AAS 13-762

EFFICIENT COVARIANCE INTERPOLATION USING BLENDING OF APPROXIMATE STATE ERROR TRANSITIONS

Sergei Tanygin[*]

Efficient storage and quick access to covariance data are important aspects of orbit catalog maintenance and conjunction analysis (CA). The catalog and CA access and storage requirements cannot accommodate running a complete estimation process whenever orbit state and covariance are requested at some time. Instead, ephemeris and reduced covariance data are recorded at discrete times. Covariance interpolation from tabulated data must preserve positive definiteness and evolve covariance similar to the estimation process. This paper describes a new covariance interpolation method which blends approximate state error transitions anchored at end points of interpolation interval to produce accurate physically meaningful covariance.

INTRODUCTION

Efficient storage of and quick access to covariance data are important aspects of orbit catalog maintenance and conjunction analysis (CA).[1] Covariance generation is part of an estimation process that may include different state variables, dynamical models, measurement models, and estimation algorithms. The estimation process is typically computationally expensive and requires significant storage for covariance data for large estimation states. The catalog and CA access and storage requirements are generally too stringent to accommodate running of full estimation processes whenever orbit state and covariance are requested for some moment in time. Instead, ephemeris and reduced covariance data are tabulated and stored at certain times. There are many established techniques for computing an orbit state given tabulated ephemeris; they lie outside of the scope of this paper which focuses on computing covariance from tabulated data. This task faces unique challenges because, unlike elements of Cartesian position and velocity, covariance elements cannot be interpolated independently. They must be interpolated in a way that at the very least preserves the positive definiteness of covariance matrix. Moreover, it is desirable to generate interpolated covariance in a way that follows closely a physically meaningful evolution that would have been generated by the underlying estimation process.

Covariance interpolation solution methods can be broadly separated into two categories: those that use state error transition and those that do not. In general, complete state error transition and process noise employed by the estimation process are not available during interpolation, so only approximate state error transitions can be used. Alfano[2] proposed using quintic splines to fit between two tabulated covariances with their first and second derivatives approximated using some representative force model, e.g. a simple two-body model. It can be argued that through the use of

[*] Senior Astrodynamics Specialist, Analytical Graphics, Inc., 220 Valley Creek Blvd., Exton, PA 19341.

the representative force model some approximation of the orbit state error transition is indirectly introduced into the computation. However, in the end the quintic spline interpolation technique is not constrained to adhere to the realistic evolution of covariance. For example, it fundamentally does not guarantee positive definiteness of interpolated covariance. Woodburn and Tanygin[3] proposed a different method specifically for interpolating of position-only tabulated covariance. This method does not incorporate state error transition information but relies on the eigen-decomposition of a 3x3 covariance matrix to separately interpolate covariance sigma values and orthogonal eigenvector (rotation) matrices. The interpolated position covariance is reassembled from the interpolated sigma values and rotation matrix with the added benefit of guaranteeing positive definiteness. This method is restricted to state representations where position can be separated.

This paper describes a new method for interpolating covariance in which an approximate state error transition is incorporated directly into the computation. This method guarantees positive definiteness of the interpolated covariance and can be applied either to the original form of the covariance matrix or to one of its several possible factorizations. In all cases, incorporating approximate state error transitions brings evolution of the interpolated covariance closer to what would have been generated by the estimation process.

PROBLEM STATEMENT

Let two covariances $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ corresponding to states $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ and computed at times t_i and t_{i+1} be related via a time update of the nonlinear estimation process[4]

$$\mathbf{P}_{i+1} = \mathbf{\Phi}_i(t_{i+1})\mathbf{P}_i\mathbf{\Phi}_i(t_{i+1})^{\mathrm{T}} + \mathbf{Q}_{i+1}\,,\ i = 0,1,... \tag{1}$$

Here $\mathbf{\Phi}_i(t) \triangleq \partial\mathbf{x}(t)/\partial\mathbf{x}_i$ denotes the state error transition matrix evaluated along the state trajectory $\mathbf{x}(t)$ and anchored at time t_i, i.e. $\mathbf{\Phi}_i(t_i) = \mathbf{I}$; also, $\mathbf{Q}_{i+1}$ denotes the process noise term accumulated from time t_i to t_{i+1}. In theory, the covariance $\mathbf{P}(t)$ for any time $t \in [t_i, t_{i+1}]$ can be evaluated using this formula provided that the state error transition matrix and the process noise term can be computed for that time. In practice, using tabulated covariances to restart the full estimation process and perform a large number of covariance evaluations at intermediate times can be too slow and require too much memory. The computational cost arises from employing accurate but computationally expensive dynamical models and the memory cost arises from using states and covariances that may include auxiliary modeling parameters required by the estimation process but generally not needed for the intended analysis, e.g. force modeling parameters not needed for conjunction analysis.

The goal of this paper is to provide an efficient alternative for computing the relevant parts of $\mathbf{P}(t)$ based on $\mathbf{x}(t)$ and tabulated $\mathbf{P}_i$, $\mathbf{P}_{i+1}$.

SOLUTION OUTLINE

Throughout the rest of the paper it should be understood that, different from the previous section, $\mathbf{P}_i$, $\mathbf{P}_{i+1}$ and $\mathbf{P}(t)$ from now on refer only to those parts of the full covariances employed by the estimation process (Eq. (1)) that are being tabulated and interpolated; that no knowledge of the process noise is assumed; and that from this point on the state error transition matrices $\mathbf{\Phi}_i(t)$ are approximate and based only on the information derivable from $\mathbf{x}(t)$, the tabulated part of the state, and possibly some limited additional information needed for the simplified dynamical model.

Let $\mathbf{P}_{(i)}(t)$ denote the state error covariance matrix created without the process noise by using the approximate $\mathbf{\Phi}_i(t)$ to propagate $\mathbf{P}_i$ from t_i to t. Then, in this notation,[*] $\mathbf{P}_{(i)} = \mathbf{\Phi}_i \mathbf{P}_i \mathbf{\Phi}_i^{\mathrm{T}}$. For the same time $t \in \left[t_i, t_{i+1}\right]$, an alternative version of the covariance matrix can be created by propagating $\mathbf{P}_{i+1}$ from t_{i+1} back to t using $\mathbf{\Phi}_{i+1}(t)$ anchored at t_{i+1}: $\mathbf{P}_{(i+1)} = \mathbf{\Phi}_{i+1} \mathbf{P}_{i+1} \mathbf{\Phi}_{i+1}^{\mathrm{T}}$. Since these formulations lack the process noise terms and use only approximate (and potentially reduced order) $\mathbf{\Phi}_i(t)$ and $\mathbf{\Phi}_{i+1}(t)$, they in general do not match how tabulated $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ have been originally related to each other by the estimation process. Hence, although computed for the same time t, in general, $\mathbf{P}_{(i)} \neq \mathbf{P}_{(i+1)} \neq \mathbf{P}$.

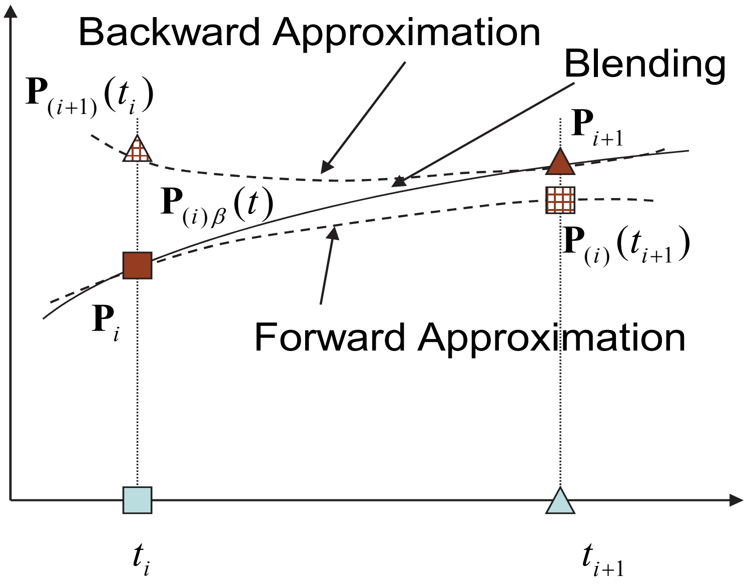

Figure 1. Smooth Interpolation using Blending of Forward and Backward Approximations.

In other words, the approximate transition that $\mathbf{\Phi}_i(t)$ imposes $\mathbf{P}_i$ moving forward in time does not match the transition that $\mathbf{\Phi}_{i+1}(t)$ imposes on $\mathbf{P}_{i+1}$ moving backward in time, and neither matches the "true" covariance $\mathbf{P}(t)$ that would have been obtained by the estimation process. These alternative approximations cover the same interval $t \in \left[t_i, t_{i+1}\right]$. The forward approximation, clearly accurate at time t_i, should remain more accurate than the backward approximation near t_i. On the other hand, the backward approximation, clearly accurate at time t_{i+1}, should remain more accurate than the forward approximation near t_{i+1}. Hence, a reasonable combined ap-

[*] Throughout the paper the time argument t is often omitted for brevity.

proximation should follow *mostly* $\mathbf{P}_{(i)}$ governed by $\mathbf{\Phi}_i(t)$ at the beginning of the interpolation interval and then progressively favor the other until it follows *mostly* $\mathbf{P}_{(i+1)}$ governed by $\mathbf{\Phi}_{i+1}(t)$ at the end of that interval (Fig. 1).

This type of approach called *blending* represents a very general concept. Any trajectories, motions or properties can be blended using specially designed scalar functions $\beta(\tau)$, where τ is the fractional portion of the interpolation interval: $\tau \triangleq (t - t_i)/(t_{i+1} - t_i) \in [0,1]$.

BLENDING VS. STANDARD INTERPOLATION

Let the covariance obtained by applying the blending function β to the forward and backward approximations, $\mathbf{P}_{(i)}$ and $\mathbf{P}_{(i+1)}$, be formally written as $\mathbf{P}_{(i)\beta} = \underset{\beta}{\text{blending}}\left\{\mathbf{P}_{(i)}, \mathbf{P}_{(i+1)}\right\}$.

In this notation a standard interpolation can be written as $\mathbf{P}_{i\alpha} = \underset{\alpha_0,\alpha_1}{\text{interpolation}}\left\{\mathbf{P}_i, \mathbf{P}_{i+1}\right\}$ where α_0, α_1 are the interpolation basis functions. In order to possess the interpolation property of passing through the tabulated nodes $\mathbf{P}_i$, $\mathbf{P}_{i+1}$ the blending function must satisfy

$$\beta(0) = 0,\ \beta(1) = 1 \tag{2}$$

and the interpolation basis functions must satisfy[*]

$$\alpha_0(0) = 1,\ \alpha_0(1) = 0 \text{ and } \alpha_1(0) = 0,\ \alpha_1(1) = 1 \tag{3}$$

For a smooth blending, the blending function must also satisfy

$$\beta'(0) = \beta'(1) = 0 \tag{4}$$

where β' denotes differentiation with respect to τ. For a smooth interpolation, a new formulation must be used $\mathbf{P}_{i\alpha} = \underset{\alpha_0,\alpha_1,\bar{\alpha}_0,\bar{\alpha}_1,}{\text{smooth interpolation}}\left\{\mathbf{P}_i, \mathbf{P}_{i+1}, \dot{\mathbf{P}}_i, \dot{\mathbf{P}}_{i+1}\right\}$ where the two additional basis functions $\bar{\alpha}_0$, $\bar{\alpha}_1$ operate on the time derivatives $\dot{\mathbf{P}}_i$, $\dot{\mathbf{P}}_{i+1}$ which are provided at times t_i and t_{i+1}, respectively, in addition to $\mathbf{P}_i$, $\mathbf{P}_{i+1}$. Additional conditions in this case include[†]

$$\alpha_k'(0) = \alpha_k'(1) = \bar{\alpha}_k(0) = \bar{\alpha}_k(1) = 0 \text{ for } k = 0,1 \tag{5}$$

and

$$\bar{\alpha}_0'(0) = 1,\ \bar{\alpha}_0'(1) = 0 \text{ and } \bar{\alpha}_1'(0) = 0,\ \bar{\alpha}_1'(1) = 1 \tag{6}$$

[*] In many cases, the basis functions are such that $\alpha_0 = 1 - \alpha_1$ and the blending function β can be identical to α_1.

[†] Higher order derivatives can be considered for even smoother interpolations in which case additional basis functions are used and additional conditions are imposed at the ends of interpolation interval.

Note the apparent similarities in a way that the blending function β and the interpolation basis functions α_0, α_1 effectively weigh the relative contributions from the two constituents while satisfying certain boundary conditions. However, a very important distinction between the approaches lies in the nature of the constituents being blended vs. interpolated. Blending occurs between values *propagated to the same time* from two tabulated times. Interpolation occurs solely based on fitting of the basis functions between tabulated times. Thus, while interpolation may incorporate higher order derivatives *at the tabulated times*, it cannot accommodate any information about possible transition *between these times* and relies instead on non-physical mathematical fitting techniques to produce interpolated time histories. The main advantage of the proposed blending approach lies in the ease with which it can incorporate approximate yet physically meaningful transitions.

Blending Functions

There are many types of blending functions but polynomial functions are of particular interest because of their simplicity. Some of them are listed in Table 1.

Table 1. Polynomial Blending Functions.

Order	Expression $\beta(\tau)$
Linear	$\beta(\tau) = \tau$
Quadratic	$\beta(\tau) = \begin{cases} 2\tau^2, \tau \in [0, 0.5] \\ 4\tau - 2\tau^2 - 1, \tau \in [0.5, 1] \end{cases}$
Cubic	$\beta(\tau) = 3\tau^2 - 2\tau^3$
Quintic	$\beta(\tau) = 10\tau^3 - 15\tau^4 + 6\tau^5$

Here all functions except linear also satisfy the boundary conditions for smooth blending.

BLENDING OF SYMMETRIC POSITIVE DEFINITE MATRICES

The simplest application of blending to symmetric positive definite matrices is called *arithmetic* because it is related to a concept of arithmetic mean.

Arithmetic Blending[*]

The arithmetic blending is derived in the Euclidean space where the distance metric between two real square matrices is defined as the Frobenius norm of their difference:[5] $d_A(\mathbf{P}_{(i)}, \mathbf{P}_{(i+1)}) \triangleq \left\| \mathbf{P}_{(i)} - \mathbf{P}_{(i+1)} \right\|_F$. The blending in this case is trivial:

[*] An alternative to arithmetic blending is called *geometric* or *Riemannian* blending. It is more involved but has a conceptual advantage of traversing a geodesic on the manifold of symmetric positive definite matrices. The metric underlying this approach is related to the geometric mean and is often used in studies of positive numbers, positive integrable functions, and positive definite operators.[5] This type of blending is not included in the main body of the paper but can be found in the Appendix because, despite its conceptual advantage and interesting mathematics, it performs poorly for orbit covariances.

$$\mathbf{P}_{(i)\beta} = \mathbf{P}_{(i)}\left[1-\beta\right] + \mathbf{P}_{(i+1)}\beta , \tag{7}$$

and so is its rate of change:

$$\dot{\mathbf{P}}_{(i)\beta} = \dot{\mathbf{P}}_{(i)}\left[1-\beta\right] + \dot{\mathbf{P}}_{(i+1)}\beta + \left[\mathbf{P}_{(i+1)} - \mathbf{P}_{(i)}\right]\beta' / \left(t_{i+1} - t_i\right). \tag{8}$$

Solution Properties

The following properties are exhibited by the arithmetic blending.[*]

Interpolation: it trivially follows from Eqs. (2, 7) that $\mathbf{P}_{(i)\beta}(t_i) = \mathbf{P}_i$ and $\mathbf{P}_{(i)\beta}(t_{i+1}) = \mathbf{P}_{i+1}$.

Continuity and Smoothness: if tabulated covariances are obtained by a smoother then the resulting blended covariances will satisfy continuity and smoothness conditions across abutting blending intervals. The continuity follows directly from the interpolation property above. The smoothness can be demonstrated by examining the blended rates at the recorded times. Let $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t),t)$ be the adopted approximate dynamical model and $\mathbf{F}(\mathbf{x}(t),t) = \partial \mathbf{f}(\mathbf{x}(t),t) / \partial \mathbf{x}$ be its Jacobian evaluated along $\mathbf{x}(t)$. Then from Eq. (8) and using the boundary conditions imposed on $\beta(\tau)$ by Eqs. (2, 4), it follows that at t_i

$$\dot{\mathbf{P}}_{(i)\beta}(t_i) = \dot{\mathbf{P}}_{(i)}(t_i) = \mathbf{F}_i\mathbf{P}_i + \mathbf{P}_i\mathbf{F}_i^T \tag{9}$$

where $\mathbf{F}_i \triangleq \mathbf{F}(\mathbf{x}_i, t_i)$ is the Jacobian evaluated at time t_i at the known state $\mathbf{x}_i$. An entirely analogous derivation can be carried out at time t_{i+1}. Their comparison then reveals that at any recorded time t_{i+1}, $\dot{\mathbf{P}}_{(i)\beta}(t_{i+1}) = \dot{\mathbf{P}}_{(i+1)\beta}(t_{i+1})$, i.e. the blended rates computed for that time from the preceding and succeeding intervals are the same.

Symmetric Positive Definiteness: this property requires the blending function to remain strictly between 0 and 1 for $0 < \tau < 1$. Then, both $\beta(\tau)$ and $1-\beta(\tau)$ remain positive for $0 < \tau < 1$, which in turn guarantees that $\mathbf{P}_{(i)\beta}(t)$ is symmetric positive definite throughout $t \in \left[t_i, t_{i+1}\right]$ based on Eq.(7).

Commutativity with Coordinate Transformation: let any two coordinate representations of the state $\mathbf{x}$ and $\tilde{\mathbf{x}}$ be related via $\tilde{\mathbf{x}} = \boldsymbol{\lambda}(\mathbf{x},t)$ with the Jacobian $\boldsymbol{\Lambda}(\mathbf{x},t) \triangleq \partial\tilde{\mathbf{x}}(t) / \partial\mathbf{x}(t)$. Then, for covariances $\tilde{\mathbf{P}}$ defined in coordinates of $\tilde{\mathbf{x}}$ instead of $\mathbf{x}$, it follows that the arithmetic blending based on Eq.(7) yields[†]

[*] The same properties are also exhibited by the geometric blending but that lies outside of scope of this paper.

[†] It is assumed here that if state representations at some intermediate time t are obtained by interpolation, then these representations $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are still related by $\tilde{\mathbf{x}} = \boldsymbol{\lambda}(\mathbf{x},t)$. In practice, due to nonlinearities, some differences in the interpolated states may occur causing in turn small differences in the corresponding covariance interpolations. Also, note that the Jacobian transformation $\boldsymbol{\Lambda} \triangleq \boldsymbol{\Lambda}(\mathbf{x},t)$ at time t is based on the actual state $\mathbf{x}$ not the approximate state implied by the approximate state error transitions.

$$\begin{aligned}\tilde{\mathbf{P}}_{(i)\beta} &= \tilde{\mathbf{P}}_{(i)}\left[1-\beta\right]+\tilde{\mathbf{P}}_{(i+1)}\beta = \mathbf{\Lambda}\mathbf{P}_{(i)}\mathbf{\Lambda}^T\left[1-\beta\right]+\mathbf{\Lambda}\mathbf{P}_{(i+1)}\mathbf{\Lambda}^T\beta \\ &= \mathbf{\Lambda}\left(\mathbf{P}_{(i)}\left[1-\beta\right]+\mathbf{P}_{(i+1)}\beta\right)\mathbf{\Lambda}^T = \mathbf{\Lambda}\mathbf{P}_{(i)\beta}\mathbf{\Lambda}^T\end{aligned} \tag{10}$$

which demonstrates that the coordinate change $\left(\mathbf{x},\mathbf{P}\right)\rightarrow\left(\tilde{\mathbf{x}},\tilde{\mathbf{P}}\right)$ affects tabulated and blended covariances in the same way. In other words, whether starting with a coordinate transformation or starting with the proposed blending, the combined result is the same.

Invariance under Coordinate Transformation: it follows from the commutativity property that converting first to a different coordinate representation, then blending and converting back produces the same result as applying that blending directly in the original coordinates.

FACTORIZATIONS

The blending approach is not limited to whole matrices. It can be also applied to various matrix factorizations from which whole matrices can be reassembled. This may be advantageous if the blending of the factorized constituents, e.g. singular values, is better understood than the blending of whole matrices.

Square Root

The square root factorization uses the Cholesky decomposition to obtain the lower triangular square root matrix $\mathbf{L}$ such that $\mathbf{P}=\mathbf{L}^T\mathbf{L}$.[6] The blending in this case resembles the arithmetic blending of the whole matrix $\mathbf{P}$ but uses $\mathbf{L}$ instead:

$$\mathbf{L}_{(i)\beta} = \mathbf{L}_{(i)}\left[1-\beta\right]+\mathbf{L}_{(i+1)}\beta\,. \tag{11}$$

The constituent square root matrices are factorized from the constituent covariance matrices and the resulting covariance matrix is reassembled from the blended square root matrix.

Sigma Correlation

The sigma correlation factorization transforms the covariance matrix into a matrix with sigma values on its diagonal and correlation coefficients everywhere else. The resulting matrix is still positive definite and symmetric and, thus, can be blended using the same method.

Complete Eigen-Decomposition

The eigen-decomposition of a positive definite matrix produces a diagonal positive eigenvalue matrix and an orthogonal eigenvector matrix which can (in theory) be blended or interpolated separately. The first challenge of using this factorization is determining which eigenvalues from the constituent covariances should be paired up for blending (or interpolation). Woodburn and Tanygin[3] advocate sorting both sets of eigenvalues by size and then pairing them up accordingly. The next challenge is determining the nearest rotation, i.e. determining the orthogonal matrix that rotates the eigenvector matrix of one covariance into the eigenvector matrix of the other covariance along the shortest geodesic arc in $SO(n)$. Once the arc is found, the blending (or interpolation) can occur along the selected geodesic. After blending or interpolating the square roots of eigenvalues and rotating the eigenvector matrices, the resulting covariance can be reconstructed. For the three-dimensional case ($n=3$) the problem is solved in Reference 3 but for higher-dimensional cases the problem is significantly more challenging. Formally, it is possible to write for any dimension

$$\mathbf{\Sigma}_{(i)\beta} = \mathbf{\Sigma}_{(i)}\left[1-\beta\right] + \mathbf{\Sigma}_{(i+1)}\beta \tag{12}$$

and

$$\begin{aligned}\mathbf{U}_{(i)\beta} &= \mathbf{U}_{(i)}\,\mathrm{Exp}(\beta\,\mathrm{Log}(\mathbf{U}_{(i)}^{T}\mathbf{U}_{(i+1)})) = \mathrm{Exp}(\left[1-\beta\right]\mathrm{Log}(\mathbf{U}_{(i)}\mathbf{U}_{(i+1)}^{T}))\mathbf{U}_{(i+1)} \\ &= \mathbf{U}_{(i)}\left(\mathbf{U}_{(i)}^{T}\mathbf{U}_{(i+1)}\right)^{\beta} = \left(\mathbf{U}_{(i)}\mathbf{U}_{(i+1)}^{T}\right)^{1-\beta}\mathbf{U}_{(i+1)}\end{aligned}, \tag{13}$$

where the diagonal positive definite matrix $\mathbf{\Sigma}$ and orthogonal matrix $\mathbf{U} \in SO(n)$ are defined from the eigen-decomposition $\mathbf{P} = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$.[7] It is assumed that $\mathbf{\Sigma}_{(i)}$, $\mathbf{\Sigma}_{(i+1)}$ are properly sorted and that the nearest rotation between $\mathbf{U}_{(i)}$ and $\mathbf{U}_{(i+1)}$ is determined.[*] Note that, even if these assumptions are met, raising higher-dimensional orthogonal matrices to a fractional power is still a numerically challenging procedure.[8]

Block Decomposition (Cartesian Position-Velocity)

The challenges of higher-dimensional eigen-decomposition can be alleviated in cases when covariance matrices are represented using Cartesian position and velocity. In these cases it is reasonable to replace a complete six-dimensional eigen-decomposition with two independent three-dimensional eigen-decompositions: one for the position portion and the other for the velocity portion. Each can be performed as described in Reference 3. It is then also possible to blend the position-velocity cross-correlation terms by applying sigma correlation method to the already block-diagonalized matrices. Let covariance matrix using Cartesian position and velocity coordinates be represented as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_p & \mathbf{W}^T \\ \mathbf{W} & \mathbf{P}_v \end{bmatrix}. \tag{14}$$

Then the blended cross-correlation terms can be obtained using

$$\mathbf{W}_{(i)\beta} = \mathbf{U}_{v(i)\beta}\mathbf{\Sigma}_{v(i)\beta}\begin{pmatrix}\left[1-\beta\right]\mathbf{\Sigma}_{v(i)}^{-1}\mathbf{U}_{v(i)}^{T}\mathbf{W}_{(i)}\mathbf{U}_{p(i)}\mathbf{\Sigma}_{p(i)}^{-1} \\ +\beta\mathbf{\Sigma}_{v(i+1)}^{-1}\mathbf{U}_{v(i+1)}^{T}\mathbf{W}_{(i+1)}\mathbf{U}_{p(i+1)}\mathbf{\Sigma}_{p(i+1)}^{-1}\end{pmatrix}\mathbf{\Sigma}_{p(i)\beta}\mathbf{U}_{p(i+1)\beta}^{T} \tag{15}$$

where subscripts "*p*" and "*v*" correspond to the position and velocity related terms, respectively.

CHOICE OF COORDINATES

As indicated previously, the choice of covariance coordinates can affect performance of certain blending methods. The blending of whole covariance matrices is unaffected by the coordinate choice (see *invariance under coordinate transformation* property). Other methods that do involve factorizations, such as the square root, sigma correlation and eigen-decomposition methods, are affected by the coordinate choice.[†]

[*] Note that this method and its three-dimensional specialization described in Reference 3 use arithmetic blending for singular values and geometric blending for orthogonal eigenvector matrices.

[†] The block decomposition method is by its very nature only applicable to certain coordinates.

Among the many different coordinate representations of the orbit state, the primary distinction important for this investigation is how quickly the first order perturbations in coordinates change over time. From this perspective, two representations, one using inertial Cartesian position and velocity, and the other using Keplerian elements, provide two characteristic examples: the former has perturbations in all coordinate changing relatively quickly, the latter has all but one of those perturbations changing relatively slowly. These effects on blending and interpolation are quantified using numerical examples later in the paper.

Invariance of Whole Matrix Blending vs. Non-Invariance of Whole Matrix Interpolation

The important distinction between blending and interpolation in different coordinates can be illustrated using the following simple example. Consider the simplest (linear) arithmetic blending

$$\mathbf{P}_{(i)linear} = \left[1-\tau\right]\mathbf{P}_{(i)} + \tau\mathbf{P}_{(i+1)} \tag{16}$$

for which the corresponding linear interpolation takes on a very similar form

$$\mathbf{P}_{ilinear} = \left[1-\tau\right]\mathbf{P}_{i} + \tau\mathbf{P}_{i+1} \,. \tag{17}$$

The key difference is that $\mathbf{P}_{(i)}$ and $\mathbf{P}_{(i+1)}$ used for blending are propagated from $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ to the same time t. This is why blending performance is invariant with respect to coordinate changes

$$\begin{aligned}\tilde{\mathbf{P}}_{(i)linear} &= \left[1-\tau\right]\tilde{\mathbf{P}}_{(i)} + \tau\tilde{\mathbf{P}}_{(i+1)} = \left[1-\tau\right]\mathbf{\Lambda}\mathbf{P}_{(i)}\mathbf{\Lambda}^T + \tau\mathbf{\Lambda}\mathbf{P}_{(i+1)}\mathbf{\Lambda}^T \\ &= \mathbf{\Lambda}\left(\left[1-\tau\right]\mathbf{P}_{(i)} + \tau\mathbf{P}_{(i+1)}\right)\mathbf{\Lambda}^T = \mathbf{\Lambda}\mathbf{P}_{(i)linear}\mathbf{\Lambda}^T\end{aligned} \tag{18}$$

whereas interpolation performance

$$\tilde{\mathbf{P}}_{ilinear} = \left[1-\tau\right]\mathbf{\Lambda}_i\mathbf{P}_i\mathbf{\Lambda}_i^T + \tau\mathbf{\Lambda}_{i+1}\mathbf{P}_{i+1}\mathbf{\Lambda}_{i+1}^T \neq \mathbf{\Lambda}\mathbf{P}_{ilinear}\mathbf{\Lambda}^T \tag{19}$$

is generally affected by them. Here, $\mathbf{\Lambda} \triangleq \mathbf{\Lambda}(\mathbf{x},t)$ and $\mathbf{\Lambda}_i \triangleq \mathbf{\Lambda}(\mathbf{x}_i,t_i)$. Note that, even if the error transitions are reduced to trivial $\mathbf{\Phi}_i(t) = \mathbf{\Phi}_{i+1}(t) = \mathbf{I}$, the distinction between blending and interpolation remains valid: from the blending perspective the covariances $\mathbf{P}_{(i)}$ and $\mathbf{P}_{(i+1)}$, although now constant and identical to $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$, are still *propagated* to the same time t. Hence, both $\mathbf{P}_{(i)}$ and $\mathbf{P}_{(i+1)}$ should be transformed to new coordinates using the same $\mathbf{\Lambda}$. This is different from what is done during interpolation where $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ are associated with their own times t_i and t_{i+1}, and are, therefore, transformed using two different $\mathbf{\Lambda}_i$ and $\mathbf{\Lambda}_{i+1}$, respectively.

APPLICATION TO ORBIT COVARIANCE

Consider the advantages of the proposed blending method over classical interpolation when applied to orbit covariance.

Arguably, the simplest case for which both approaches should work perfectly is when covariance propagation is carried out along an exact two-body orbit and is parameterized in Keplerian elements. The state error transition matrix in this case is constructed as an identity matrix plus a single non-zero off-diagonal element[9]

$$\{\Phi_i\}_{Ma} = m(t-t_i) \text{ with } m = -(3/2)(n/a) = \text{const} \tag{20}$$

where $\{\ \}_{rs}$ denotes the rs element of a matrix, M denotes the mean anomaly, n denotes the mean motion, and a denotes the semi-major axis. The Jacobian of the two-body force model in these coordinates is constant with all elements equal to zero except for a single off-diagonal element

$$\{\mathbf{J}\}_{\dot{M}a} \triangleq \partial \dot{M} / \partial a = m \tag{21}$$

Thus, covariance propagation from $\mathbf{P}_i$ at time t_i to $\mathbf{P}$ at time t can be described in a closed polynomial form as

$$\{\mathbf{P}\}_{rs} = \{\mathbf{P}_i\}_{rs} \text{ for } r \neq M \text{ , } s \neq M \text{ ,} \tag{22}$$

$$\{\mathbf{P}\}_{Ms} = \{\mathbf{P}_i\}_{Ms} + \{\mathbf{P}_i\}_{as}\, m(t-t_i) \text{ for } s \neq M \text{ ,} \tag{23}$$

and

$$\{\mathbf{P}\}_{MM} = \{\mathbf{P}_i\}_{MM} + 2\{\mathbf{P}_i\}_{aM}\, m(t-t_i) + \{\mathbf{P}_i\}_{aa}\, m^2 (t-t_i)^2 \tag{24}$$

Note that, since the Jacobian $\mathbf{J}$ is nilpotent and constant, covariance derivatives are trivial to compute:

$$\dot{\mathbf{P}} = \mathbf{J}\mathbf{P} + \mathbf{P}\mathbf{J}^T \text{ , } \ddot{\mathbf{P}} = 2\mathbf{J}\mathbf{P}\mathbf{J}^T \text{ and } \mathbf{P}^{(n)} = \mathbf{0} \text{ for } n > 2 \tag{25}$$

Specifically, the first derivatives are

$$\{\dot{\mathbf{P}}\}_{rs} = \mathbf{0} \text{ for } r \neq M \text{ , } s \neq M \text{ ,} \tag{26}$$

$$\{\dot{\mathbf{P}}\}_{Ms} = \{\mathbf{P}\}_{as}\, m \text{ for } s \neq M \text{ and } \{\dot{\mathbf{P}}\}_{MM} = 2\{\mathbf{P}\}_{aM}\, m \text{,} \tag{27}$$

and the second derivatives are

$$\{\ddot{\mathbf{P}}\}_{rs} = \mathbf{0} \text{ except for } \{\ddot{\mathbf{P}}\}_{MM} = 2\{\mathbf{P}\}_{aa}\, m^2 \tag{28}$$

The above derivations are meant to demonstrate that polynomial basis functions should have no problem accurately interpolating orbit covariance propagated using a two-body force model. Indeed, even linear interpolation may be adequate over short interpolation intervals since having $m \ll 1$ makes it possible to neglect the quadratic term in Eq. (24) as long as $\left|m(t-t_i)\right| \ll 1$ and $\sqrt{\{\mathbf{P}_i\}_{aa}}$ is comparable or smaller than $\left|\{\mathbf{P}_i\}_{aM}\right|$. Certainly, these derivations show that the higher order interpolating polynomials should capture two-body covariance propagation *exactly*.

Similarly, the blending approach can easily handle this case: by construction, the two-body covariance propagations from t_i and from t_{i+1} to the same time t will match. In other words in the notation of this paper, $\mathbf{P}_{(i)} = \mathbf{P}_{(i+1)}$, and any blending function will trivially lead to the exact answer of $\mathbf{P} = \mathbf{P}_{(i)} = \mathbf{P}_{(i+1)}$ - a result that is independent from covariance coordinates and matrix factorizations. This type of invariance, however, is not found when using standard interpolation. Indeed, as can be seen from example in Reference 2, in Cartesian coordinates the higher derivatives $\mathbf{P}^{(n)}$ become progressively smaller but never vanish. Hence, while the two-body evolution of covariance elements in Keplerian coordinates is at most quadratic, their evolution in Cartesian coordinates cannot be *exactly* captured by interpolating polynomials of *any order*.

The important practical question is how well the two approaches would perform when propagation deviates from exact two-body. Both approaches can retain and employ some two-body approximations: the higher order interpolation polynomials can use two-body Jacobians to approximate the first and second derivatives of tabulated covariances[1] while the blending can use approximate two-body error transitions.

Numerical Tests

The performances of various methods are examined using the following example. The LEO orbit is propagated from the initial state listed in Table 2 using Runge-Kutta-Fehlberg 7th order integrator with 8th order error control implemented in STK 10.[10] The propagated ephemeris and covariance data recorded at every second represent the “truth”.

Table 2. Example LEO Initial State.

Reference Epoch	22 Nov 2008 19:00:00 UTCG		
Position in ICRF (X, Y, Z)	-2397.20 km	4217.85 km	5317.45 km
Velocity in ICRF (X, Y, Z)	-1.3039 km/s	5.5589 km/s	-4.8396 km/s
Gravitational Constant	398600.4418 km^3/s^2		

The full force model is described in Table 3 with covariance propagation including the effects of gravity, drag and SRP.

Table 3. Full Force Model Parameters.

Gravity Field	WGS84_EGM96 21x21		
Tides	Permanent Solid Tides and Ocean Tides 4x4		
Third Body Gravity	Sun		Moon
Space Object	Spherical	Area 20 m^2	Mass 0.04 kg
Drag	Jacchia-Roberts (with flux file)		C_d = 2.2
SRP	Dual Cone Shadow Model		C_r = 1
Eclipsing Bodies	Earth		Moon
Albedo	Simple Reflection Model		

Relativity Correction	Included

The selected object is poorly tracked as evident from its very large initial error covariance (Table 4). This in conjunction with its highly dynamic propagation environment provides a challenging test case for interpolation and blending techniques.

Table 4. Example LEO Initial Error Covariance.

Reference Epoch	22 Nov 2008 19:00:00 UTCG		
Position Sigmas in ICRF (X, Y, Z)	98.676 km	420.547 km	366.438 km
Velocity Sigmas in ICRF (X, Y, Z)	0.194 km/s	0.341 km/s	0.430 km/s
Position Correlations in ICRF (XY, XZ, YZ)	-0.999985	0.999983	-0.999997
Velocity Correlations in ICRF (XY, XZ, YZ)	-0.999998	-0.999997	0.999997
Cross-Correlations in ICRF			
(Position X, Velocity X, Y, Z)	-0.999982	0.999989	0.999983
(Position Y, Velocity X, Y, Z)	0.999997	-0.999999	-0.999995
(Position Z, Velocity X, Y, Z)	-0.999996	0.999996	0.999999

These challenges are highlighted in Figure 2 which displays evolution of the position sigmas and correlation coefficients in ICRF over 10 min interval. Note that around the 476 second mark, when σ_{xx} reaches its sharp minimum of ≈ 0.56 km, some of the correlation coefficients transition rapidly between ≈ -1 and ≈ 1.

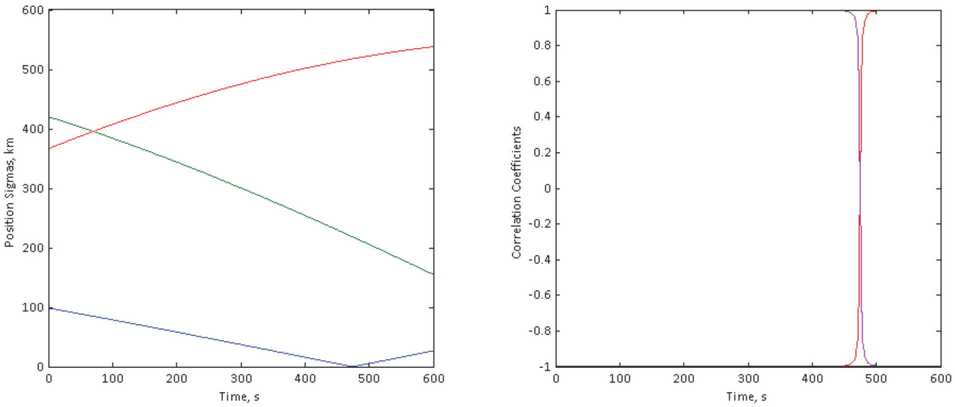

Figure 2. Evolution of Position Sigmas and Correlation Coefficients in ICRF over 10 min.

Two-Body Orbit. Consider first the challenges that even exact two-body propagation may present to interpolation methods. Indeed, the following can be observed interpolating in Cartesian coordinates (see Figures 3, 4): the poor accuracy of linear interpolation is improved when using the higher order polynomials but the small errors still remain even for the quintic polynomials.[*]

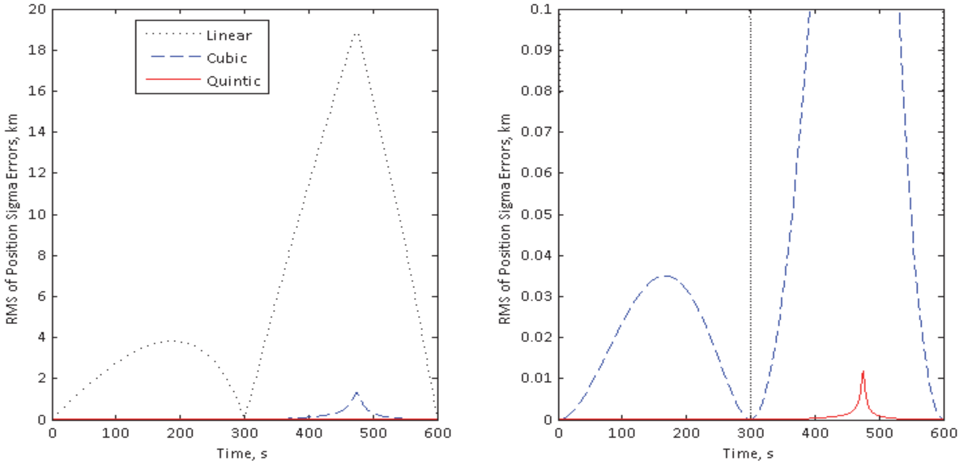

Figure 3. Position Sigma Errors Interpolating in ICRF in Cartesian Coordinates during two 5 min steps along Two-Body Reference Orbit.

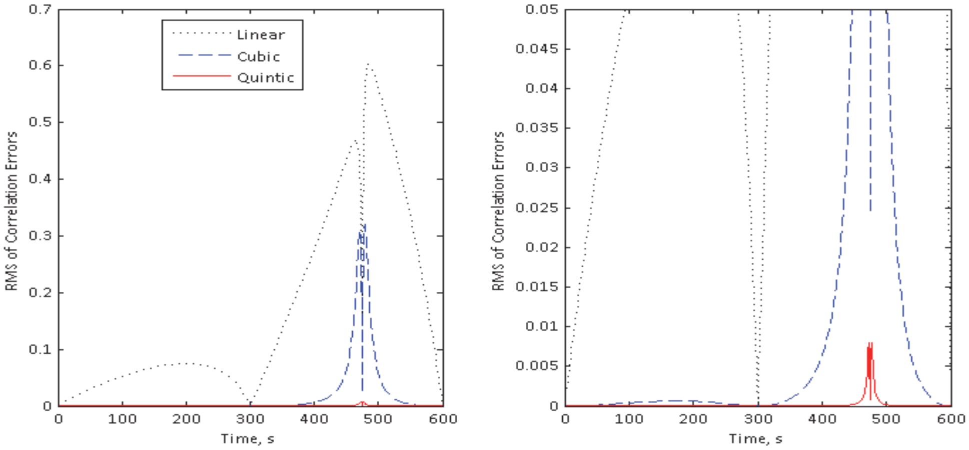

Figure 4. Correlation Errors Interpolating in ICRF in Cartesian Coordinates during two 5 min steps along Two-Body Reference Orbit.

By contrast, the same interpolating polynomials produce much better results when applied to the Keplerian elements (see Figure 5): in these coordinates even linear interpolation during 5 min steps is almost perfect[†] and, as predicted, the higher order polynomial interpolations are exact.

[*] The "true" and interpolated covariances are compared at 1 second steps. The data for interpolation is created by copying the "true" covariance data and then pruning it in accordance with the selected interpolation step.

[†] The second order term in Eq. (24) is about 4 orders of magnitude smaller than the first order term in this case.

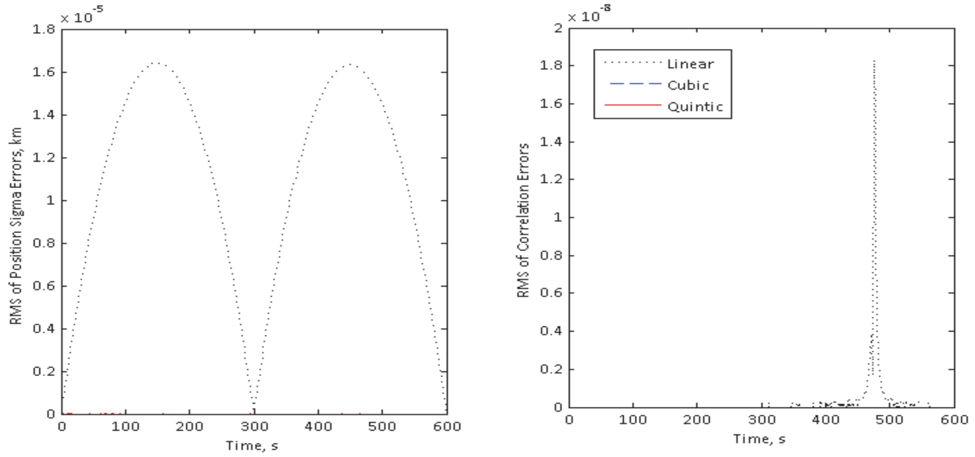

Figure 5. Position Sigma and Correlation Errors in ICRF Interpolating Keplerian Elements during two 5 min steps along Two-Body Reference Orbit.

Still, even in this most favorable case the blending approach is superior because using two-body error transitions (trivially) produces results that are by construction exactly accurate regardless of factorizations, coordinates or blending functions used.

Full Force Model Orbit. The practical tests of covariance interpolation must include a more physically realistic force model. The force model listed in Table 3 produces noticeable deviations from the two-body orbit and from the two-body error transitions. The tests using this force model are performed over longer interpolation intervals and over a longer period of time - 2 hours (see Figure 6).

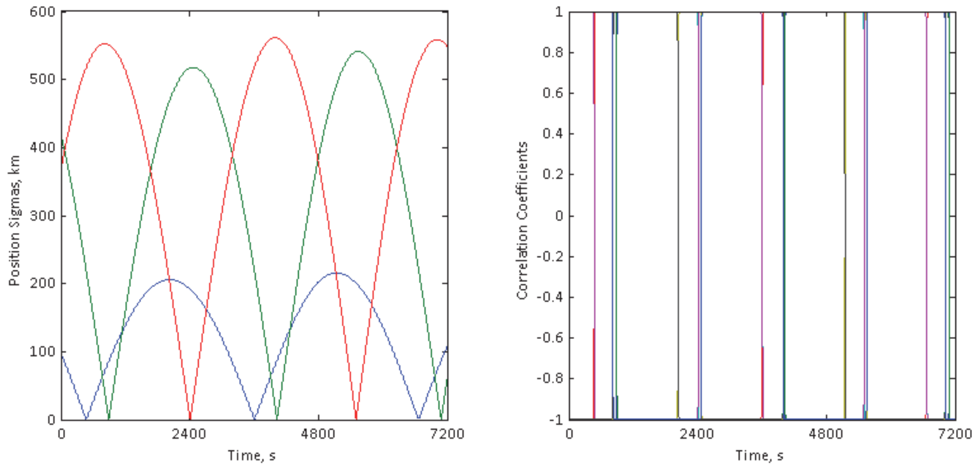

Figure 6. Evolution of Position Sigmas and Correlation Coefficients in ICRF over 2 hours.

Since the low order polynomial interpolations perform poorly in these tests, for brevity, only results obtained by using the quintic polynomials are included in the paper. Even at this high order, interpolation in Cartesian coordinates is significantly inferior to interpolation of the Keplerian elements (see Figure 7). In particular, the errors in the ICRF Cartesian interpolation spike near the minimums of the ICRF sigma values when the correlation coefficients rapidly change signs.

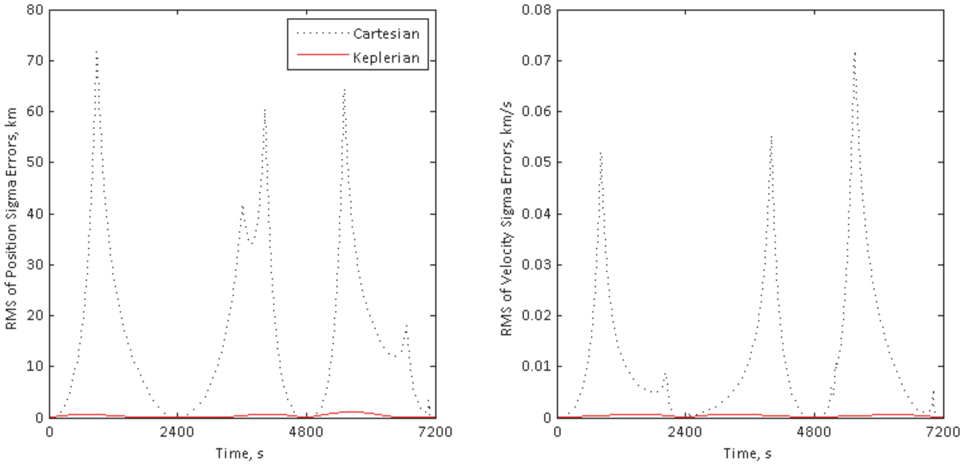

Figure 7. Position and Velocity Sigma Errors in ICRF Interpolating in Cartesian Coordinates and in Keplerian Elements using Quintic Polynomials during three 40 min steps along Full Force Model Reference Orbit.

The geometric blending also performs poorly in the full force model tests and so its results too are also omitted from the paper for brevity. The arithmetic blending performs well but shows little variability for blending functions of different orders. Hence, again for brevity, only results based on the quadratic blending function are included in the paper. As stated previously, the choice of coordinates does not affect blending of whole matrices but can affect blending of square root or sigma correlation factorizations.* These results are illustrated in Figure 8 where the quadratic blending of these factorizations in Cartesian and Keplerian coordinates is shown alongside with the quadratic blending of whole covariance matrices and their position/velocity block decompositions.

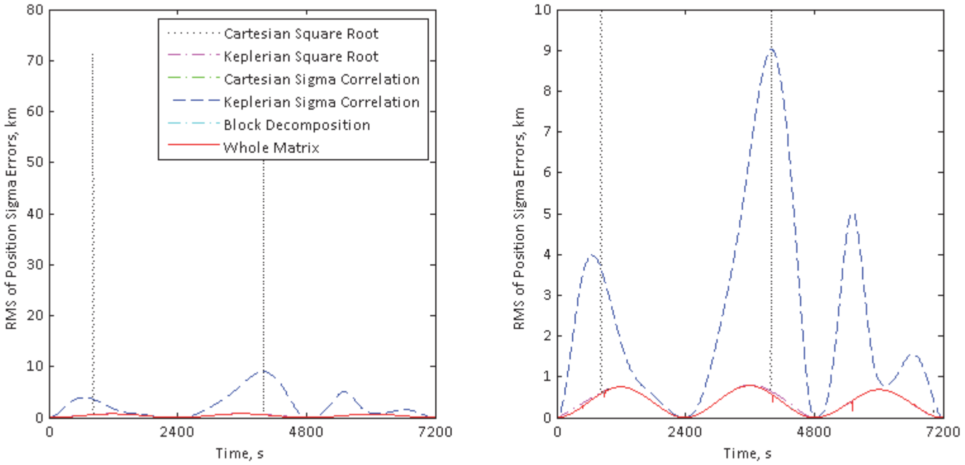

Figure 8. Position Sigma Errors in ICRF Blending Various Decompositions using Quadratic Function during three 40 min steps along Full Force Model Reference Orbit.

* Complete eigen-decompositions of 6x6 matrices are not examined here as they would be too computationally costly.

The blending results for the whole covariance matrices and for their position/velocity block decompositions are virtually identical. Also, very close to them are the blending results for the Cartesian sigma correlation factorizations and the Keplerian square root factorizations. It is interesting that the blending of Cartesian square root factorizations suffers from dramatic error spikes near some of the minimums of the ICRF sigma values, the effect that is not seen during the ICRF blending of whole covariance matrices. This demonstrates the dangers of applying non-physical mathematical fitting techniques in poorly selected (highly dynamic) coordinates.* In this light, it may seem surprising that the blending accuracy of the Keplerian sigma correlation matrices is noticeably worse than that of the Cartesian sigma correlation matrices. Note, however, that by construction the ICRF sigma values of the Cartesian sigma correlation matrices are blended identically to those of the original covariance matrices. The same cannot be said about the ICRF sigma values created after blending of the Keplerian sigma correlation matrices: in this case, the errors introduced by the non-physical mathematical fitting techniques applied in Keplerian elements propagate though Keplerian to Cartesian partials to all Cartesian values.

To summarize, it appears that, when blending with two-body error transitions, the best chance to avoid unintended numerical behavior is to skip various factorizations in favor of a straightforward blending of whole covariance matrices. In essence, this minimizes the impact of non-physical mathematical fitting in favor of the physical (albeit approximate) evolution of covariance matrices.

The above conjecture leads to another question: is it possible to improve approximate error transitions without resorting to numerical integration of more sophisticated force models? After two-body gravity, one of the most important dynamical effects is due to J2 – the dominant non-spherical contribution of gravity. It is possible to account for the secular effect of J2 by using the analytical error transitions based on Kozai mean elements[11]. The result of quadratic blending with the secular J2 error transitions is shown in Figure 9. It also shows the result of quadratic blending with the two-body error transitions and the result of interpolation using the quintic polynomials.

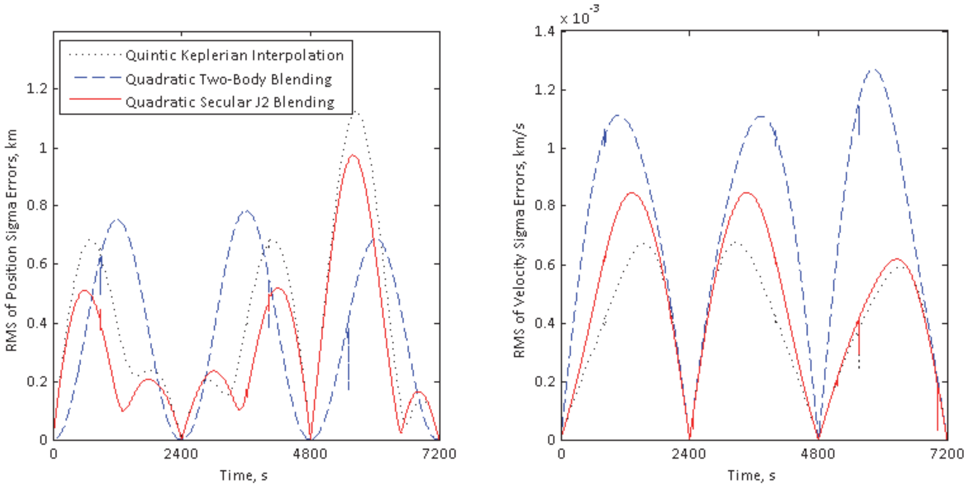

Figure 9. Position and Velocity Sigma Errors in ICRF Interpolating in Keplerian Elements and Blending using Two-Body and using Secular J2 Error Transitions during three 40 min steps along Full Force Model Reference Orbit.

* A similar observation is made by Alfano in Reference 2 with regards to interpolation.

These results illustrate the best possible solutions to the covariance interpolation problem among all of the different interpolation and blending techniques considered in this paper. Note that, although the sigma errors appear large in absolute terms, all of the errors are less than 0.4 % of the respective sigma values. This level of accuracy can be quite adequate for covariance data which is often expected to be known only with one or two significant digits.

In order to better assess the accuracy and robustness of the three best methods, they are applied over the same two hour interval but using covariances tabulated at different steps: 60 sec, 300 sec, 600 sec, 1200 sec, 1800 sec, 2400 sec, and 3600 sec. For each test case, the maximum and the average RMS over the entire two hour interval are computed based on the RMS of errors recorded at 1 second steps. These are plotted in Figures 10-12 as functions of the covariance step.

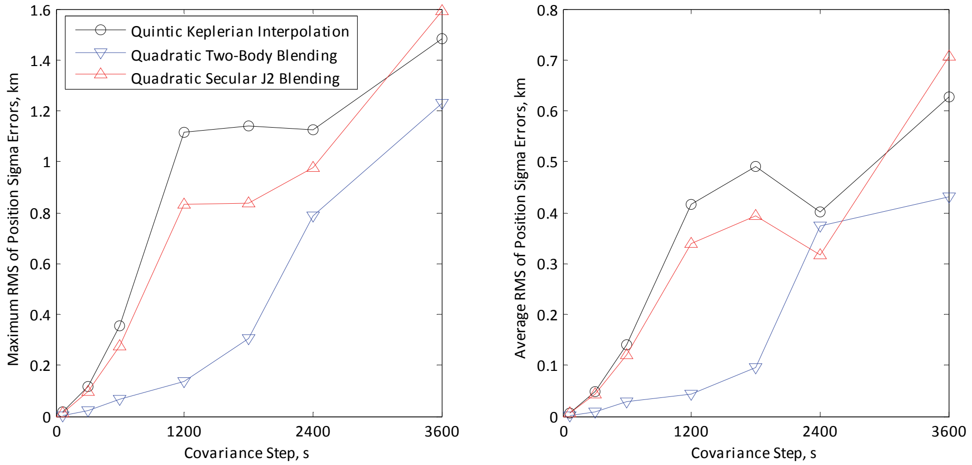

Figure 10. Maximum and Average Position Sigma Errors in ICRF Interpolating in Keplerian Elements and Blending using Two-Body and using Secular J2 Error Transitions as Functions of Covariance Step along Full Force Model Reference Orbit.

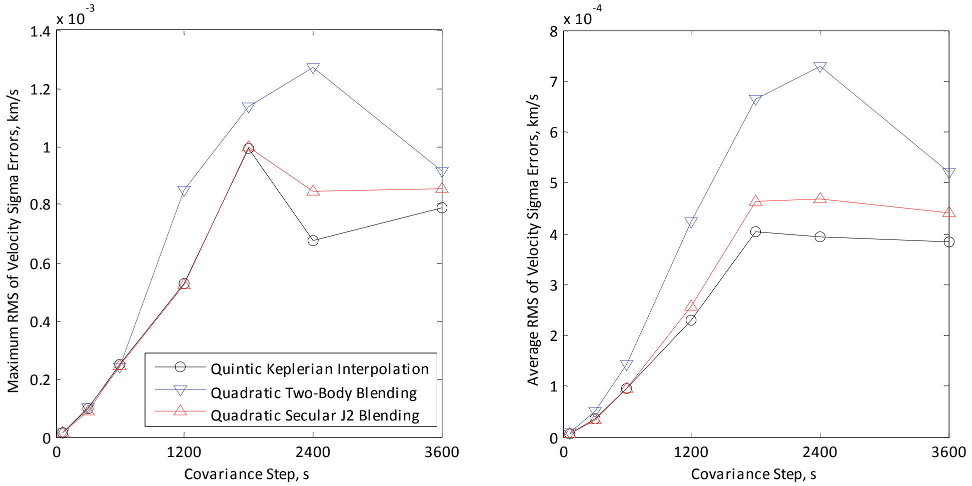

Figure 11. Maximum and Average Velocity Sigma Errors in ICRF Interpolating in Keplerian Elements and Blending using Two-Body and using Secular J2 Error Transitions as Functions of Covariance Step along Full Force Model Reference Orbit.

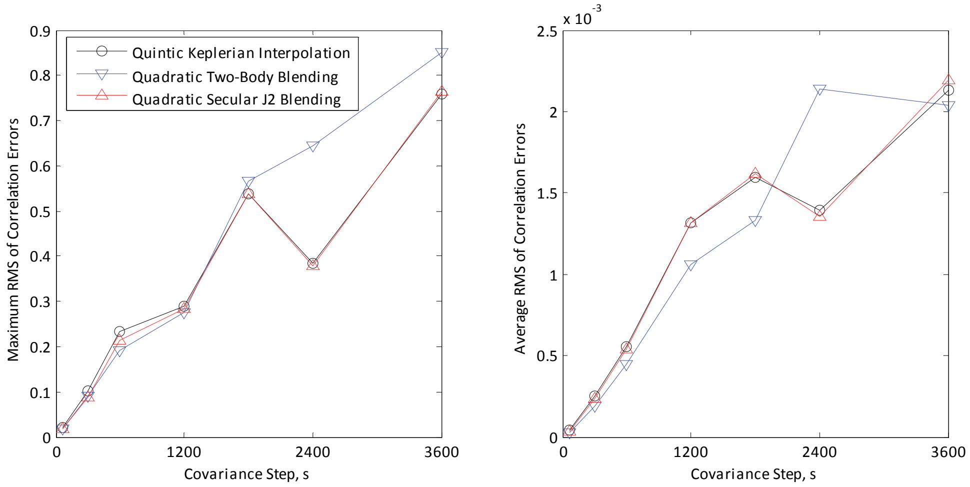

Figure 12. Maximum and Average Correlation Errors in ICRF Interpolating in Keplerian Elements and Blending using Two-Body and using Secular J2 Error Transitions as Functions of Covariance Step along Full Force Model Reference Orbit.

The quadratic blending with the two-body error transitions generally outperforms the other two methods for accurate interpolation of the position sigmas (see Figure 10) and the quintic interpolation in Keplerian elements generally outperforms the other two methods for accurate interpolation of the velocity sigmas (see Figure 11). The quadratic blending with the secular J2 error transitions generally remains in between of the other two methods but significantly closer to the quintic interpolation in Keplerian elements (see Figures 10-12). The average RMS interpolation errors of the correlation coefficients are quite low (they are less than 0.0025 even with one hour interpolation step), however, the high maximum RMS errors indicate that occasional narrow spikes in correlation errors persist (Figure 12).

DISCUSSION

The best three methods for interpolating orbit covariance identified in the previous section all have comparable accuracy. However, they do differ in terms of their computational robustness and cost.

The quintic interpolation is simple to evaluate in Keplerian elements but, assuming that covariance data is stored natively in Cartesian coordinates, requires computing Cartesian to Keplerian and Keplerian to Cartesian partials. These are somewhat expensive computations which require special care for near-circular and near-equatorial orbits. A greater potential problem with this method is its inability to ensure positive definiteness of the interpolated covariance. This problem may seem unlikely because it can be argued that four of the five degrees of freedom of the quintic polynomials are spent matching the quadratic two-body behaviors at the end points of the interpolation interval and that the remaining degree of freedom is simply used to linearly transition these behaviors through the interval. However, this is not a rigorous argument and at least in theory the quintic interpolation can lead to the loss of positive definiteness.

The other two methods both use the blending approach and therefore guarantee positive definiteness of the blended covariance. The secular J2 blending method, similar to the quintic interpolation in Keplerian elements, requires additional transformations from and to Cartesian coordinates, and therefore also suffers from additional computational costs.

The two-body blending method appears to be the most computationally straightforward, can be implemented easily in Cartesian coordinates, and ensures that the blended covariance remains positive definite via a simple well understood transition of its elements between the propagated two-body covariances. In addition, this method is shown to yield the most accurate interpolation of the position portion of orbit covariance which makes it especially attractive for CA applications.

Reduction of Storage Footprint

If accurate interpolation can be retained using covariances tabulated at longer steps, then the amount of stored covariance data required for the same analysis interval can be reduced. Consider binary storage of ephemeris and covariance data: the time is stored as a double precision value in seconds since reference epoch; the ephemeris is stored as 6 double precision values, 3 for Cartesian position and 3 for Cartesian velocity; the covariance is stored in a lower triangular form, as 21 double precision values. Figure 13 shows storage footprints for various covariance steps over the analysis interval of 5 days.

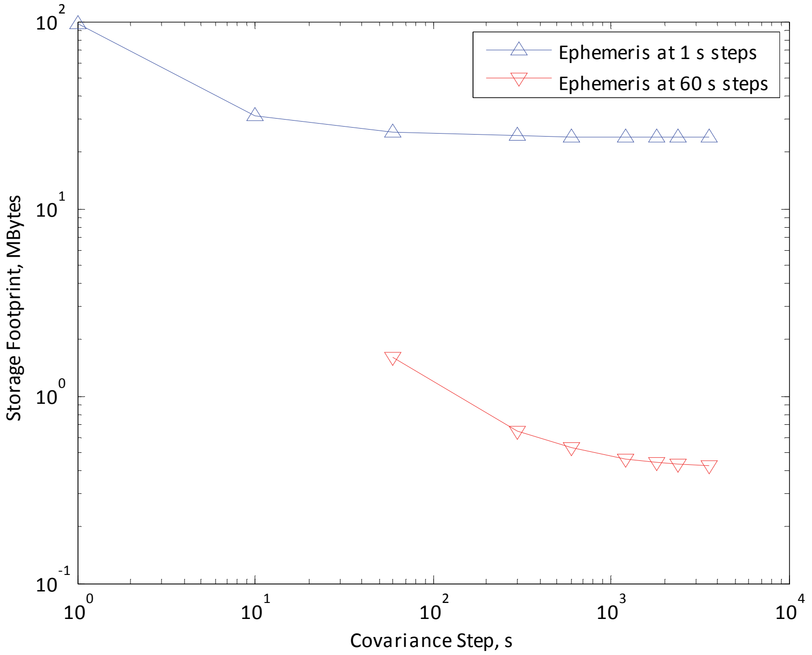

Figure 13. Storage Footprint as Function of Covariance Step over 5 day Interval.

CONCLUSIONS

This paper introduced a new approach for covariance interpolation based on blending of approximate error transitions. The approach guarantees positive definiteness of the blended covariances and can be applied to both whole matrices and various factorizations. The approach is implemented and tested for orbit covariance interpolation. The tests demonstrate that accurate, robust and efficient covariance interpolation can be obtained by a simple arithmetic blending of covariances propagated using two-body error transitions. Retaining accurate interpolation for covariances tabulated at longer steps provides an opportunity to reduce storage footprint of covariance data which can be important for maintaining and accessing large catalogs.

APPENDIX

Geometric Blending

The geometric (or Riemannian) blending traverses a geodesic on the manifold of symmetric positive definite matrices. This manifold is equipped with a distance metric that is based on the principal logarithm of $\mathbf{P}_{(i)}^{-1}\mathbf{P}_{(i+1)}$:[*] $d_G(\mathbf{P}_{(i)}, \mathbf{P}_{(i+1)}) \triangleq \left\| \mathrm{Log}(\mathbf{P}_{(i)}^{-1}\mathbf{P}_{(i+1)}) \right\|_F = \sqrt{\sum \ln^2 \mathrm{eig}(\mathbf{P}_{(i)}^{-1}\mathbf{P}_{(i+1)})}$. Formulation of geometric blending is more complicated than that of arithmetic blending (see Eq.(7)):

$$\mathbf{P}_{(i)\beta} = \mathbf{P}_{(i)}\,\mathrm{Exp}(\beta\,\mathrm{Log}(\mathbf{P}_{(i)}^{-1}\mathbf{P}_{(i+1)})) = \mathrm{Exp}(\left[1-\beta\right]\mathrm{Log}(\mathbf{P}_{(i)}\mathbf{P}_{(i+1)}^{-1}))\mathbf{P}_{(i+1)}\,, \tag{29}$$

which can also be written in one of the following equivalent forms:

$$\begin{aligned} \mathbf{P}_{(i)\beta} &= \mathbf{P}_{(i)}\left(\mathbf{P}_{(i)}^{-1}\mathbf{P}_{(i+1)}\right)^{\beta} = \left(\mathbf{P}_{(i)}\mathbf{P}_{(i+1)}^{-1}\right)^{1-\beta}\mathbf{P}_{(i+1)} \\ &= \mathbf{L}_{(i)}^{T}\left(\mathbf{L}_{(i)}^{-T}\mathbf{P}_{(i+1)}\mathbf{L}_{(i)}^{-1}\right)^{\beta}\mathbf{L}_{(i)} = \mathbf{L}_{(i+1)}^{T}\left(\mathbf{L}_{(i+1)}^{-T}\mathbf{P}_{(i)}\mathbf{L}_{(i+1)}^{-1}\right)^{1-\beta}\mathbf{L}_{(i+1)} \end{aligned} \tag{30}$$

where, as before, $\mathbf{L}$ is defined from $\mathbf{P} = \mathbf{L}^T\mathbf{L}$ and obtained via the Cholesky factorization.[6]

REFERENCES

[1] P. García, D. Escobar, A. Agueda, and F. Martínez, "Covariance Determination, Propagation and Interpolation Techniques for Space Surveillance," European Space Surveillance Conference, Madrid, Spain, June 7–9, 2011.

[2] S. Alfano, "Orbital Covariance Interpolation," Paper AAS 04-223, AAS/AIAA Space Flight Mechanics Meeting, Maui, HI, Feb. 8–12, 2004.

[3] J. Woodburn, and S. Tanygin, "Position Covariance Visualization," Paper AIAA 2002-4985, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Monterey, CA, Aug. 5–8, 2002. doi: 10.2514/6.2002-4985

[4] O. Montenbruck, and E. Gill, *Satellite Orbits*. Springer-Verlag Berlin Heidelberg, Berlin, 2nd ed., 2001, Ch. 8.3.

[5] M. Moakher, "A Differential Geometric Approach to the Geometric Mean of Symmetric Positive-Definite Matrices," *SIAM. J. Matrix Anal. & Appl*., Volume 26, Issue 3 (2005), pp. 735–747. doi: 10.1137/S0895479803436937

[6] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*. Dover Publications, Inc., New York, 2006, Appendix III.A.

[7] G. H. Golub, and C. F. Van Loan, Matrix Computations. The Johns Hopkins University Press, Baltimore, 3rd ed., 1996, Ch. 8.4.3.

[8] N. J. Higham, and L. Lin, "A Schur–Padé Algorithm for Fractional Powers of a Matrix," *SIAM. J. Matrix Anal. & Appl*., Volume 32, Issue 3 (2011), pp. 1056–1078. doi: 10.1137/10081232X

[9] O. Montenbruck, and E. Gill, *Satellite Orbits*. Springer-Verlag Berlin Heidelberg, Berlin, 2nd ed., 2001, Ch. 7.1.

[10] http://www.agi.com/

[11] Y. Kozai, "The motion of a close earth satellite," *The Astronomical Journal*, Vol. 64, 1959, pp. 367–377. doi: 10.1086/107957

[*] The eigenvalues of $\mathbf{P}_{(i)}^{-1}\mathbf{P}_{(i+1)}$ are guaranteed to be real positive even though the matrix itself is not symmetric.[5]