathcal{W}_{k,k-m_b+1}} + \underbrace{\begin{bmatrix} \mathbf{H}_k \boldsymbol{\Phi}_{k,k-m_b+1} \\ \mathbf{H}_{k-1} \boldsymbol{\Phi}_{k-1,k-m_b+1} \\ \vdots \\ \mathbf{H}_{k-m_b+1} \end{bmatrix}}_{=\mathcal{O}_{k,k-m_b+1}} \mathbf{e}_{k-m_b+1} \\
&+ \underbrace{\begin{bmatrix} \mathbf{H}_k & \mathbf{H}_k\mathbf{F}_{k-1} & \mathbf{H}_k\boldsymbol{\Phi}_{k,k-2} & \cdots & \mathbf{H}_k\boldsymbol{\Phi}_{k,k-m_b+2} \\ \mathbf{0}_{m\times n} & \mathbf{H}_{k-1} & \mathbf{H}_{k-1}\mathbf{F}_{k-2} & \cdots & \mathbf{H}_{k-1}\boldsymbol{Phi}_{k-1,k-m_b+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times n} & \mathbf{H}_{k-m_b+2} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times n} & \mathbf{0}_{m\times n} \end{bmatrix}}_{=\mathcal{M}^w_{k-1,k-m_b+1}} \underbrace{\begin{bmatrix} \boldsymbol{\nu}_{k-1} \\ \boldsymbol{\nu}_{k-2} \\ \vdots \\ \boldsymbol{\nu}_{k-m_b+1} \end{bmatrix}}_{=\mathcal{V}_{k-1,k-m_b+1}},
\end{aligned} \tag{10}
$$

where $\mathcal{R}$, $\mathcal{V}$, and $\mathcal{W}$ are vectorized time series of the prefit measurement innovations, process noise samples, and measurement noise samples respectively. $\mathcal{O}$ is the observability matrix, and is full rank in this work. $\mathcal{M}^w$ is the mapping matrix from the process noise samples to the measurement innovations. It should be noted in the notation of this paper, $\mathbf{F}$ and $\boldsymbol{\Phi}$ are both state transition matrices. The form $\mathbf{F}_k$ is used to designate a single time step forward, meaning from time step k to $k+1$, while $\boldsymbol{\Phi}_{k,k-m_b+1}$ is used to signify the state transition matrix from $k-m_b+1$ to k. These quantities are used to calculate $\mathcal{A}$ and $\mathcal{B}$ matrices,

$$
\begin{aligned}
\mathcal{A}_k &= \left[\mathcal{M}^{-1}_{k,k-m_b+1}\mathcal{O}^T_{k,k-m_b+1}, \mathbf{0}_{n\times m}\right] - \left[\mathbf{0}_{n\times m}, \mathbf{F}_{k-m_b}\mathcal{M}^{-1}_{k-1,k-m_b}\mathcal{O}^T_{k-1,k-m_b}\right], \\
\mathcal{B}_k &= \left[\mathcal{M}^{-1}_{k,k-m_b+1}\mathcal{O}^T_{k,k-m_b+1}\mathcal{M}^w_{k-1,k-m_b+1}, \mathbf{I}_{n\times n}\right] - \\
&\qquad \left[\mathbf{0}_{n\times n}, \mathbf{F}_{k-m_b}\mathcal{M}^{-1}_{k-1,k-m_b}\mathcal{O}^T_{k-1,k-m_b}\mathcal{M}^w_{k-2,k-m_b}\right].
\end{aligned} \tag{11}
$$

The block rows of these matrices, $\mathbf{A}_i^k$ and $\mathbf{B}_i^k$ are used to calculate estimates of the autocorrelation, $\hat{\mathbf{C}}_{k,k-p}$,

$$
\begin{aligned}
&\underbrace{\begin{bmatrix} \operatorname{vech}\left(\hat{\mathbf{C}}_{k,k}\right) \\ \operatorname{vec}\left(\hat{\mathbf{C}}_{k,k-1}\right) \\ \vdots \\ \operatorname{vec}\left(\hat{\mathbf{C}}_{k,k-m_b}\right) \end{bmatrix} - \begin{bmatrix} \sum_{i=0}^{m_b} \mathbf{A}_i^k \otimes_h \mathbf{A}_i^{k\,T} \\ \sum_{i=1}^{m_b} \mathbf{A}_{i-1}^{k-1} \otimes_u \mathbf{A}_i^{k\,T} \\ \vdots \\ \sum_{i=m_b}^{m_b} \mathbf{A}_{i-m_b}^{k-m_b} \otimes_u \mathbf{A}_i^{k\,T} \end{bmatrix} \operatorname{vech}(\mathbf{R})}_{=\hat{\mathbf{c}}_k} = \\
&\underbrace{\begin{bmatrix} \sum_{i=1}^{m_b} \mathbf{B}_i^k \otimes_h \mathbf{B}_i^{k\,T} \\ \sum_{i=2}^{m_b} \mathbf{B}_{i-1}^{k-1} \otimes_u \mathbf{B}_i^{k\,T} \\ \vdots \\ \sum_{i=m_b+1}^{m_b} \mathbf{B}_{i-m_b-1}^{k-m_b-1} \otimes_u \mathbf{B}_i^{k\,T} \end{bmatrix}}_{=\hat{\mathbf{D}}_k} \mathbb{D}\mathbb{Q}\,\mathbf{s}_k, \quad \mathbb{Q} = \frac{\partial \mathbf{Q}_{\text{diag}}}{\partial \mathbf{s}_k}, \\
&\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}, \quad \begin{aligned} \operatorname{vec}(\mathbf{Q}) &= \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{21} & q_{22} & q_{23} & q_{31} & q_{32} & q_{33} \end{bmatrix}^T, \\ \operatorname{vech}(\mathbf{Q}) &= \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{22} & q_{23} & q_{33} \end{bmatrix}^T, \end{aligned}
\end{aligned} \tag{12}
$$

where $\mathbf{s}_k$ is the square root of the continuous time process noise covariance matrix diagonal, and $\mathbb{Q}$ is the derivative of the continuous time process noise diagonal with respect to the square root of that diagonal. The matrix $\mathbb{D}$ transforms the continuous time process noise covariance matrix diagonal to the upper triangular of the discrete time process noise covariance matrix which is governed by (**??**),

$$
\hat{\mathbf{Q}} = \begin{bmatrix} \frac{\Delta t^5}{20}\mathbf{Q} & \frac{\Delta t^4}{8}\mathbf{Q} & \frac{\Delta t^3}{6}\mathbf{Q} \\ \frac{\Delta t^4}{8}\mathbf{Q} & \frac{\Delta t^3}{3}\mathbf{Q} & \frac{\Delta t^2}{2}\mathbf{Q} \\ \frac{\Delta t^3}{6}\mathbf{Q} & \frac{\Delta t^2}{2}\mathbf{Q} & \Delta t\mathbf{Q} \end{bmatrix}, \tag{13}
$$

where $\mathbf{Q}$ is the continuous time process noise. The symbols $\otimes_h$ and $\otimes_u$ represent special Kroneker products that apply when $\mathbf{X}$ is symmetric,

$$
\begin{aligned}
\operatorname{vec}(\mathbf{AXB}) &= \left(\mathbf{B}^T \otimes_u \mathbf{A}\right) \operatorname{vech}(\mathbf{X}), \\
\operatorname{vech}\left(\mathbf{AXA^T}\right) &= \left(\mathbf{A} \otimes_h \mathbf{A}\right) \operatorname{vech}(\mathbf{X}).
\end{aligned} \tag{14}
$$

From here, (**??**) is then used in an information least squares framework to estimate the process noise in the system using,

$$\begin{aligned}
\mathbf{\Lambda}_k^+ &= \left(\mathbf{\Psi}_k^+\right)^{-1}, \quad \mathbf{d}_k^+ = \mathbf{\Lambda}_k^+ \mathbf{s}_k^+, \\
\mathbf{\Lambda}_{k+1}^- &= \left(\mathbf{I}_{3\times3} - \mathbf{\Lambda}_k^+ \left(\mathbf{\Lambda}_k^+ + \mathbf{Q}_w^{-1}\right)^{-1}\right) \mathbf{\Lambda}_k^+, \\
\mathbf{d}_{k+1}^- &= \left(\mathbf{I}_{3\times3} - \mathbf{\Lambda}_k^+ \left(\mathbf{\Lambda}_k^+ + \mathbf{Q}_w^{-1}\right)^{-1}\right) \mathbf{d}_k^+, \\
\mathbf{d}_{k+1}^+ &= \mathbf{d}_{k+1}^- + \hat{\mathbf{D}}_{k+1}^T \mathbf{R}_w^{-1} \hat{\mathbf{c}}_{k+1}, \\
\mathbf{\Lambda}_{k+1}^+ &= \mathbf{\Lambda}_{k+1}^- + \hat{\mathbf{D}}_{k+1}^T \mathbf{R}_w^{-1} \hat{\mathbf{D}}_{k+1},
\end{aligned} \tag{15}$$

where $\mathbf{d}$ is the information state, $\mathbf{\Lambda}$ is the information matrix, and $\mathbf{Q}_w$ and $\mathbf{R}_w$ are the process noise and measurement noise for the adaptation itself.

3.2 Splitting Algorithm

As mentioned in Section **??**, splitting algorithms have two components, the decision to split and the direction in which to split. Many of the previous works on component splitting algorithms have been developed for GMFs to mitigate nonlinear dynamics or nonlinear measurement functions. The decision to split relies on some measure of nonlinearity or non-Gaussianity and the determination on how to split components attempts to preserve the PDF pre- and post-splitting. In this work, the reason for splitting is based on the possibility of unmodeled target vehicle burns. A mismatch in the filter dynamics model and the true dynamics can be detected by analyzing the prefit measurement innovations of the system. In a perfectly modeled system, the prefit measurement innovations will appear to be nonbiased, or zero mean.

A buffer with a time series of measurement innovations, $\mathcal{R}^j$, is maintained in each node, j, for use in the adaptive process noise block. This buffer is used to detect when a split should occur. The user defines the number of time steps to include in the split decision, $m_s \in [1, 2, ..., m_b]$. The split decision is made using a scaled, non-dimensional innovation squared (SNDIS),

$$\text{if} \quad \frac{1}{m_s} \sum_{i=0}^{m_s-1} \left(\mathbf{r}_{y,i}^{-,j}\right)^T \mathbf{R}_i^{-1} \mathbf{r}_{y,i}^{-,j} > \epsilon_s \quad \text{split the jth node} \tag{16}$$

where $\mathbf{r}_{y,i}^{-,j}$ is the ith measurement innovation prior to the current time step k of the jth node in the filter and ϵ_s is the user defined threshold for splitting. It should be noted here that this threshold is set heuristically in this work. While the chosen threshold impacts the results, the chosen value of the splitting threshold can be set roughly by considering the capabilities of the target vehicle. If the capabilities of the target vehicle are not well known, it is advisable to set the threshold 'low'. Setting the threshold 'low' might lead to more frequent unnecessary splits, but will ensure that unmodeled maneuvers are captured. The measurement noise covariance matrix is used to make **??** non-dimensional. The possibility of using the prefit innovation covariance matrix, $\mathbf{P}_{yy,i}^{-,j}$ in place of $\mathbf{R}_i$ was investigated and discarded, because $\mathbf{P}_{yy,i}^{-,j}$

has an indirect dependence on the process noise estimate $\hat{\mathbf{Q}}$ and can therefore lead to a delayed splitting response to a large perturbation.

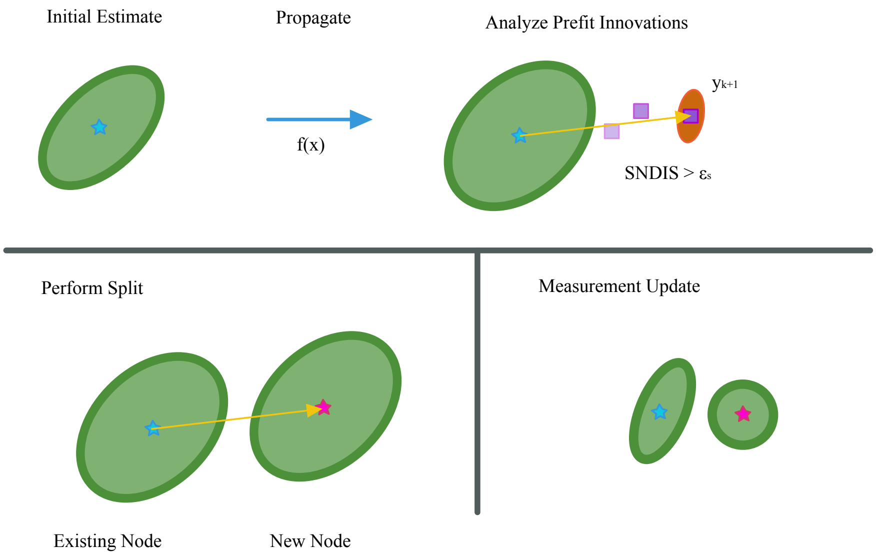

Fig. 1 A visual representation of how the splitting algorithm operates. Beginning with the initial state, the blue star, and covariance, the green ellipse in the top left. The state is then propagated in time to the next time step. The prefit measurement innovation, symbolized with the yellow arrow, of the current timestep is evaluated. Then it is added to the prefit measurement innovation buffer. The buffer is symbolized with the additional, more transparent purple squares. If the buffer of prefit measurement innovations result in a SNDIS value that exceeds the splitting criterion, (**??**), then a new node is generated. This new node is centered on the pink star and the new estimate is generated using (**??**) and (**??**). This is shown in the bottom left. The nodes are then processed in the measurement update independently. If the splitting criterion is not exceeded, then no split occurs.

Once a split has been determined to be necessary for the jth node, the node and its measurement innovation buffer are used to construct a new node. In this work, it is assumed that an unmodeled acceleration acting on the target vehicle is the cause of the bias in the SNDIS. Thus the direction of the split is determined based on the estimated unmodeled target acceleration over the portion of the buffer included in the split decision, m_s. Again, the relative target states in this work are fully observable with range and bearing measurements.

In addition to a buffer of measurement innovations, a buffer of measurements, $\mathcal{Y}$ is also maintained to estimated unmodeled target accelerations. Note that each filter receives the same measurements, hence a single buffer is maintain for all nodes. $\mathcal{Y}$ is used to construct relative range estimates, $\hat{\mathbf{r}}^{\mathbf{N}}_{i,t/c}$,

$$\begin{aligned}
\hat{r}^{\mathbf{L}}_{t/c,x,i} &= \hat{r}^{\mathbf{L}}_{t/c,z,i}\tan(A_i)\\
\hat{r}^{\mathbf{L}}_{t/c,y,i} &= \hat{r}^{\mathbf{L}}_{t/c,z,i}\tan(B_i)\\
\hat{r}^{\mathbf{L}}_{t/c,z,i} &= \sqrt{\frac{\rho_i^2\cos^2(A_i)}{1+\cos^2(A_i)\tan^2(B_i)}}\\
\hat{\mathbf{r}}^{\mathbf{N}}_{t/c,i} &= \mathbf{NL}_i\hat{\mathbf{r}}^{\mathbf{L}}_{t/c,i}
\end{aligned} \tag{17}$$

where i indicates the ith quantity prior to the current time step. These estimated relative range vectors based on the measurements and the buffer of state estimates are then used to calculate an estimated mismatch in the unmodeled target acceleration over the period, $\Delta\mathbf{a}_t^N$. In order to detect a change in the unmodeled target acceleration state, enough time must pass to see the impact in the measurement innovations. It is assumed that the unmodeled target burn begins half way back in the time contained in the buffers, $\frac{m_s}{2}$, so only the latter half of the buffer is used in in the unmodeled target acceleration calculation,

$$\Delta\mathbf{a}_t^{\mathbf{N},j} = \frac{1}{m_s}\sum_{i=0}^{\frac{m_s}{2}}\frac{1}{\Delta t_i^2}\left(\hat{\mathbf{r}}^{\mathbf{N},j}_{t/c,i} - \mathbf{r}^{\mathbf{N},j}_{t/c,i},\right). \tag{18}$$

While the design choice was made to assume that the unmodeled target burn begins half way back in the time contained in the buffers, this is not a strictly necessary assumption. Since the measurements are received at discrete time steps, this quantity, $\frac{m_s}{2}$ is rounded to the nearest timestep. With this assumption, the target vehicle relative position and velocity and unmodeled acceleration of the new node, $*$, are set as,

$$\begin{bmatrix}\mathbf{r}^{\mathbf{N},*}_{t/c}\\ \mathbf{v}^{\mathbf{N},*}_{t/c}\\ \mathbf{a}^{\mathbf{N},*}_{t}\end{bmatrix} = \begin{bmatrix}\mathbf{r}^{\mathbf{N},j}_{t/c}\\ \mathbf{v}^{\mathbf{N},j}_{t/c}\\ \mathbf{a}^{\mathbf{N},j}_{t}\end{bmatrix} + \begin{bmatrix}\frac{1}{2}\Delta t^2_{\frac{m_s}{2}}\Delta\mathbf{a}^{\mathbf{N},j}\\ \Delta t_{\frac{m_s}{2}}\Delta\mathbf{a}^{\mathbf{N},j}\\ \Delta\mathbf{a}^{\mathbf{N},j}\end{bmatrix}, \tag{19}$$

where $\Delta t_{\frac{m_s}{2}}$ is the change in time from the current time back to the half way point of the buffer. In addition to the adjustments to the target vehicle states, there are a few more quantities that must be addressed. The initial weight that the new node is allotted is $\frac{1}{2}$ the weight of the node that it splits from. The covariance of the new node is a copy of the covariance of the node that it splits from. The process noise estimate and its corresponding covariance matrix are reset to the same values used at initialization. The buffers of the new node are initialized to be empty and all other variables for the new node $*$ are set to match the node it splits from j. Finally, the node that was found to need a split is prevented to split again for a user-determined number of time steps. This is done to ensure that the algorithm does not perform more splits than necessary.

When there is a single node in the system, the filter returns the state estimate and covariance matrix estimate of its one node. When there is more than one node in the system, the filter returns a weighted average of the state and covariance matrix estimates,

$$\begin{aligned}\tilde{\mathbf{x}}_k &= \sum_{1}^{n_n} w^j \mathbf{x}_k^j \\ \tilde{\mathbf{P}}_k &= \sum_{1}^{n_n} w^j \left(\mathbf{P}_k^j + \left((\mathbf{x}_k^j - \tilde{\mathbf{x}}_k\right)\left((\mathbf{x}_k^j - \tilde{\mathbf{x}}_k\right)^T\right)\end{aligned} \tag{20}$$

4 Numerical Results

This section covers the results for two different scenarios in which there is a crewed target vehicle that experiences unmodeled venting perturbations as well as performing a burn without the chaser vehicle's knowledge. The venting perturbations are the result of the crew life support system. To maintain the health of the crew, CO_2 gas and waste water must be eliminated from the vehicle.

The vents in this work are modeled after information released from NASA on Gateway [27, 32, 6]. The vents are applied according to Table **??** and are modeled as discrete 'on' or 'off' in the true dynamics.

Table 1 Venting information, assumes the spacecraft mass is $39000kg$ [27]

Vent Type	Frequency	1σ Magnitude $[m/s^2]$	Duration $[s]$	Body Frame Direction
CO_2	every 10 min	1.7094e-5	10	$[-0.5, -0.866, 0]^T$
Waste Water	every 6 hours	1.8803e-05	60	$[-0.5, 0, -0.866]^T$

The two scenarios differ in orbital regime. In the first scenario, the vehicles are in Low Earth Orbit (LEO). In the second, the vehicles operate in cislunar space and ultimately end in a 9:2 Near Rectilinear Halo Orbit (NRHO) [38]. The Artemis missions themselves operate in both LEO and NRHO orbits; thus, having onboard algorithms that can operate in both LEO and NRHO orbits is of practical current interest.

In both scenarios, measurements are received every second from a flash LiDAR and an accelerometer on board the chaser vehicle. It is assumed that the target vehicle is approximately centered in the FOV of the LiDAR. The LiDAR is modeled to have a 128x128 pixel array with a 5°x5° field of view [1] with a range and azimuth/elevation uncertainties of 0.01m and 0.04°, respectively.

$$
\begin{aligned}
\mathbf{R}_1 &= \begin{bmatrix} (0.01m)^2 & 0 & 0 \\ 0 & (0.04°)^2 & 0 \\ 0 & 0 & (0.04°)^2 \end{bmatrix} \quad \text{or} \\
\mathbf{R}_2 &= \begin{bmatrix} \mathbf{R}_1 & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & (3.433\text{e-}6\mathbf{I}_{3\times3}m/s^2/\sqrt{s})^2 \end{bmatrix}.
\end{aligned} \tag{21}
$$

The gating threshold of the accelerometer is 1e-$4m/s^2$. Accelerometer measurements with a magnitude under this threshold are treated as direct measurements of the chaser accelerometer bias. The measurement noise matrix $\mathbf{R}_2$ is used in these instances. Accelerometer measurements that exceed the threshold are adjusted as discussed in Section **??** and used in the chaser vehicle propagation. The measurement noise matrix $\mathbf{R}_1$ is used in these instances.

Each scenario is tested by generating a unique set of truth information for a target and chaser vehicle. This truth is used to test the BALOO algorithm in addition to two IMM filters designed with prior information about the target vehicle's burn. A 100 run ($N_{\text{MC}} = 100$) Monte Carlo (MC) simulation is conducted for each scenario and filter combination, where each MC run has its own truth trajectory. The error estimates as well as the error interval estimates from each filter are used to assess how well each filter performs. The true errors are used to calculate the sample average error and corresponding sample covariance,

$$
\begin{aligned}
\bar{\mathbf{e}}_k^+ &= \frac{1}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} \mathbf{e}_k^{i+} = \frac{1}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} (\hat{\mathbf{x}}_k^{i+} - \mathbf{x}_k^i), \\
\mathbf{P}_k^+ &= \frac{1}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} \mathbf{e}_k^{i+} \mathbf{e}_k^{i+T},
\end{aligned} \tag{22}
$$

where the truth state, $\mathbf{x}_k^i$, is subtracted from the posterior estimated state, $\hat{\mathbf{x}}_k^{i+}$ of each MC run i, at time k.

The sample errors and covariance for each run are compared against the average estimated covariance over all MC runs,

$$
\bar{\mathbf{P}}_k^+ = \frac{1}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} \hat{\mathbf{P}}_k^{i+}, \tag{23}
$$

4.1 Low Earth Orbit

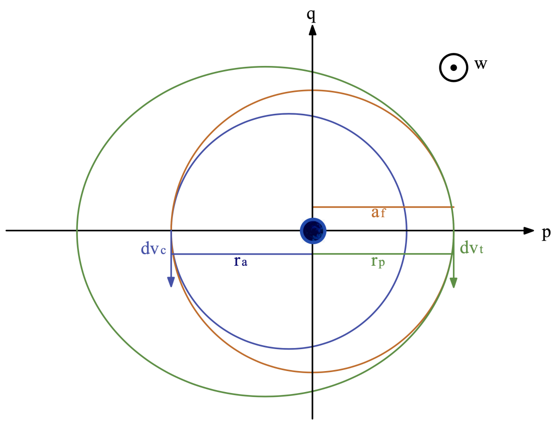

Fig. 2 Sketch of LEO mission scenario in the target vehicle's initial perifocal frame centered on the Earth; p is in the direction of periapsis, q is $\mathbf{90}°$ from p in the direction of orbital motion, and w (out of the page) is in the orbit normal direction [43]. The larger outer orbit (green) is the initial target orbit and $\mathbf{dv_t}$ (green vector) represents the target vehicle's burn at periapsis. The smaller orbit (blue) is the initial chaser orbit and $\mathbf{dv_c}$ (blue vector) represents the chaser vehicle's burn at apoapsis. The intermediate circular orbit (orange) is the final orbit that each vehicle maneuvers into. This sketch is not drawn to scale.

4.1.1 Scenario Setup

In the LEO scenario, the target and chaser vehicles begin in coplanar orbits. The radius at periapsis of the target vehicle's initial orbit is the same as the chaser vehicle's initial radius at apoapsis. The goal in this scenario is to have the two spacecraft rendezvous in a circular orbit with a semimajor axis equal to this shared value. A concept sketch of this setup is shown in Figure **??**. Both vehicles are subject to Earth's gravity. In this work the Earth is modeled as an oblate spheroid. The gravity model includes the gravity due to a point mass in the two body context as well as J_2–J_4 zonal harmonic terms,

$$\mathbf{g} = \mathbf{g}_{\mathrm{TBP}} + \sum_{i=2}^{4} \mathbf{g}_{J_i}. \tag{24}$$

This equation is the result of a derivation found in [43]. The relevant constant terms used in the Earth gravity model are located in Table **??**. In [24], it is shown that the venting perturbations overwhelm gravity modeling errors. Thus the gravity model used in the true dynamics is the same as the model used in the filter. The true dynamics on the target vehicle include regular venting perturbations as well as a finite burn. The filter dynamics model for the target vehicle does *not* include any a priori information on the venting perturbations. The chaser vehicle also performs a finite burn in order to rendezvous with the target vehicle. The burn information can be found in Table **??**.

Table 2 Earth constants used in propagation.

Parameter	μ $[m^3/s^2]$	$R_{Earth}[m]$	$J_2[-]$	$J_3[-]$	$J_4[-]$
Value	3.986e14	6.378e6	1.083e-3	-2.532e-6	-1.620e-6

Table 3 Burn information for the LEO scenario.

Vehicle	Burn Start $[s]$	Burn Length $[s]$	Magnitude $[m/s^2]$	Inertial Direction
Target	121	60	6.34e-2	$[0.0000, -0.8660, -0.5000]^T$
Chaser	2989	30	3.805e-4	$[0.0913, -0.8624, -0.4979]^T$

In each MC run, the vehicles end in the same final orbit, shown in Table **??**.

Table 4 Final Keplerian elements for the LEO scenario.

Element	$a[m]$	$e[-]$	$inc[°]$	$\Omega[°]$	$\omega[°]$	$M_f[°]$
Target	6.878e6	0.000	30.00	0.000	0.000	180.0
Chaser	6.878e6	0.000	30.00	0.000	0.000	180.0 + 5.73e-4

In order to ensure that the vehicles end in the same final orbit, each truth scenario is generated by back propagating from the final conditions to the beginning of the scenario. Each MC has its own set of randomly sampled target vehicle venting perturbations. The timing of the first CO_2 venting perturbation is randomly sampled from a uniform distribution within the 300s. From there each CO_2 perturbation occurs at the frequency in Table **??**. Due to the infrequency of the waste water vents, only one is modeled in the scenario. The waste water event occurs 1400s after the first CO_2 vent. In addition to the timing of the vents, the magnitude of each vent is also randomized by randomly sampling from a Gaussian distribution centered on zero with an uncertainty listed in Table **??**, then taking the absolute value of the sample. In the event a venting perturbation occurs during the target vehicle burn, the vent is skipped.

Measurements are received every second. The true chaser accelerometer bias is initialized to be 2.9430e-6 m/s^2 for each axis. The velocity random walk of the chaser accelerometer is set to be 3.433e-6 $m/s^2/\sqrt{s}$ [24]. The discrete time process noise is set to be 1.179e-9$\mathbf{I}$ m/s^2. The inverse of the time constant for the chaser accelerometer bias is set to be $\beta = 3600^{-1}s^{-1}$ for each axis.

4.1.2 BALOO Setup

To initialize each filter run, the initial state is sampled from a Gaussian distribution, centered on the true initial conditions with a covariance matrix,

$$\mathbf{P}_0 = \begin{bmatrix} \left(\frac{25}{3}\right)^2 \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \left(\frac{1}{10}\right)^2 \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & (1\text{e-}9)^2 \mathbf{I}_{3\times3} \end{bmatrix}. \tag{25}$$

This covariance matrix is selected to represent a recent loss of GPS coverage. The covariance matrix has units of m^2, $(m/s)^2$, and $(m/s^2)^2$. In addition to the initial state of the filter, the initial estimate of the filter process noise is also randomized. Each element of the diagonal of the initial process noise in the filter is sampled from a discrete uniform distribution from the set $\left[10^{-22}, 10^{-20}, ..., 10^{-10}, 10^{-8}\right]$. This random sample is added to an initial guess of $\mathbf{s}_0 = 10^{-12}\mathbf{1}_{3x1}$. The initial covariance matrix of the process noise estimate is set to be $\mathbf{\Psi}_0 = 1e - 3\mathbf{I}_{3x3}$. The process noise and measurement noise matrices of the filter process noise estimate are $\mathbf{Q}_w = 1e - 6\mathbf{I}$ and $\mathbf{R}_w = 1e - 5\mathbf{I}$. The size of the adaptation buffer is set to be $m_b = 15$. The size of the buffer used to determine a split and to estimate the error in the unmodeled uncertainty, m_s, is 10. The splitting threshold, ϵ_s, is set to be 71.5. This splitting threshold is set heuristically through testing the scenario. Nodes are considered statistically insignificant and pruned off it the node's weight falls below 1e–12. When a node is pruned, the node is simply deleted.

4.1.3 Seven Model IMM Setup

In the seven model IMM, each of the models is treated as an EKF with fixed process noise. Each EKF node runs independently over the propagation step in (**??**) and the measurement update step (**??**). After each measurement update step, the weights of the nodes are evaluated using the measurement likelihood. Then the interacting state is calculated by combining the estimates in a weighted sum. This interacting state is then given to each node as the new estimate prior to the next round of propagation. Each node applies its model through a consider unmodeled target acceleration state. Table **??** shows the acceleration vector of each node. In the interacting portion of the IMM, only the non-consider states and covariance estimates are combined. The probability that the model stays the same is 1/3 for each model. The transitional probabilities are all set to be 1/9.

Table 5 Acceleration models for each IMM node.

Node	1	2	3	4	5	6	7
$\mathbf{a}_{j,x}$	0.3	-0.3	0	0	0	0	0
$\mathbf{a}_{j,y}$	0	0	0.3	-0.3	0	0	0
$\mathbf{a}_{j,z}$	0	0	0	0	0.3	-0.3	0

It should be mentioned that trials were conducted with a smaller burn magnitudes, closer to that of the true unmodeled target acceleration. However, each of those trials resulted in at least one MC where the seven model IMM diverged.

The process noise is set to be zero for the target vehicle states. The continuous time process noise matrix for the chaser vehicle dynamics states is set to be $\mathbf{Q}_c = 1e - 12\mathbf{I}_{3x3}$, and the process noise on the chaser accelerometer bias is set to be 1.179e-9$\mathbf{I}m/s^2$.

4.1.4 Two Model IMM Setup

In the two model IMM, each of the models is treated as an EKF with fixed process noise. The two models are a high noise and low noise EKF. In each model, the unmodeled target acceleration is estimated. The inclusion of the unmodeled target acceleration states in the Two Model IMM functions the same as in the adaptive BALOO filter.

Table 6 Continuous time process noise matrices applied to the target vehicle states for each IMM node.

Node	1	2
$\mathbf{Q}_t$	$1e-2\mathbf{I}_{3x3}$	$1e-12\mathbf{I}_{3x3}$

It should be mentioned that trials were conducted with lower "high noise" process noise. However, each of those trials resulted in at least one MC where the two model IMM diverged.

In the interacting portion of the IMM, only the non-consider states and covariance estimates are combined. The probability that the model stays the same is 2/3 for each model. The transitional probability is set to be 1/3. The continuous time process noise matrix for the chaser vehicle dynamics states is set to be $\mathbf{Q}_c$ = 1e-12$\mathbf{I}_{3x3}$, and the process noise on the chaser accelerometer bias is set to be 1.179e-9$\mathbf{I}m/s^2$.

4.1.5 Results

The analysis of the results begins by examining the time surrounding the unmodeled target burn. The burn begins at 121s and concludes at 181s.

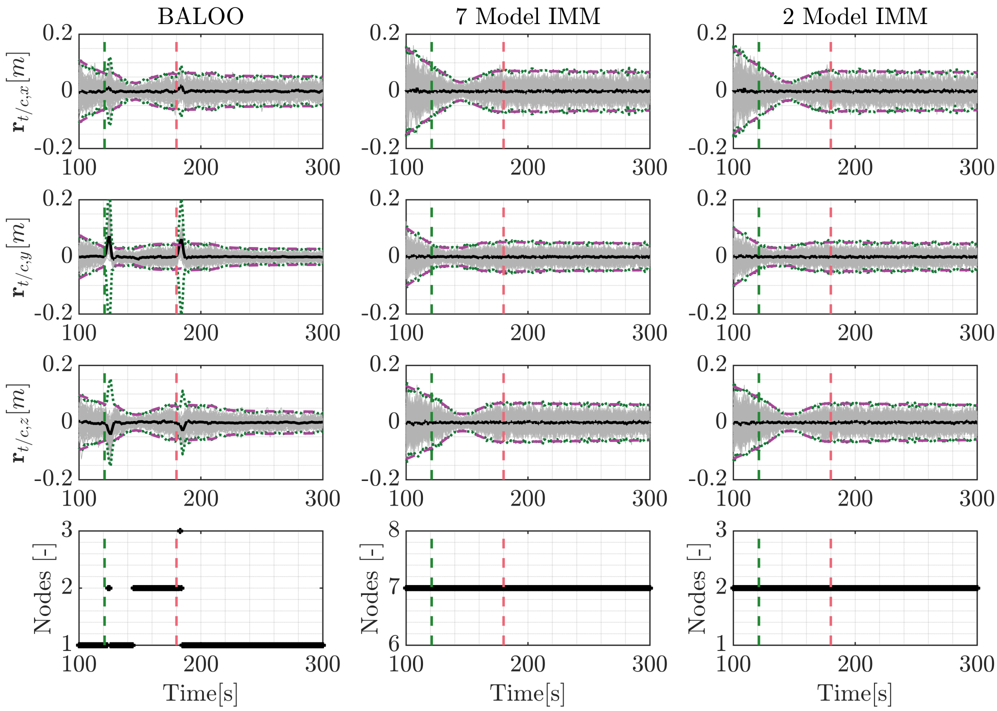

Fig. 3 Error in the relative target position state versus time for each filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs. The vertical dashed green line indicates where the target vehicle burn begins. The red vertical dashed line indicates where the target vehicle burn ends. The final row of plots shows the average number of filter nodes used at each time step across the MCs.

Figure **??** shows the MC results from each filter for the target relative position errors versus time. Each of the y axis limits are set to the the same for ease of comparison. It is clear that the filters all produce competitive relative position errors surrounding the time of the burn. The BALOO relative position errors show small spikes in the error estimates at the beginning and end of the burn, while the seven model and two model IMMs do not. However, the BALOO algorithm returns errors and error interval estimates that are smaller than those of the IMMs.

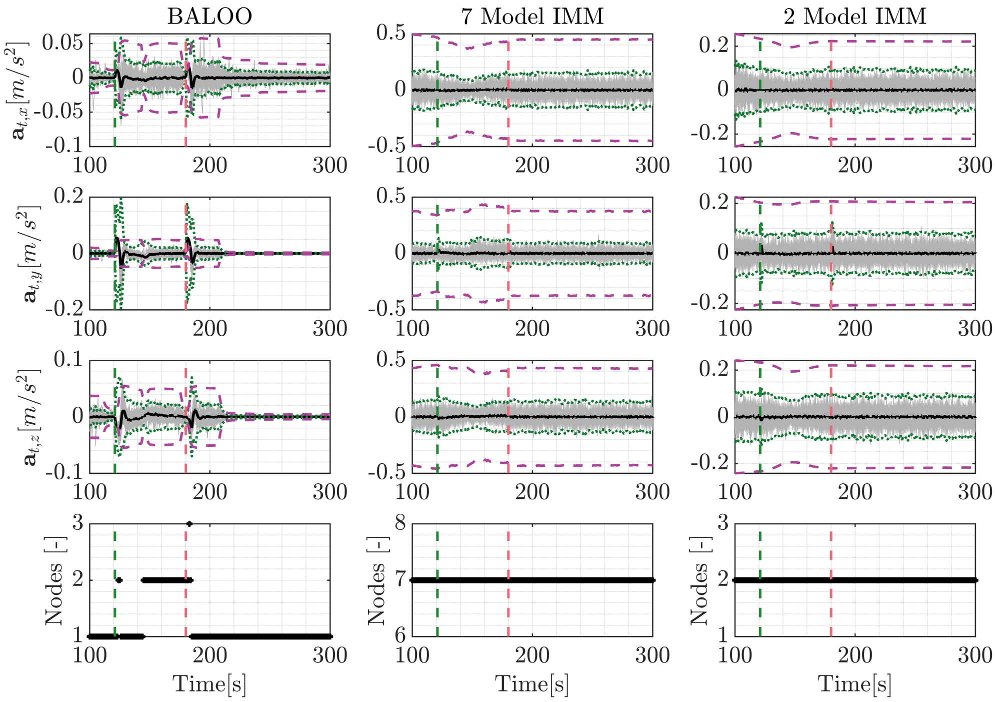

Fig. 4 Error in the unmodeled target acceleration state versus time for each filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs. The vertical dashed green line indicates where the target vehicle burn begins. The red vertical dashed line indicates where the target vehicle burn ends. The final row of plots shows the average number of filter nodes used at each time step across the MCs.

Figure **??** shows the MC results from each filter for the unmodeled target acceleration errors versus time. In this set of plots, the y axis limits are set individually to provide the best visibility of the data. The y axis limits of the results from the BALOO algorithm are significantly different from the y limits of either IMM plots. While the relative position error estimates were fairly similar from one algorithm to the next, the BALOO algorithm produces target acceleration results that are significantly better than those of the IMMs. The spikes in the average error of the BALOO estimate are larger than those of the IMMs. However, as an adaptive algorithm, there must be a detectable departure from the expected behavior in order to trigger adaptation. This can be seen in Figure **??**. The average measurement innovation shows a lag for the perturbation to produce a detectable departure from expected to induce a split. Additionally, the sample error intervals and the average estimated error intervals are a far better match for the BALOO algorithm than either of the IMMs. Finally, once the burn has concluded the BALOO algorithm quickly adjusts the estimated error intervals to capture the reduction in model uncertainty.

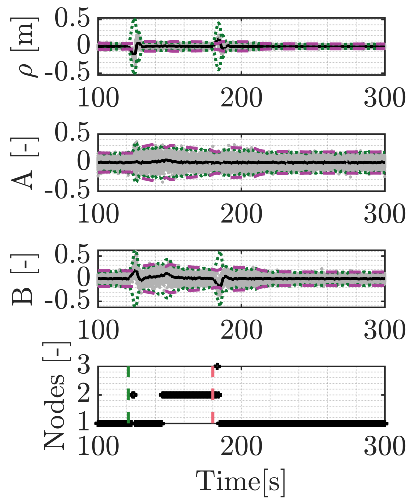

Fig. 5 Prefit measurement innovations versus time for the BALOO filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs. The vertical dashed green line indicates where the target vehicle burn begins. The red vertical dashed line indicates where the target vehicle burn ends.

It should be noted that the final row of plots in Figures **??**-**??** contains the average number of models that are active in the system at any given time. The BALOO algorithm is initialized with a single model, compared with the two and seven model IMMs. Once a maneuver is detected, the BALOO algorithm jumps to two and up to three models at the end of the maneuver. However, shortly after the maneuver concludes, the BALOO algorithm returns to just one model. The use of fewer models in the system allows for a lighter computational burden compared with systems with more models in use.

Shifting focus to the end of the scenario where steady state behavior is observed. Figures **??** and **??** show the results from 1000 s to the end of the scenario at 3000 s. In Figure **??**, each of the y axis limits are set to the the same for the three filters for ease of comparison. Similar to the results surrounding the burn, the error in the target relative position is fairly competitive between the BALOO, seven model IMM, and two model IMM. Again, similar to the time surrounding the burn, the BALOO algorithm produces both errors and error intervals that are smaller than those of the two IMM filters.

Figure **??** shows the MC results from each filter for the unmodeled target acceleration errors at steady state. In this set of plots, the y axis limits of each plot differ in order to provide the best visibility of the data; this is necessary since the BALOO algorithm significantly outperforms the IMMs. The BALOO algorithm not only provides smaller errors and error intervals, but also the average estimated intervals produced from the BALOO algorithm are a better match when compared with the sample intervals. The process noise adaptation applied to the BALOO algorithm allows the intervals to be adjusted as needed. The seven model IMM on the other hand must include acceleration models that can capture the burn thereby increasing the uncertainty intervals. Similarly, the two model IMM must include a high noise model that is large enough to accommodate for the burn and in turn also increases the noise floor of the filter.

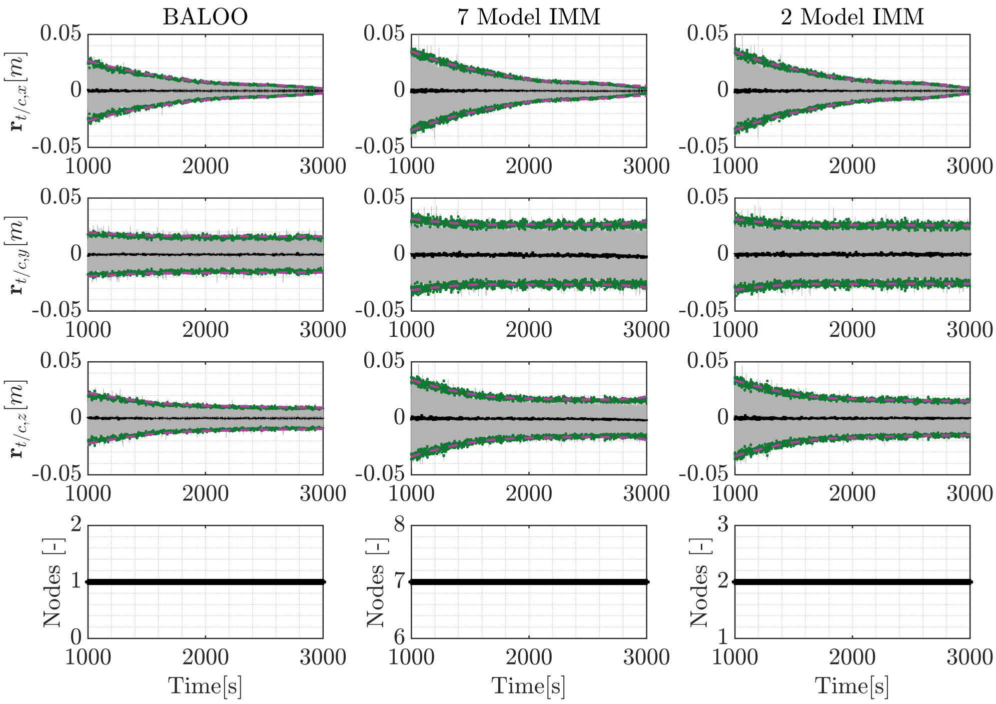

Fig. 6 Error in the relative target position state versus time for each filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs.

In Figures **??**–**??**, it can be seen that the adaptive BALOO algorithm is capable of providing superior 3σ interval estimates that are a closer match to the sample 3σ of the MC in addition to providing real time information on the unmodeled target vehicle burn.

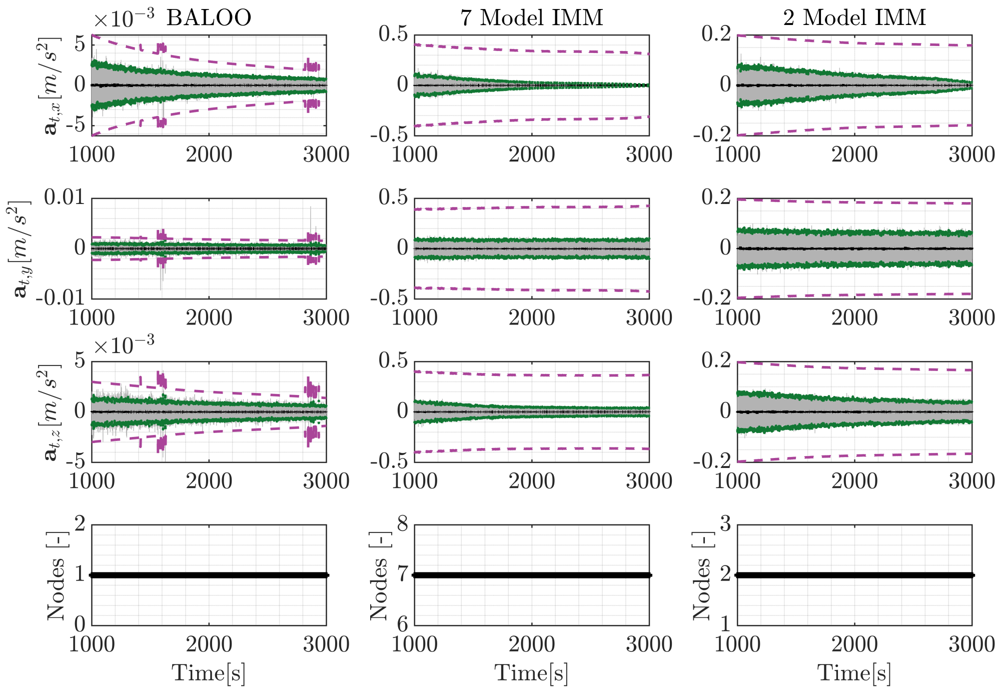

Fig. 7 Error in the unmodeled target acceleration state versus time for each filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs.

Another important metric to incorporate in the steady state discussion is that the BALOO algorithm operates with an average of just one model after the burn has passed and the discrepancy between the truth and the filter dynamics is small. This can be seen in the bottom rows of Figures **??**-**??**. With fewer models, the BALOO algorithm provides superior steady state performance.

Table 7 Burn information for the cislunar scenario.

Vehicle	Burn Start $[s]$	Burn Length $[s]$	Magnitude $[m/s^2]$	Rotating Frame Direction
Target	351	30	1e-1	$[0.5774, 0.5774, 0.5774]^T$
Chaser	2850	30	1e-3	$[0.5774, 0.5774, 0.5774]^T$

4.2 Cislunar Orbit

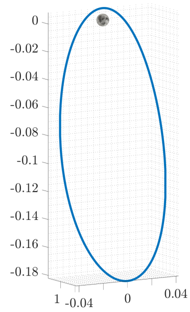

Fig. 8 Desired final orbit trajectory plotted in the rotating frame centered on the barycenter of the Earth and Moon.

4.2.1 Scenario Setup

In the cislunar scenario, both vehicles are subject to Circular Restricted Three Body Problem (CR3BP) dynamics,

$$\mathbf{g}_{\text{CR3BP}} = \begin{bmatrix} 2v_y + r_x - \frac{(1-\mu_{\text{CR3BP}})(r_x+\mu_{\text{CR3BP}})}{r_1^3} - \frac{\mu_{\text{CR3BP}}(r_x-1+\mu_{\text{CR3BP}})}{r_2^3} \\ -2v_x + r_y - \frac{(1-\mu_{\text{CR3BP}})r_y}{r_1^3} - \frac{\mu_{\text{CR3BP}} r_y}{r_2^3} \\ \frac{-(1-\mu_{\text{CR3BP}})r_z}{r_1} - \frac{\mu_{\text{CR3BP}} r_z}{r_2^3} \end{bmatrix},$$
$$r_1 = \sqrt{(r_x + \mu_{\text{CR3BP}})^2 + r_y^2 + r_z^2} \quad r_2 = \sqrt{(r_x - 1 + \mu_{\text{CR3BP}})^2 + r_y^2 + r_z^2}, \tag{26}$$

where the state is represented in the nondimensional rotating frame, $\mu_{\text{CR3BP}} = 1.215e-2$ is the nondimensional mass ratio between the Earth and Moon, 3.752e5 s is the characteristic time, and 3.844e8 m is the characteristic length. It should be noted that this simulated cislunar scenario takes place around perilune. Again, in [24], it is shown that the venting perturbations overwhelm gravity modeling errors. Thus the gravity model used in the true dynamics is the same as the model used in the filter. The true dynamics on the target vehicle include regular venting perturbations as well as a finite burn. Again, the filter dynamics model for the target vehicle does *not* include any prior information on the venting perturbations. The chaser vehicle also performs a finite burn in order to rendezvous with the target vehicle. The burn information can be found in Table **??**.

In each MC run, the vehicles end in the same final orbit, shown in Table **??**. In

Table 8 Final nondimensional and rotating states for the NRHO scenario.

Relative Target	-7.356e-9	-1.644e-10	-5.203-9	-1.644e-10	7.356e-10	0.000
Chaser	9.877e-1	1.088e-2	4.767e-3	6.334e-2	1.151	-7.886e-1

order to ensure that the vehicles end in the same final orbit, each truth scenario is generated by back propagating from the final conditions to the beginning of the scenario. In a similar manner to the LEO scenario, each MC has its own set of randomly sampled target vehicle venting perturbations. The timing of the first CO_2 venting perturbation is randomly sampled from a uniform distribution within the 30 s. From there each CO_2 perturbation occurs at the frequency in Table **??**. Due to the infrequency of the waste water vents, only one is modeled in the scenario. The waste water event occurs 1400s after the first CO_2 vent. In addition to the timing of the vents, the magnitude of each vent is also randomized by randomly sampling from a Gaussian distribution centered on zero with an uncertainty listed in Table **??**, then taking the absolute value of the sample. In the event a venting perturbation occurs during the target vehicle burn, the vent is skipped.

Measurements are received every second. The true chaser accelerometer bias is initialized to be to be 2.9430e-6 m/s^2 for each axis. The velocity random walk of the chaser accelerometer is set to be 3.433e-6 $m/s^2/\sqrt{s}$, these values were taken from [24] which adjusted the values originally taken from [3] and altered them to suit the study. The discrete time process noise is set to be 1.179e-9$\mathbf{I}m/s^2$. The inverse of the time constant for chaser accelerometer bias is set to be $\beta = 0 s^{-1}$ for each axis.

4.2.2 BALOO Setup

To initialize each filter run, the initial state is sampled from a Gaussian distribution, centered on the true initial conditions with a covariance matrix,

$$\mathbf{P}_0 = \begin{bmatrix} (2.5e-9)^2\,\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & (2.5e-11)^2\,\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & (3.5\text{e-}7)^2\mathbf{I}_{3\times3} \end{bmatrix}, \tag{27}$$

The covariance matrix is dimensionless and in the rotating frame. In addition to the initial state of the filter, the initial estimate of the filter process noise is also randomized. Each element of the diagonal of the initial process noise in the filter is sampled from a discrete uniform distribution from the set $\left[10^{-22}, 10^{-20}, ..., 10^{-10}, 10^{-8}\right]$. This random sample is added to an initial guess of $\mathbf{s}_0 = 10^{-4}\mathbf{1}_{3x1}$. The initial covariance matrix of the process noise estimate is set to be $\mathbf{\Psi}_0 = 1e-6\mathbf{I}_{3x3}$. In the adaptive noise filter for this scenario, the measurement noise matrix is set to be $\mathbf{R}_w = 1e-6\mathbf{I}$. The process noise matrix for the adaptive noise filter is set to be $\mathbf{Q}_w = 1e-10\mathbf{I}$. The size of the adaptation buffer for this scenario is set to be $m_b = 15$, while the split decision buffer is set to be $m_s = 10$. The decision to split is made when the SNDIS exceeds a value of 60. Nodes are considered statistically insignificant and pruned off it the node's weight falls below 1e–12. When a node is pruned, the node is simply deleted.

4.2.3 Seven Model IMM Setup

In the seven model IMM, each of the models is treated as an EKF with fixed process noise. Each node applies its model through a consider unmodeled target acceleration state. A table with the acceleration vector of each node is shown below.

Table 9 Acceleration models for each IMM node.

Node	1	2	3	4	5	6	7
$\mathbf{a}_{j,x}$	0.1	-0.1	0	0	0	0	0
$\mathbf{a}_{j,y}$	0	0	0.1	-0.1	0	0	0
$\mathbf{a}_{j,z}$	0	0	0	0	0.1	-0.1	0

In the interacting portion of the IMM, only the non-consider states and covariance estimates are combined. The probability that the model stays the same is 1/3 for each model. The transitional probabilities are all set to be 1/9. The process noise is set to be zero for the target vehicle states. The continuous time process noise matrix for the chaser vehicle dynamics states is set to be $\mathbf{Q}_c = \mathbf{0}_{3x3}$, and the process noise on the chaser accelerometer bias is set to be 1.179e-9$\mathbf{I}m/s^2$.

4.2.4 Two Model IMM Setup

In the two model IMM, each of the models is treated as an EKF with fixed process noise. The two models are a high noise and low noise EKF. In each model, the unmodeled target acceleration is estimated. Again, the inclusion of the unmodeled target acceleration states in the Two Model IMM functions the same as in the adaptive BALOO filter.

Table 10 Continuous time process noise matrices applied to the relative target vehicle states for each IMM node.

Node	1	2
$\mathbf{Q}_t$	$1e7\mathbf{I}_{3x3}$	$1e-12\mathbf{I}_{3x3}$

In the interacting portion of the IMM, only the non-consider states and covariance estimates are combined. The probability that the model stays the same is 2/3 for each model. The transitional probability is set to be 1/3. The continuous time process noise matrix for the chaser vehicle dynamics states is set to be $\mathbf{Q}_c = 1e - 12\mathbf{I}_{3x3}$, and the process noise on the chaser accelerometer bias is set to be 1.179e-9$\mathbf{I}m/s^2$.

4.2.5 Results

Again, the analysis of the results begins by examining the time surrounding the unmodeled target burn. The burn begins at 351s and concludes at 381s.

Figure **??** shows the MC results from each filter for the target relative position errors versus time. Each of the y axis limits are set to the the same for ease of comparison. The results of the NRHO scenario are very similar to those of the LEO scenario. All filters produce competitive relative position errors around the time of the burn. The BALOO relative position errors show small spikes in the error estimates at the beginning and end of the burn while the SNDIS grows due to the unmodeled change in the target acceleration, which can be visualized in Figure **??** by examining the behavior of the average innovation near the beginning and end of the burn. The IMMs are both tuned to the burn itself and take little time to transition to the correct combination of models to represent the burn.

Figure **??** shows the MC results from each filter for the unmodeled target acceleration errors versus time. The y axis limits are set to be the same for ease of comparison. The seven model IMM provides errors with a sustained bias over the burn period, however, the errors are well bounded within the 3σ interval estimates. The BALOO algorithm displays spiked in the average error at the beginning and end of the burn, but quickly detects the error and corrects its estimate of the unmodeled target acceleration. The BALOO estimated 3σ intervals rise and fall in a similar manner to the errors themselves. The two model IMM results show similar spikes at the beginning and end of the burn, but the overall 3σ error bounds are smaller than those of the BALOO or seven model IMM. The two model IMM likely has smaller errors than the BALOO algorithm of the seven model IMM in the NRHO example due to the larger magnitude of the maneuver in the NRHO scenario compared with the LEO example. The BALOO algorithm still provides an excellent estimate given that it has no a priori information on the burn, but takes on more error due to the larger burn causing a faster divergence between the true and the filtered dynamics. The seven model IMM on the other hand has a similar level of a priori information to the two model. The seven model IMM has larger covariance intervals due to the impact of the magnitude of the acceleration models on the covariance estimate.

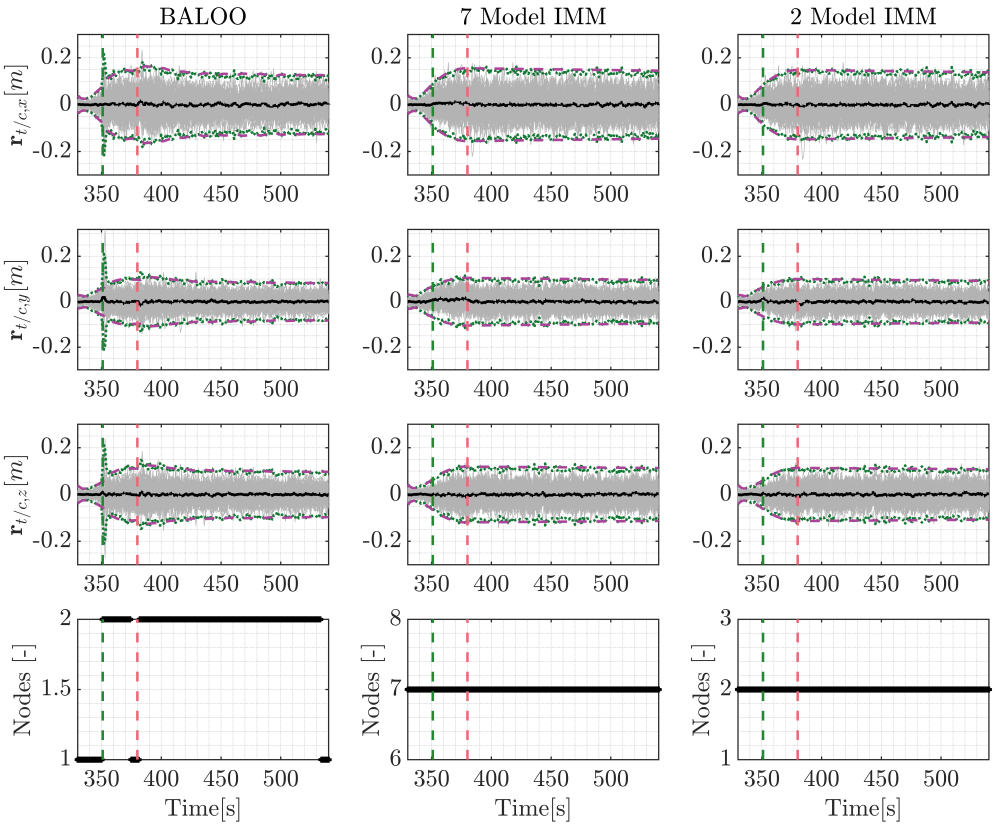

Fig. 9 Error in the relative target position state versus time for each filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs. The vertical dashed green line indicates where the target vehicle burn begins. The red vertical dashed line indicates where the target vehicle burn ends.

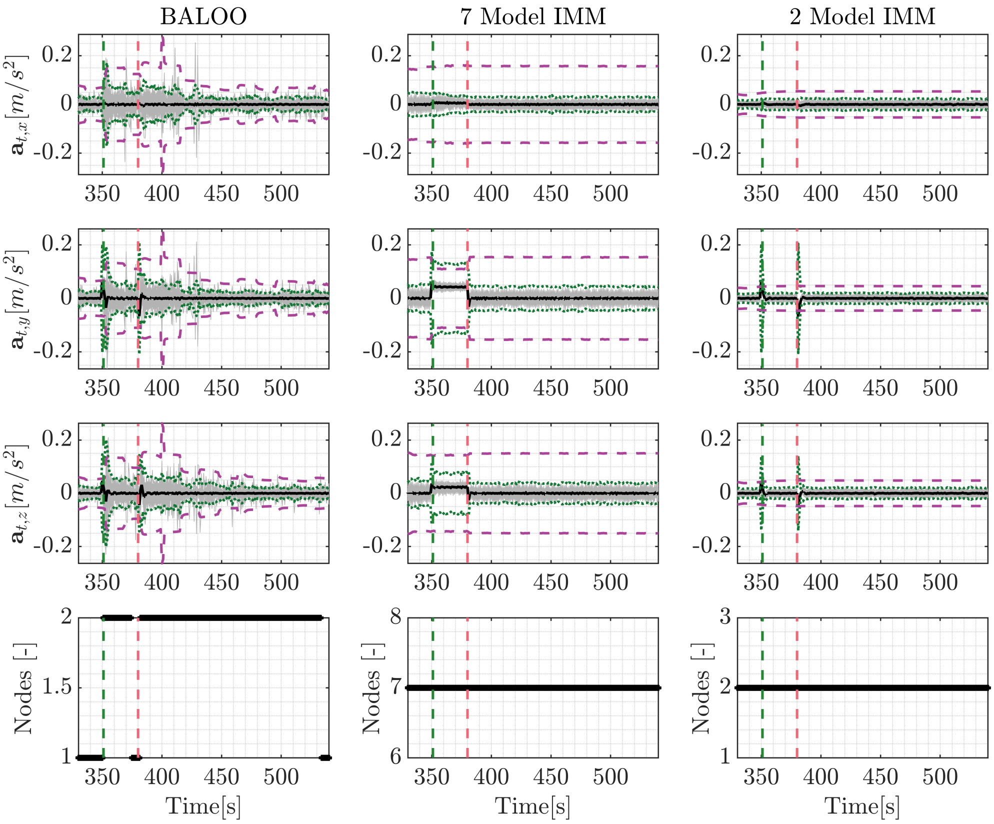

Fig. 10 Error in the unmodeled target acceleration state versus time for each filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs. The vertical dashed green line indicates where the target vehicle burn begins. The red vertical dashed line indicates where the target vehicle burn ends.

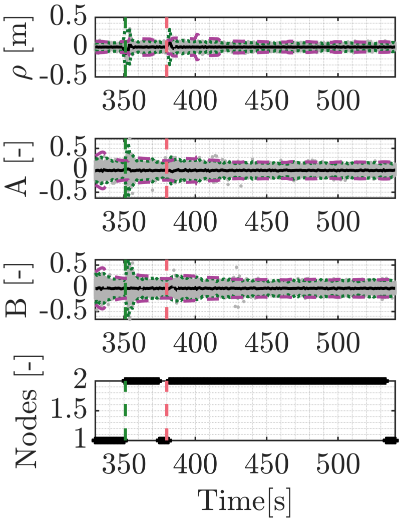

Fig. 11 Prefit measurement innovations versus time for the BALOO filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs. The vertical dashed green line indicates where the target vehicle burn begins. The red vertical dashed line indicates where the target vehicle burn ends.

Once again, examining the bottom row of plots in Figures **??**-**??**, the average number of models that are active in the system at any given time can be seen. Again, the BALOO algorithm is initialized with a single model, compared with the two and seven model IMMs. Once a maneuver is detected, the BALOO algorithm jumps to two models in use. However, shortly after the maneuver concludes, the BALOO algorithm returns to just one model. Using fewer models in the system allows for a lighter computational burden compared with systems with more models in use.

The next set of figures depict the end of the scenario where steady state behavior is observed. Figures **??** and **??** show the results from 1000s to the end of the scenario at 3000s. In Figure **??**, each of the y axis limits are set to the the same for the three filters for ease of comparison. Again, the differences between each of the filters' relative position error estimates are very similar.

Figure **??** shows the MC results from each filter for the unmodeled target acceleration errors at steady state. In this set of plots, the y axis limits are set individually to provide the best visibility of the data. The BALOO algorithm not only provides smaller errors and error intervals, but also the average estimated intervals produced from the BALOO algorithm are a better match when compared with the sample intervals. The process noise adaptation applied to the BALOO algorithm allows the intervals to be adjusted as needed. The two IMM filters are tuned with prior information about the burn in order to be able to capture such a large perturbation. This tuning leads to an overly conservative filter in the areas of smaller model uncertainty. If the IMM filters were tuned to match the steady state behavior, the IMM filters would experience similar performance drawbacks around the burn.

In Figures **??** and **??** it can be seen that BALOO displays comparative results with the two IMM filters that are tuned prior to the scenario. In Figures **??** and **??**, it can be seen that the adaptive BALOO algorithm is capable of providing superior 3σ interval estimates that are a closer match to the sample 3σ of the MC in the steady state.

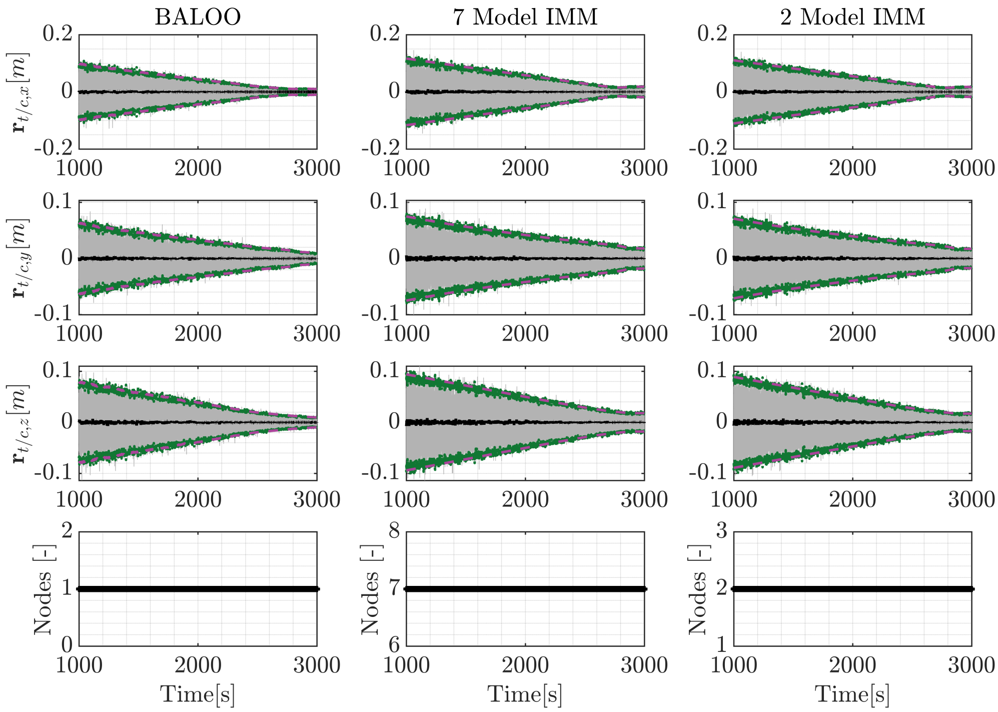

Fig. 12 Error in the relative target position state versus time for each filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs.

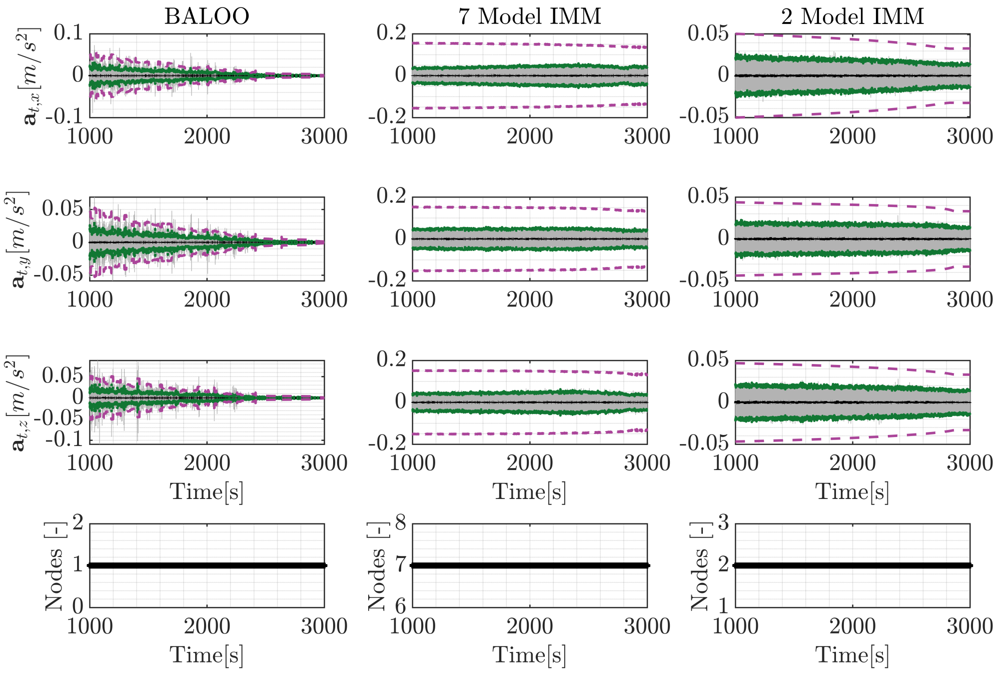

Fig. 13 Error in the unmodeled target acceleration state versus time for each filter. The black line is the average error across all 100 MC. The gray lines are the ith MC estimation errors. The green dotted lines are the sample 3σ interval across all MC runs. The purple dashed lines is the average filter estimated 3σ intervals across all MC runs.

Again, the BALOO algorithm operates with an average of just one model after the burn has passed and the discrepancy between the truth and the filter dynamics is small. This can be seen in the bottom rows of Figures **??**-**??**. Again, with fewer models, the BALOO algorithm provides superior steady state performance when compared with the two and seven model IMMs.

5 Conclusions

In this work an onboard adaptive strategy to handle close proximity unmodeled target vehicle burns is proposed. Much of the current literature on burn detection has been focused on offline algorithms and on algorithms that have a certain degree of a priori information on the system. It is shown for both a LEO and cislunar example that the BALOO algorithm is capable of responding to unmodeled target vehicle burns. The BALOO algorithm has no a priori information on either the burn or the venting perturbations and is still capable of providing real time estimates of the target vehicle's unmodeled acceleration.

With the desire to pursue more complex space missions, the ability to adapt onboard to observed departures from the expected becomes more necessary. Complexity adds risk, and being able to respond quickly to observed differences improves mission safety. Estimating an upper bound for unexpected behavior in the form of process noise has a great deal of heritage in the aerospace industry. However, there are cases in which these estimates become obsolete due to unforeseen events that impact the dynamics of the spacecraft. In many cases, these differences may be small, and simply adaptively adjusting the process noise estimate is sufficient to overcome the perturbation. In more severe model mismatch cases, the splitting algorithm proposed in BALOO is capable of providing real time adjustments to capture these mismatches. Real time detection of a burn could be the difference between a safe, successful rendezvous and a catastrophic collision as it would provide a more accurate relative velocity for collision avoidance algorithms to operate on. By providing solutions to compensate for both small and large perturbations, the BALOO algorithm provides a robust solution to poorly modeled or unknown dynamics perturbations. Additionally, the BALOO algorithm is capable of providing competitive to superior unmodeled target acceleration estiation when compared with a priori tuned IMM filters which utilize more models on average than the BALOO algorithm.

The work presented in this paper is focused on providing an initial assessment of the ability of the BALOO algorithm to detect and provide estimates of severe model mismatches in real time. In future work, it would be beneficial to examine alternative solutions where the relative target vehicle position is not instantaneously observable. Restricting the measurements to either range only or angles only could provide insight into the applicability of this algorithm for use on lower budget missions with more restricted measurement capabilities. Another avenue for future work is to investigate more realistic attitude modeling. More realistic attitude might include both adding in attitude estimation errors as well as imperfect tracking of the target vehicle during both translational or rotational maneuvers. The inclusion of attitude modeling errors poses interesting complications between attitude errors and the relative state estimation. Incorporating imperfect tracking would provide an additional avenue for unmodeled burn detection through the target vehicle straying from the center of the field of view during an unmodeled burn. However, this potential benefit of detection may add a great deal more to the complexity of the algorithms, as this could add outages and gaps in the measurement stream. Finally, it would also be beneficial to investigate modeling the impacts of sensors with dynamically changing resolutions that correspond to the relative distance of the two vehicles. The work in this paper assumes that the sensors provide consistent noise characteristics and resolution regardless of relative distance. Sensor performance variability is a reality that many missions must work around, thus it is important to investigate scenarios in which dynamically changing sensors are modeled.

Funding Sources

This work was supported by a NASA Space Technology Graduate Research Opportunity, NASA grant number: 80NSSC22K1204.

References

[1] Farzin Amzajerdian et al. “Lidar Sensors for Autonomous Landing and Hazard Avoidance”. In: *AIAA SPACE 2013 Conference and Exposition*. 2013, p. 5312.

[2] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2001.

[3] Northrop Grumman Corporation. *LN-200 Inertial Measurement Unit Brochure*. Accessed on Febuary, 2024. 2023. URL: https://cdn.prd.ngc.agencyq.site/-/media/wp-content/uploads/LN-200S-Inertial-Measurement-Unit-IMU-datasheet.pdf.

[4] Chris D D’Souza and Renato Zanetti. “A First Look at the Navigation Design and Analysis for the Orion Exploration Mission 2”. In: *AAS/AIAA Astrodynamics Specialist Conference 2017*. JSC-CN-40226. 2017.

[5] Christopher D’Souza and Renato Zanetti. “Navigation Design and Analysis for the Orion Earth-Moon Mission”. In: *AAS/AIAA Space Flight Mechanics Conference*. JSC-CN-30283. 2014.

[6] Diane Davis et al. “Orbit Maintenance and Navigation of Human Spacecraft at Cislunar Near Rectilinear Halo Orbits”. In: *AAS/AIAA Space Flight Mechanics Meeting*. JSC-CN-38626. 2017.

[7] Kyle J DeMars, Robert H Bishop, and Moriba K Jah. “Entropy-Based Approach for Uncertainty Propagation of Nonlinear Dynamical Systems”. In: *Journal of Guidance, Control, and Dynamics* 36.4 (2013), pp. 1047–1057. DOI: 10.2514/1.58987.

[8] M. Efe and D.P. Atherton. “Maneuvering Target Tracking with an Adaptive Kalman Filter”. In: *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*. Vol. 1. 1998, 737–742 vol.1. DOI: 10.1109/CDC.1998.760773.

[9] Felipe Giraldo-Grueso, Andrey A Popov, and Renato Zanetti. “Adaptive Mart Entry Guidance with Atmospheric Density Estimation”. In: *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Broomfield, Colorado*. 2024.

[10] Felipe Giraldo-Grueso, Andrey A Popov, and Renato Zanetti. “Precision Mars Entry Navigation with Atmospheric Density Adaptation Via Neural Networks”. In: *Journal of Aerospace Information Systems* (2024), pp. 1–14. DOI: 10.2514/1.I011426.

[11] Cheryl Gramling et al. “Preliminary Operational Results of the TDRSS Onboard Navigation System (TONS) for the Terra Mission”. In: *Spaceflight Dynamics*. 2000.

[12] Marcus J Holzinger, Daniel J Scheeres, and Kyle T Alfriend. “Object Correlation, Maneuver Detection, and Characterization Using Control Distance Metrics”. In: *Journal of Guidance, Control, and Dynamics* 35.4 (2012), pp. 1312–1325.

[13] Yulong Huang et al. “Robust Kalman Filters Based on Gaussian Scale Mixture Distributions With Application to Target Tracking”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.10 (2019), pp. 2082–2096. DOI: 10.1109/TSMC.2017.2778269.

[14] Andris D Jaunzemis, Midhun V Mathew, and Marcus J Holzinger. "Control Cost and Mahalanobis Distance Binary Hypothesis Testing for Spacecraft Maneuver Detection". In: *Journal of Guidance, Control, and Dynamics* 39.9 (2016), pp. 2058–2072.

[15] Jack A. Joshi and Christopher N. D'Souza. "Modeling Venting Disturbances Within Linear Covariance Toolset for Crewed Artemis Missions". In: *AAS/AIAA Astrodynamics Specialist Conference*. 2023.

[16] Jack A. Joshi et al. "Artemis II: Linear Covariance Analysis Using Explicit Vent Force States". In: *AAS/AIAA Astrodynamics Specialist Conference*. 2024.

[17] Rudolph E Kalman and Richard S Bucy. "New Results in Linear Filtering and Prediction Theory". In: *Transactions of the ASME, Journal of Basic Engineering* 83.1 (1961), pp. 95–108. DOI: 10.1115/1.3658902.

[18] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME, Journal of Basic Engineering* 82.1 (1960), pp. 35–45. DOI: 10.1115/1.3662552.

[19] Jackson Kulik and Keith A. LeGrand. *Nonlinearity and Uncertainty Informed Moment-Matching Gaussian Mixture Splitting*. 2024. DOI: 10.48550/arXiv.2412.00343. arXiv: 2412.00343 [stat.ML].

[20] J.R. Layne and U.C. Piyasena. "Adaptive Interacting Multiple Model Tracking of Manuevering Targets". In: *16th DASC. AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. Proceedings*. Vol. 1. 1997, pp. 5.3–16. DOI: 10.1109/DASC.1997.635095.

[21] X. Rong Li and Yaakov Bar-Shakm. "Mode-Set Adaptation in Multiple-Model Estimators for Hybrid Systems". In: *1992 American Control Conference*. 1992, pp. 1794–1799. DOI: 10.23919/ACC.1992.4792420.

[22] Xiao-Rong Li and Y. Bar-Shalom. "Multiple-model estimation with variable structure". In: *IEEE Transactions on Automatic Control* 41.4 (1996), pp. 478–493. DOI: 10.1109/9.489270.

[23] Daniel P Lubey and Daniel J Scheeres. "An Optimal Control Based Estimator for Maneuver and Natural Dynamics Reconstruction". In: *Proceedings of the 2013 Advanced Maui Optical and Space Surveillance Technologies Conference*. 2013.

[24] Rachel Mamich and Renato Zanetti. "Adaptation to Venting Disturbances for Onboard Navigation". In: *Journal of Guidance, Control, and Dynamics* 49.2 (2026), pp. 577–587. DOI: 10.2514/1.G009037. eprint: https://doi.org/10.2514/1.G009037. URL: https://doi.org/10.2514/1.G009037.

[25] E. Mazor et al. "Interacting Multiple Model Methods in Target Tracking: A Survey". In: *IEEE Transactions on Aerospace and Electronic Systems* 34.1 (1998), pp. 103–123. DOI: 10.1109/7.640267.

[26] James S McCabe and Kyle J DeMars. "Square–Root Consider Filters with Hyperbolic Householder Reflections". In: *Journal of Guidance, Control, and Dynamics* 41.10 (2018), pp. 2098–2111. DOI: 10.2514/1.G003417.

[27] Melissa L McGuire, Steven L McCarty, and Laura M Burke. *Power and Propulsion Element (PPE) Spacecraft Reference Trajectory Document*. Tech. rep. 2020.

[28] Raman Mehra. "On the Identification of Variances and Adaptive Kalman Filtering". In: *IEEE Transactions on Automatic Control* 15.2 (1970), pp. 175–184. DOI: 10.1109/TAC.1970.1099422.
[29] Rahul Moghe, Renato Zanetti, and Maruthi Akella. "Covariance Matching Kalman Filter for Observable LTI Systems". In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 6372–6377. DOI: 10.1109/CDC.2018.8619203.
[30] A. Munir and D.P. Atherton. "Maneuvering Target Tracking Using an Adaptive Interacting Multiple Model algorithm". In: *Proceedings of 1994 American Control Conference - ACC '94*. Vol. 2. 1994, 1324–1328 vol.2. DOI: 10.1109/ACC.1994.752274.
[31] K. Myers and B. Tapley. "Adaptive Sequential Estimation with Unknown Noise Statistics". In: *IEEE Transactions on Automatic Control* 21.4 (1976), pp. 520–523. DOI: 10.1109/TAC.1976.1101260.
[32] Clark P Newman et al. "Stationkeeping, Orbit Determination, and Attitude Control for Spacecraft in Near Rectilinear Halo Orbits". In: *AAS Astrodynamics Specialists Conference*. JSC-E-DAA-TN59836. 2018.
[33] H. A. P. Blom. "An Efficient Filter for Abruptly Changing Systems". In: *The 23rd IEEE Conference on Decision and Control*. 1984, pp. 656–658. DOI: 10.1109/CDC.1984.272089.
[34] Murali R. Rajamani and James B. Rawlings. "Estimation of the Disturbance Structure from Data Using Semidefinite Programming and Optimal Weighting". In: *Automatica* 45.1 (2009), pp. 142–148. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2008.05.032. URL: https://www.sciencedirect.com/science/article/pii/S000510980800366X.
[35] Duane Roth et al. "Cassini In-Flight Navigation Adaptations". In: *2018 SpaceOps Conference*. 2018, p. 2647.
[36] Andrew P. Sage and G. W. Husa. "Adaptive Filtering with Unknown Prior Statistics". In: *Proceedings of the Joint Automatic Control Conference*. 1969, pp. 760–769.
[37] Bob Schutz, Byron Tapley, and George H Born. *Statistical Orbit Determination*. Elsevier, 2004.
[38] Emily M Zimovan Spreen. "Trajectory Design and Targeting for Applications to the Exploration Program in Cislunar Space". PhD thesis. Purdue University, 2021.
[39] Nathan Stacey and Simone D'Amico. "Adaptive and Dynamically Constrained Process Noise Estimation for Orbit Determination". In: *IEEE Transactions on Aerospace and Electronic Systems* 57.5 (2021), pp. 2920–2937.
[40] Jason Stauch and Moriba Jah. "Unscented Schmidt–Kalman Filter Algorithm". In: *Journal of Guidance, Control, and Dynamics* 38.1 (2015), pp. 117–123. DOI: 10.2514/1.G000467.
[41] Kirsten Tuggle and Renato Zanetti. "Automated Splitting Gaussian Mixture Nonlinear Measurement Update". In: *Journal of Guidance, Control, and Dynamics* 41.3 (2018), pp. 725–734. DOI: 10.2514/1.G003109.

[42] Kirsten Elizabeth Tuggle. "Model Selection for Gaussian Mixture Model Filtering and Sensor Scheduling". PhD thesis. 2020.

[43] David A Vallado. *Fundamentals of Astrodynamics and Applications*. Vol. 12. Springer Science & Business Media, 2001.

[44] Renato Zanetti and Chris D'Souza. "Dual Accelerometer Usage Strategy for Onboard Space Navigation". In: *Journal of Guidance, Control, and Dynamics* 35.6 (2012), pp. 1899–1902. DOI: 10.2514/1.58154.

[45] Renato Zanetti et al. "Absolute Navigation Performance of the Orion Exploration Flight Test 1". In: *Journal of Guidance, Control, and Dynamics* 40.5 (2017), pp. 1106–1116. DOI: 10.2514/1.G002371.

[46] Enrico M. Zucchelli and Brandon A. Jones. "A Bayesian Approach to Low-Thrust Maneuvering Spacecraft Tracking". In: *Journal of Guidance, Control, and Dynamics* (Apr. 2024), pp. 1–16. ISSN: 1533-3884. DOI: 10.2514/1.g007849. URL: http://dx.doi.org/10.2514/1.G007849.