IMPROVED ORBIT DETERMINATION USING SECOND-ORDER GAUSS-MARKOV PROCESSES

Felipe G. Nievinski,* Brandon Yonko,† and George H. Born‡

Confronted with fundamental uncertainties regarding orbital system dynamics and given the abundance of tracking data from GPS, the estimation of empirical accelerations has become standard practice in precise orbit determination (POD) of near-Earth satellites. In sequential filtering an exponentially time-correlated system noise process is typically postulated, corresponding to a first-order Gauss-Markov process (GMP). In the present work we address the possibility of using second-order GMP in POD. Improvements are demonstrated for sparse and intermittent tracking as well as for orbit predictions at unobserved future epochs, in terms of both more accurate states and more realistic covariance envelopes.

INTRODUCTION

Confronted with fundamental uncertainties regarding orbital system dynamics and given the abundance of tracking data from, e.g., GPS, the estimation of empirical accelerations has become standard practice in precise orbit determination (POD) of Earth-orbiting satellites.¹ By acknowledging the imperfections in the dynamic model, this approach has become known generically as reduced dynamic modeling (RDM).²,³

There are several variants of RDM. A common instance is known as dynamical model compensation (DMC), and it consists in postulating an exponentially time-correlated system noise process in sequential filtering.⁴ Such a formulation, parameterized in terms of a variance and a correlation time, corresponds to a first-order Gauss-Markov process (GMP).⁵ Another well-established RDM practice is the estimation of trigonometric coefficients at fixed frequencies (e.g., one cycle per revolution).⁶ A third option lies in physically-based auto-covariance models describing, e.g., errors of commission and omission in gravity.⁷ A fourth variant exists in batch estimation, where piecewise-constant time-dependent accelerations can be included as unknown parameters.⁸

The seminal paper introducing first-order GMP (GMP1) for POD raised the possibility that “a second-order GMP [GMP2] could provide even better estimation results.”⁵ In the present work we address that prospect. It could be argued that the mathematical formulation of GMP2 is well-

---

¹ Ph.D. candidate, Department of Aerospace Engineering, University of Colorado, Boulder, CO 80309-0431
² Master’s student, Department of Aerospace Engineering and Colorado Center for Astrodynamics Research, University of Colorado, Boulder, CO 80309-0431 (now at: Lincoln Laboratory, Massachusetts Institute of Technology, 244 Wood Street, Lexington, MA 02420-9108)
³ Professor, Department of Aerospace Engineering Sciences and Director, Colorado Center for Astrodynamics Research, University of Colorado, Boulder, CO 80309-0431
known in the control literature. Yet it is our impression that GMP2 are not employed for routine POD as widely as they perhaps deserve to be. For example, none of the main textbooks on POD discusses GMP2 in their current editions.

We argue that the capability of GMP2 for modeling random, approximately oscillatory phenomena makes it adequate in accounting for a variety of accelerations that tend to repeat as function of satellite orbital revolution, such as gravity, atmospheric drag, and solar radiation pressure. GMP2 could serve as a generic replacement for more specialized, physically-based auto-covariance models, not to mention accounting for accelerations lacking such a more rigorous auto-covariance description. GMP2 offers a more flexible auto-covariance model, that embraces and extends the two extremes typically adopted in RDM, between a pure exponential decay and a pure sinusoidal oscillation (see Figure 1), by varying the degree of continuity in the model.

In the next section we briefly review the underlying theory concerning GMP. After that we describe the setup and discuss the results of numerical experiments comparing GMP1 and GMP2 for POD. Then we address the problem of calibrating the coefficients driving the stochastic model. The article closes with a summary and suggestions for future work. An appendix provides details on implementing GMP2.

THEORY

Here we follow closely the derivation offered by (p.180–185), though adapting it to the notation of .

Basic system, external noise-shaping filter, and augmented system

Let the state vector be denoted \( X \); it contains satellite position and velocity coordinates, as well as parameters representing some of accelerations affecting the satellite, such as atmospheric drag and gravity. Then let:

\[
\dot{X}(t) = F(t, X),
\]

\[
Y(t) = G(t, X),
\]

be the dynamic model and the observation model, respectively. The former relates the state vector to the state-rate vector \( \dot{X} \), while the latter relates the state vector to the observation vector \( Y \). We proceed linearizing these two models as:

\[
\dot{x}(t) = A(t) x(t) + B(t) n(t),
\]

\[
y(t) = \bar{H}(t) x(t) + \epsilon(t),
\]

where \( x(t) = \dot{X}(t) - X(t) \) and \( \dot{x}(t) = \dot{X}(t) - \dot{X}(t) \) are the state respectively state-rate deviation vectors, in terms of a priori state respectively state-rate vectors, \( \dot{X}(t) \) and \( \dot{X}(t) \). The matrices \( A(t) \) and \( \bar{H}(t) \) are the Jacobians:

\[
A(t) = \left. \frac{\partial F(t, X)}{\partial X} \right|_{\dot{X}(t)},
\]

\[
\bar{H}(t) = \left. \frac{\partial G(t, X)}{\partial X} \right|_{\dot{X}(t)},
\]

both of which being evaluated at the current epoch’s a priori state, \( \dot{X}(t) \).
While the observation noise vector \( \epsilon \) is postulated uncorrelated (i.e., white noise), the state noise vector \( n \) is allowed to be time-correlated. This is achieved passing a white noise vector \( w \) of suitable dimensions through a separate shaping filter:

\[
\dot{x}_s(t) = A_s(t) x_s(t) + B_s(t) w(t), \tag{7}
\]

\[
y_s(t) = \tilde{H}_s(t) x_s(t). \tag{8}
\]

Then the shaping filter output, \( n(t) = y_s(t) \), can be used to drive the basic system, \( \dot{x} = A(t) x + B(t) n(t) \). In the next sub-section we detail some of the most commonly used noise shaping filters. Instead of running two separate filters (one for each basic system and noise shaping), we may define an augmented state vector \( X_a = [x^T, x^T_s] \), where \( X_b = X \), and corresponding augmented matrices (details in the appendix):

\[
\dot{x}_a(t) = A_a(t) x_a(t) + B_a(t) w(t), \tag{9}
\]

\[
y_a(t) = \tilde{H}_a(t) x_a(t). \tag{10}
\]

**Gauss-Markov processes**

Each zeroth-, first-, and second-order Gauss-Markov process (GMP0, GMP1, GMP2) augments the basic state vector with respectively zero, one, and two parameters per spatial dimension (e.g., six parameters for a GMP2 in three-dimensional space). Below we will refer to the noise auto-covariance function,

\[
\Psi_{nn}(\tau) = E\{n(t) n(t + \tau)\}, \tag{11}
\]

where \( E[\cdot] \) denotes statistical expectation, \( n(t) \) is a scalar noise (assumed zero mean), and \( \tau \) represents time-lag between two arbitrary epochs \( \tau = t' - t \). White noise corresponds to GMP0 [Gelb, 1974] and has auto-covariance function represented in terms of Dirac’s delta function \( \delta(\cdot) \)

![Figure 1. Auto-covariance functions for different instances of second-order GMP.](image)
and a noise variance $\sigma^2$:

$$\Psi_{nn}(\tau) = \sigma^2 \delta(\tau).$$  \hfill (12)

GMP1 corresponds to exponentially time-correlated noise:

$$\Psi_{nn}(\tau) = \sigma^2 \exp(-|\tau|/T),$$  \hfill (13)

where $T$ is the correlation time. Its state description (for the scalar case) is:

$$\dot{X}(t) = -(1/T) X(t) + w(t).$$  \hfill (14)

Finally, GMP2 leads to:

$$\Psi_{nn}(\tau) = \frac{\sigma^2}{\cos \eta} \exp(-\zeta \omega |\tau|) \cos \left(\sqrt{1 - \zeta^2} \omega |\tau| - \eta\right).$$  \hfill (15)

Notice that it is periodically negative, i.e., this type of noise tends to alternate sign over time. Notice further that its degree of continuity at $\tau = 0$ can be selected by setting $\zeta$, which results in a smoother or rougher noise time series. In other words, $\zeta$ adjusts the degree of decay (exponential factor) vs. oscillation (sinusoidal factor) exhibited by the noise. Its state description (again for the scalar case) reads:

$$\begin{bmatrix} \dot{X}(t) \\ \ddot{X}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\omega \end{bmatrix} \begin{bmatrix} X(t) \\ X(t) \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} w(t),$$  \hfill (16)

where intermediary variables are:

$$a = \sqrt{2\sigma^2 \omega} \sin(\alpha - \eta)/\cos \eta,$$  \hfill (17)

$$b = \sqrt{2\sigma^2 \omega} \sin(\alpha + \eta)/\cos \eta,$$  \hfill (18)

$$\alpha = \arctan \left(\zeta/\sqrt{1 - \zeta^2}\right).$$  \hfill (19)

In practice the parameters $\sigma, \omega, \zeta, \eta$ are chosen a priori, to fit empirical data. In the following we will focus on the case $\eta = 0$.

Figure 1 illustrates the versatility of the GMP2 auto-covariance function (Eq.(15)), which includes the previous ones as special cases. For example, for $\zeta = 1$, it degenerates into a pure exponential decay (GMP1, Eq. (13)). On the other extreme, $\zeta = 0$ leads to a pure sinusoidal oscillation.

**Time update**

Now let us turn to the problem of time-updating the state $X(t)$ and covariance $P(t)$ estimates. The general equation reads:

$$\bar{P}_{Xx}(t) = \Phi(t, t_0) P_{xX}(t_0) \Phi^T(t, t_0) + \int_{t_0}^{t} \Phi(t', t_0) G(t') Q(t') \Phi^T(t', t_0) dt',$$  \hfill (20)

$$\dot{\bar{P}}_{Xx}(t) = F(t) P_{xX}(t) + P_{xX}(t) F^T(t) + G(t) Q(t) G^T(t),$$  \hfill (21)

(where $t'$ is just a dummy variable of integration); for a derivation based on stochastic calculus, please refer to\textsuperscript{11} sec. 4.5–4.10, p.147–174). These equations are valid for a continuous dynamic model; we could evaluate them, e.g., integrating numerically between observation epochs. For a version specifically for the discrete case, we obtain\textsuperscript{11} (eq.4–135b, p.172):
where the subscript “d” helps distinguish between the instantaneous in the continuous case from the integrated in the discrete case. The relationship between the two is defined as:

\[ Q_d(t_i) = \int_{t_i}^{t_{i+1}} \Phi(t', t_i) G(t') Q(t') G^T(t') \Phi^T(t', t_i) \, dt'. \]  

(23)

In practice it may be computed as:

\[ Q_d(t_i) = \int_{t_i}^{t_{i+1}} \dot{Q}_d(t', t_i) \, dt', \]  

(24)

\[ \dot{Q}_d(t, t_i) = F(t)Q(t, t_i) + Q(t, t_i) F^T(t) + G(t) Q(t) G^T(t), \]  

(25)

with initial condition \( Q(t, t) = 0 \). In \(^{11}\) (p.172) it is stated that “for time-invariant or slowly varying \( F \) and \( GQG^T \), if the sampling period is short compared to the system’s natural transients, a first order approximation to the solution can often be used:”

\[ \Phi(t_{i+1}, t_i) \approx I - F(t_i) \cdot (t_{i+1} - t_i), \]  

(26)

\[ Q_d(t_{i+1}, t_i) \approx G(t_i) Q(t_i) G^T(t_i) \cdot (t_{i+1} - t_i). \]  

(27)

Again, notice that \( Q_d(t_i) \neq Q(t) \). Notice further that “if a model is derived originally in the first form [continuous], then an ‘equivalent discrete-time model’ can be generated, but with \( G_d(t_i) \) equal to an identity matrix for all times.” \(^{11}\) (p.175); that is distinct from \( G \), still employed to obtain \( Q_d \). In an appendix we provide details about the main matrices and vectors involved.

**EXPERIMENT**

Setup

The method of solution consisted in (1) generating synthetic though noisy observations, for a fictitious satellite and tracking network, followed by (2) the inversion of such observations using a sequential filter. The estimation error is readily obtained via comparison against the known true solution. Range and range-rate observations are corrupted with Gaussian white noise having standard deviation equal to 0.1 m and 0.01 m/s, respectively.

The target satellite is in low Earth orbit (radius \( \sim 7,000 \) km). The tracking network is spaceborne, made of three satellites in geostationary orbit (radius \( \sim 42,000 \) km), equally spaced in longitude. The force model is simplified to include only atmospheric drag and gravity, more specifically point mass, Earth oblateness (\( J_2 \) potential coefficient), and \( J_3 \). Although \( J_3 \) is used in generating the observations, it is not used in the dynamic model for the sequential filter. In other words, \( J_3 \) will be left unmodeled in the deterministic model, and an attempt will be made at modeling it stochastically using GMP. No clock errors were considered.

We will be comparing GMP1 to GMP2 by assessing their performance in terms of errors in estimated satellite position and empirical accelerations (in Earth-centered, inertial coordinates, ECI). While the procedure of calibrating the respective auto-covariance models is described in a latter section, here we introduce some of its results (Figure 2). Aiming at having a fair comparison, the calibration of GMP1 was done as well as possible, within its limitations. More specifically, GMP1 and GMP2 share the same values for \( \sigma \) and \( \omega = 1/T \); beyond that, GMP2 had an optimal \( \zeta \) value, while GMP1 obviously forces \( \zeta \) to unity. (It must be noted that we did not implement a specialized filter separately for GMP1, we simply ran the more general GMP2 with \( \zeta = 1 \); this ensured that no bug exclusive to GMP1 was introduced.)
Results

The comparison was carried out under three distinct tracking regimes. Figure 3 shows acceleration results for a period with good tracking, in which range and range-rate observations were available every 10 seconds throughout the entire three-hour period. Ignoring the first half an hour prior to filter convergence, GMP2 in general follows the true J3 acceleration more closely than GMP1 does, as summarized in the statistics shown in Table 1. GMP2 excels best at the Y coordinate, which reflects the quality of the auto-covariance fit in each of the ECI coordinates (X, Y, and Z, see Figure 2).

Improvements in the accuracy of estimated accelerations are valuable in themselves, especially for gravity mapping missions (e.g., GRACE). They do not carry over directly to improvements in position, though, which although positive are more modest (see Table 1). This can be understood by the fact that the effect on position of an inaccurate acceleration at an earlier epoch can be compensated for with another acceleration at the current epoch. Furthermore, while the Y component showed the greatest acceleration improvement due to GMP2, in position it was the Z component that improved the most. We believe this reflects the observation geometry, as follows: the GEO tracking network is confined to the equatorial plane, thus the target satellite’s out-of-plane position (Z) is more poorly determined than the in-plane components (X and Y). Consequently, Z ends up reaping the most benefit from a tighter stochastic constraint, as provided by GMP2.

A similar assessment was repeated for a poorly observed period, in which tracking is lost 90% of the time (at random intervals). Not surprisingly, RMS values increased (e.g., from 39 cm to 1.45 m, for the GMP1 position error norm RMS). Yet the relative performance (i.e., in percent) of GMP2 with respect to GMP1 remained essentially unaltered (Table 2), for both position and acceleration. It could perhaps be expected that the superiority of GMP2 would be even greater under such poor tracking conditions.

Figure 2: J3 acceleration auto-covariance (in gray) and first- and second-order GMP models (in red and blue, respectively).
Figure 3: J3 acceleration (in gray) and first- and second-order GMP estimates (in red and blue, respectively), for a well observed period.

Figure 4: Positioning error, for first- and second-order GMP (in red and blue, respectively), for an unobserved period (future predictions).
Table 1: Well observed period: RMS of estimation error (with respect to truth). Change is calculated as (RMS2-RMS1)/RMS1. Variables X, Y, Z are the vector components; R is the vector norm. Units are \( \mu m/s^2 \) and m for accelerations and positions, respectively.

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMP1</td>
<td>5.48</td>
<td>1.08</td>
<td>1.82</td>
<td>2.19</td>
</tr>
<tr>
<td>GMP2</td>
<td>0.45</td>
<td>0.50</td>
<td>1.22</td>
<td>1.39</td>
</tr>
<tr>
<td>Change</td>
<td>-17%</td>
<td>-54%</td>
<td>-33%</td>
<td>-36%</td>
</tr>
</tbody>
</table>

Finally, we ran the filters without observations, to assess their performance in making future predictions. The period was the same as in the two previous comparisons. The parameters’ covariance matrix was initialized fully populated (obtained from a previous smoother, forward/backward run). Positioning results (Figure 4) indicate clearly that, for the acceleration at hand (J3) – calibrated as in Figure 2 –, GMP2 position estimates are always more accurate than GMP1. Another way of interpreting Figure 2 is noticing that the time to reach a prescribed position error threshold is much longer for GMP2 than it is for GMP1. For example, GMP1 exceeds 75-cm error in about an hour, threshold which GMP2 takes three hours to reach.

CALIBRATION

The values assigned to the GMP coefficients (\( \sigma, \omega, \zeta \) in Eq. 15) are assumed known during filter runs. Here we address the problem of calibrating such coefficients. The calibration is a three-step procedure:

(a) obtain sample accelerations;
(b) form covariances;
(c) fit auto-covariance model to samples.

This calibration or tuning is usually addressed under the name of “adaptive filtering or estimation.”\(^{15,16,17,18}\) Step (a) is the most difficult one, so let us briefly address the others first.

Step (c), fitting model Eq. (15) to the samples as in Figure 2, can be done in at least three ways: (c.i) manually, or automatically (c.ii) via a non-linear optimization routine or (c.iii) via ad-hoc methods. (c.i) involves trial and error, thus it is admittedly a pedestrian method. But it is undoubtedly feasible and even efficient, in a typical usage scenario. After all, with a single satellite mission of interest, its calibration needs to be performed only once. Furthermore, solutions from the simpler (c.i) may serve as good initial solutions for the more complicated method (c.ii), enabling or at least speeding up convergence in the latter. Thus (c.ii) is especially valuable in refining pre-existing approximate solutions. Although (c.ii) is in principle complicated, its utilization is facilitated by the possibility of treating the model Eq. (15) as a black-box. That way, its partial
derivatives (with respect to the coefficients) are obtained numerically. Although this is more computationally-intensive, it is certainly less tedious and also less error-prone, compared to analytical Jacobians. All other methods fall under the umbrella of (c.iii); these include but are not limited to, e.g., spectral analysis to identify peak frequencies thus \( \omega \), sample standard deviation to obtain \( \sigma \), and finite differencing or curve fitting near the origin to approximate \( \zeta \).

Step (b), forming covariances is trivial given sample accelerations regularly spaced in time. This can be accomplished simply as \( \text{IFFT}(|\text{FFT}(s)| \circ |\text{FFT}(s)|) \), where \( s \) is a vector of sample accelerations, \(|\cdot|\) is the element-wise complex modulus, \( \circ \) is the element-wise product, and FFT/IFFT are the Fast Fourier Transform and its inverse. For better methods, especially more accurate normalizations, please refer to 19. When accelerations are not sampled at regular time intervals, forming the auto-covariance is more laborious. It can be done as follows. Collect the time of each sample in an \( n \times 1 \) vector \( t = [t_k] \). Form the time-lag matrix as

\[
T = \text{abs}([t, t, ..., t]_{n \times n} - [t, t, ..., t]^T_{n \times n}),
\]

which is symmetric, with has zeros in the main diagonal. Construct the corresponding sample covariance matrix \( S = s s^T \), where \( s = [s_k] \) is the sample vector. The so-called sample covariance matrix can be obtained plotting points with \( t_{ij} \) as abscissa and \( s_{ij} \) as ordinate, i.e., made of corresponding elements in the two matrix. Finally, the sample covariance function can then be found binning the cloud by time-lag at regular intervals \( \tau = [\tau_i] \), i.e., \( C_i = \sum q r S_{qr} / N_i \), where \( N_i \) is the number of samples within that bin, and the binning criteria yields \( q, r \) indices such that \( (\tau_{i-1} + \tau_i)/2 < t_{qr} < (\tau_i + \tau_{i+1})/2 \). In either case (regular or irregularly spaced), it is important to have a sufficiently long sample vector, because the auto-covariance behavior at large time-lags is more poorly determined (i.e., there are less numerous pairs of samples separated by larger time lags).

Now returning to the main step, (a), sample accelerations can be obtained as predictions given

Figure 5: J3 acceleration auto-covariance, based on predictive physical model (in black) and on different GMP1-estimated sample accelerations: well-calibrated, backward-smoothed (yellow), well-calibrated, forward-filtered only (green), and poorly-calibrated, backward-smoothed (orange).
by a physical model (over a nominal satellite orbit) or as estimates from a preliminary filter run. Predictive physical models can be, e.g., an expansion of the gravity potential in spherical harmonics, a climatology of atmospheric density, etc. The other option is a hard problem: given tracking data only, it seeks to bootstrap the optimal filter calibration, starting from a sub-optimal one. The filter is ran first with best-guesses about $\sigma, \omega, \zeta$; the resulting preliminary acceleration samples are then fed into (b) and then (c). This three-step calibration procedure is iterated until convergence. In general there is no guarantee that such a bootstrapping will converge to an unbiased solution. Its reliability depends heavily on the sensitivity of the sample accelerations to the coefficient values. Notice that this also involves the quantity and quality of tracking data. In other words, it depends on how well constrained the estimation of empirical accelerations is. For example, sufficiently precise observations could compensate for a loose stochastic constraint in the preliminary filter runs.

To investigate the difficulty in performing the calibration step (a), in Figure 5 we compare the model-predicted J3 acceleration auto-covariance (taken as truth) to different estimated sample auto-covariances. A well-calibrated and well-observed GMP1 (as in the numerical experiments above) performed remarkably well when it was backward-smoothed. When the same GMP1 was forward-filtered only, though, it failed to capture the correct degree of continuity near the origin (its performance remained decent at larger time-lags). Thus backward-smoothing seems mandatory for bootstrapping the GMP calibration. We also tested a backward-smoothed, well-observed, but poorly-calibrated GMP1, meaning one with $\omega$ assuming 10 times its optimal value. Its performance is decidedly better than that of a well-calibrated but forward-filtered only solution, across the whole range of time-lags. Nevertheless the behavior near the origin was again severely degraded. This seems to suggest that the degree of continuity near the origin is hard to infer from tracking data alone; it would be more reliable to constrain it based on a physical justification. On a more encouraging tone, notice that the accuracy demanded from a predictive model is less stringent in the GMP calibration, compared to using the same predictive model as part of the dynamic model. The reason is that, while in the latter usage needs the expected value of acceleration at instantaneous epochs, in the former we only need to know how accelerations remain correlated over increasing time-lags.

CONCLUSIONS

Summary

In precise orbit determination (POD), empirical accelerations typically are modeled with two extreme types of auto-covariance models: pure exponential decay or pure sinusoidal oscillation. Second-order Gauss-Markov processes (GMP2) offer an auto-covariance model with an intermediate degree of continuity, thus enabling a more flexible fit against the observed behavior. It must be noted that improvements in orbit determination due to the utilization of GMP2 are contingent upon the actual variation exhibited by the data at hand. In this sense, our strongest advice is not so much for the indiscriminate utilization of GMP2 in POD, but for the verification and validation of the stochastic model, via inspection of its calibration against sample auto-covariances.

We have demonstrated improvements in POD due to the use of a GMP2 for modeling J3 accelerations that were unaccounted for in the dynamic model. It led to more accurate estimates of satellite position as well as of empirical accelerations, the latter having implications for gravity mapping missions, such as CHAMP, GRACE, and GOCE. Finally, a more adequate stochastic model was shown to extend the time before future predictions exceed a prescribed positioning error threshold.
We also illustrated the calibration of the auto-covariance model, i.e., the procedure by which the coefficients of the auto-covariance model (assumed known during the filter run) are obtained. We described how it can be accomplished following a three-step procedure, and discussed in more depth what we considered its most difficult step, namely, how to obtain samples of the unknown acceleration. We highlighted some of the challenges involved in bootstrapping the GMP calibration based on a preliminary filtering of the tracking data. Although admittedly limited, this latter part of the investigation indicated the importance of backward-smoothing in the preliminary run, and also suggested the need for a physical justification for the degree of continuity near the origin.

**Future work**

One venue of future work is in handling accelerations in the RIC frame (radial, in-track, and cross-track coordinates), instead of ECI as done above. Initial results indicate this to be promising, in the sense that it leads to an easier-to-model sample auto-covariance (see Figure 6). The filter equations will get more complicated, though. For example, it is not immediately self-evident if the transformation of accelerations between ECI and RIC frames can neglect the rate of change in the transformation of positions (as discussed in, e.g., the documentation for NASA’s SPICE system\(^*\)).

Another venue lies in applying a compound GMP. For example, the J3 auto-covariance in the X and Z components could not be perfectly fit using a single GMP2, especially in comparison with the good fit in the Y component (see Figure 2). This misfit can be attributed to the presence of two dominant frequencies in both X and Z. In Figure 7 we demonstrate the possibility of linearly combining two GMP2 for each component. To guarantee observability of such a compound model, the frequencies involved must be sufficiently separate.

A third opportunity exists in applying GMP2 not directly to the whole empirical acceleration itself, but only to certain parameters driving the acceleration. For example, applying it to atmospheric density \(\rho\) in a model of atmospheric drag \(\ddot{\mathbf{r}}_D\):

\[
\ddot{\mathbf{r}}_D = - \frac{1}{2} \frac{A}{m} C_D \rho \|\dot{\mathbf{r}} - \ddot{\mathbf{r}}_{\text{atm}}\| (\dot{\mathbf{r}} - \ddot{\mathbf{r}}_{\text{atm}}),
\]

where \(C_D, A, m,\) and \(\ddot{\mathbf{r}}_{\text{atm}}\) are, respectively, a coefficient of proportionality, satellite cross-section area, satellite mass, and the speed of the parcel of the atmosphere hitting the satellite.

A great deal remains to be done in terms of offering reusable GMP2 calibrations, valid for a given type of empirical acceleration (gravity truncation, atmospheric drag, etc.) and a certain class of satellite orbit. For example, it could be envisioned tabulations of GMP2 coefficients \(\sigma, \omega, \zeta\) versus orbit semi-major axis \(a\), inclination \(i\), etc. for, e.g., atmospheric drag\(^2\).\(^{21}\)

Similar to what was done in \(^5\), it would be instructive to assess the performance of GMP2 in the presence of higher order spherical harmonics of the gravity potential (above J3) as well as solar/lunar perturbations. Such an assessment should compare GMP2 not only to pure exponential GMP1 but also to pure sinusoidal model. Finally, the bootstrapping of GMP2 calibration based on a preliminary filter run should be demonstrated with real tracking data.

**ACKNOWLEDGMENTS**

The first author acknowledges funding provided by Fulbright/Capes.

---

\(^*\) <http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/MATLAB/req/rotation.html#State transformations>
Figure 6: J3 acceleration auto-covariance in RIC coordinates.

Figure 7: J3 acceleration auto-covariance modeled as a compound GMP2 in ECI coordinates.
APPENDIX: BLOCK STRUCTURE

To assist the interested reader in implementing the model above, here we briefly detail the block structure of the main matrices and vectors involved. Let the basic state vector (using transposes to save space on the page) be made of satellite three-dimensional position \( r = [x, y, z]^T \) and velocity \( \dot{r} \) (in Earth-centered, quasi-inertial coordinates, ECI – the scalar \( x \) is not to be confused with the vector \( X \)), as well as \( m \) nuisance parameters (e.g., atmospheric drag coefficients) collected in a vector \( \beta \):

\[
X_b = [r^T, \dot{r}^T, \beta^T]^T.
\]  

The shaping-filter state vector \( X_s \) is made of two vectors, each with three scalars:

\[
X_s = [X_{s1}^T, X_{s2}^T]^T = [X_{s1x}, X_{s1y}, X_{s1z}, X_{s2x}, X_{s2y}, X_{s2z}]^T.
\]  

The state-rate vector corresponding to the augmented state vector, \( X_a = [X_b^T, X_s^T]^T \), follows from Eq. (16) as:

\[
\dot{X}_a = [\ddot{r}^T, \dot{\beta}^T, X_{s2}^T, (-\omega \circ \omega \circ X_{s1} - 2 \zeta \circ \omega \circ X_{s2})^T]^T.
\]  

Notice that there is one different set of scalar GMP parameters \( \sigma, \omega, \zeta \) per coordinate, which make up the respective vectors as in, e.g., \( \zeta = [\zeta_x, \zeta_y, \zeta_z]^T \); in other words, we are using a separate GMP2 for each coordinate. The operation \( \omega \circ \omega = [\omega_x \omega_x, \omega_y \omega_y, \omega_z \omega_z]^T \) is simply the entry-wise product of two vectors. Velocity \( \dot{r} \), necessary in the state-rate \( \dot{X}_a \), is obviously taken directly from the state \( X_s \) itself. Augmented accelerations are obtained as \( \ddot{r} = \ddot{r}_s = \ddot{r}_b + \ddot{r}_s \). The basic accelerations \( \ddot{r}_b \) are those in the dynamic deterministic model (i.e., gravity, etc.). The empirical accelerations are here simply postulated equal to the shaping filter output, \( \ddot{r}_s = X_{s1} \) (although it can be made non-trivial, i.e., the transformation might be more complex than an identity).

The A-matrix (Jacobian of the state-rate with respect to the state) corresponding to the augmented system has the following partitioning:

\[
A_a = \begin{bmatrix}
    A_b & B_b \tilde{H}_s \\
    0_{6 \times (6+p)} & A_s
\end{bmatrix}
\]

\[
= \begin{bmatrix}
    0 & 1 & \frac{\partial \ddot{r}_b}{\partial \beta} & 0 & 0 \\
    \frac{\partial \ddot{r}_b}{\partial r} & \frac{\partial \ddot{r}_b}{\partial \dot{r}} & \frac{\partial \ddot{r}_b}{\partial \beta} & 1 & 0 \\
    \frac{\partial \dot{\beta}}{\partial r} & \frac{\partial \dot{\beta}}{\partial \dot{r}} & \frac{\partial \dot{\beta}}{\partial \beta} & 0 & 0 \\
    0 & 0 & \text{diag}(-\omega \circ \omega) & \text{diag}(-2 \zeta \circ \omega) & \end{bmatrix}
\]  

The unadorned identity \( I \) and null matrices have dimensions \( 3 \)-by-\( 3 \). Notice that \( \tilde{H}_s(t) = [I, 0] \) follows from \( X_{s1} \) being the shaping filter output, and \( B_b = [0, I = (\partial \ddot{r}_s/\partial X_{s1})^T, 0_{p \times 3}] \) follows from \( X_{s1} \) being an acceleration in the same coordinate basis as \( \ddot{r}_s \). If that was not postulated so, it could give rise to a second term in \( \partial \ddot{r}_s/\partial r = \partial \ddot{r}_b/\partial r + \partial \ddot{r}_s/\partial r \) (and similarly in \( \partial \ddot{r}_b/\partial r \)). In that case the upper-right portion of \( A_a \) would no longer be simply \( A_b \) (although we would still remain with \( \partial X_{s1}/\partial r = \partial X_{s1}/\partial \dot{r} = 0 \) in the lower-left of \( A_a \)).
The diagonal matrices in Eq. (32) read, e.g.,

\[
\text{diag}(-\omega \ast \omega) = \begin{bmatrix}
-\omega_x^2 & 0 & 0 \\
0 & -\omega_y^2 & 0 \\
0 & 0 & -\omega_z^2
\end{bmatrix},
\]

where 0 in Eq.(33) is the scalar zero, not the 3-by-3 null matrix 0 used in Eq. (32). The H-matrix (Jacobian of observations with respect to state) is simply:

\[
\bar{H}_a = [\bar{H}_b \quad 0_{m \times 6}],
\]

i.e., none of the \( m \) range and range-rate scalar observations depend on the empirical accelerations. The B-matrix (Jacobian of the state-rate with respect to white noise) reads:

\[
B_a = \begin{bmatrix}
0_{(6+p) \times 3} \\
B_s
\end{bmatrix} = \begin{bmatrix}
0_{(6+p) \times 3} \\
\text{diag}(a) \\
\text{diag}(b)
\end{bmatrix}.
\]

The scalar components in the vector \( a = [a_x, a_y, a_z]^T \) are calculated as per Eq. (17), and similarly for the vector \( b \), per Eq. (18).

The \( \Phi \)-matrix (state transition matrix, the Jacobian of current state with respect to previous state) is obtained via numerical integration of \( A_a \). The interested reader may find analytical expressions for the GMP contribution in Bryson & Ho (1975). In any case, usually this integration includes the state-rate vector, to obtain the time-updated state (corresponding to a reference trajectory). When solving the initial value problems for these ordinary differential equations using a step-adaptive algorithm (e.g., Runge-Kutta), a tolerance must be prescribed. It is not immediately obvious what would be a reasonable tolerance value for each element. We found it adequate to set the tolerance only for the elements corresponding to the spacecraft position coordinates, and a very large value as tolerance for all other elements. It is easier to come up with reasonable tolerance values for elements having more intuitive units, such as length in the case of position. This avoids having an unnecessarily long computation time because of an otherwise too stringent tolerance. Notice that the integration accuracy of the remaining elements will be commensurate with that experienced by spacecraft position, since their integration is performed simultaneously. If a relative tolerance is needed, a conservative value may be obtained dividing the absolute tolerance (say, 1 mm) by the orbit nominal semi-major axis dimension.

The time-updated state covariance matrix has two terms: \( \bar{P} = \bar{P}' + \bar{P}'' \); the first one is the familiar \( \bar{P}' = \Phi P \Phi^T \); the second term reads:

\[
\bar{P}'' = G_d Q_d G_d^T = G_d \left( G Q G^T \cdot (t_{i+1} - t_i) \right) G_d^T.
\]

Applying the simplifications discussed in the main text, \( G_d = I, Q = I \), and noting further that \( G = B_a = \partial X_a / \partial w \), we obtain, finally:

\[
\bar{P}'' = \begin{bmatrix}
0_{(6+p) \times (6+p)} & 0_{(6+p) \times 3} & 0_{(6+p) \times 3} \\
0_{3 \times (6+p)} & \text{diag}(a \ast a) & \text{diag}(a \ast b) \\
0_{3 \times (6+p)} & \text{diag}(b \ast a) & \text{diag}(b \ast b)
\end{bmatrix} : |t_{i+1} - t_i|,
\]

where the absolute function surrounding the time-lag is necessary when running the filter backwards, as a smoother. The measurement update is just as in the standard sequential formulation.
REFERENCES


