

ORBIT AND COVARIANCE INTERPOLATION/BLENDING WITH OREKIT

Vincent Cucchietti

Maxime Journot

Pascal Parraud

May 2023

Abstract

This technical note presents the results as well as the implementation of the covariance blending method, based on "Efficient Covariance Interpolation Using Blending Of Approximate State Error Transitions" by Sergei Tanygin, in the open source low level space dynamics library Orekit available at <https://www.orekit.org/>.

Introduction

As of 2023, space situational awareness is becoming a crucial issue as constellations, made of thousands of satellites, are becoming a reality. In this light, covariance propagation and interpolation are necessary for operators to compute the right collision probability and hence, to realize a collision avoidance maneuver when necessary. That is why Orekit, which currently does not handle covariance interpolation, needs new methods to stay up to date with this evolution in the space field.

First, this document briefly explain how blending differs from a standard interpolation. Main differences and difficulties between [1] and the current implementation are detailed. Results with two-body keplerian model and full force model are presented using a very similar test case as the one presented in [1]. Finally, an investigation is conducted on orbit blending to better understand its effectiveness compared to standard quintic Hermite interpolation.

Covariance blending

The "blending" approach of [1] will be briefly explained in this section. For more information, feel free to investigate [1] as it will be much more detailed.

In the beginning of [1], the commonly known "interpolating" approach is promptly differentiated from the introduced "blending" approach. Indeed, while the interpolation uses the information from the tabulated states and covariances (and eventually their time derivatives), the goal of blending is to propagate both tabulated state covariances to the same time t and blend them using a smoothstep function.

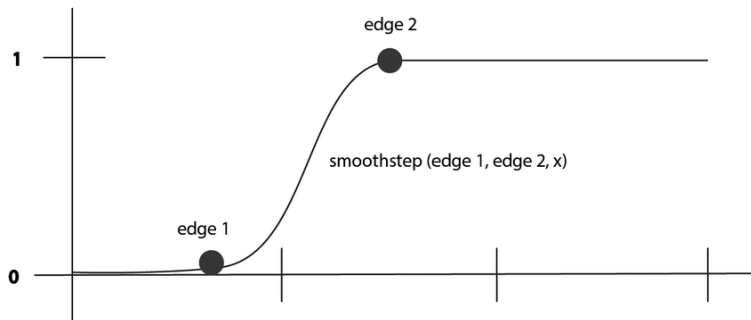


Figure 1: A smoothstep function

This method brings multiple benefits, the main one being that it ensures the positive definiteness of the blended covariance matrix whereas a standard interpolation technique does not fundamentally guarantee this property.

Additional complexity

One of the main problem encountered while implementing the new blending method was that "approximate state error transition matrix" mentioned in [1] was not detailed. Thus, it was first assumed that it was referring to the approximate state error transition matrix shown in [2, p. 811] :

$$\Phi_{Cartesian} \cong \begin{bmatrix} 1 + \frac{3\mu\Delta t^2 r_I^2}{2r_o^5} - \frac{\mu\Delta t^2}{2r_o^3} & \frac{3\mu\Delta t^2 r_I r_J}{2r_o^5} & \frac{3\mu\Delta t^2 r_I r_K}{2r_o^5} & \Delta t & 0 & 0 \\ \frac{3\mu\Delta t^2 r_I r_J}{2r_o^5} & 1 + \frac{3\mu\Delta t^2 r_J^2}{2r_o^5} - \frac{\mu\Delta t^2}{2r_o^3} & \frac{3\mu\Delta t^2 r_J r_K}{2r_o^5} & 0 & \Delta t & 0 \\ \frac{3\mu\Delta t^2 r_I r_K}{2r_o^5} & \frac{3\mu\Delta t^2 r_J r_K}{2r_o^5} & 1 + \frac{3\mu\Delta t^2 r_K^2}{2r_o^5} - \frac{\mu\Delta t^2}{2r_o^3} & 0 & 0 & \Delta t \\ \frac{3\mu\Delta t r_I^2}{r_o^5} - \frac{\mu\Delta t}{r_o^3} & \frac{3\mu\Delta t r_I r_J}{r_o^5} & \frac{3\mu\Delta t r_I r_K}{r_o^5} & 1 & 0 & 0 \\ \frac{3\mu\Delta t r_I r_J}{r_o^5} & \frac{3\mu\Delta t r_J^2}{r_o^5} - \frac{\mu\Delta t}{r_o^3} & \frac{3\mu\Delta t r_J r_K}{r_o^5} & 0 & 1 & 0 \\ \frac{3\mu\Delta t r_I r_K}{r_o^5} & \frac{3\mu\Delta t r_J r_K}{r_o^5} & \frac{3\mu\Delta t r_K^2}{r_o^5} - \frac{\mu\Delta t}{r_o^3} & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

However, this approximate state error transition matrix gave very unsatisfying results. Indeed, this matrix is built using Taylor series and cannot be used when Δt becomes significant in front of the orbital period. In [3], Alfano suggests that Δt should not exceed 15° of orbital motion.

A straightforward solution to this problem was to use the exact state error transition matrix for a pure keplerian motion defined in equinoctial elements to avoid singularities^{1 2}:

$$\Phi_{Keplerian} = \Phi_{Equinoctial} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{3}{2} \frac{n}{a} \Delta t & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Then the following procedure is needed to propagate the tabulated covariances to the interpolating time t in cartesian elements :

1. Convert adjacent tabulated covariances from Cartesian to equinoctial elements (assuming that tabulated covariances were defined in Cartesian elements) (see [4]).
2. Compute equinoctial state error transition matrix using 2.
3. Compute forwarded and backwarded covariances to interpolating time t .
4. Blend them using a blending function.
5. Convert the blended covariance from equinoctial to cartesian elements (assuming that the user want to define output covariance in Cartesian elements).

However, this solution comes with an additional cost being the need to know the state in order to convert from cartesian to equinoctial elements and vice versa. This is especially true at interpolating time t where we do not know the state. Hence, the need to interpolate the state as well as the covariance at t .

¹Note that, later on, it will still be referred to as the keplerian state error transition matrix as it is the most intuitive way to understand it.

²An equivalent state transition matrix which include J2 impact could be used to be even more accurate. However, it was not studied here because of time constraints.

Orekit implementation

In order to implement blending inside Orekit, it was first decided to implement the basic elements necessary for blending in Hipparchus³:

- Addition of a `Blendable` interface providing a method to blend arithmetically object with other object using a blending value comprised between [0:1]
- Addition of a `SmoothStepFactory` class enabling easy creation of `SmoothStepFunction` for the user.

These additions can be visualized in 2 below :

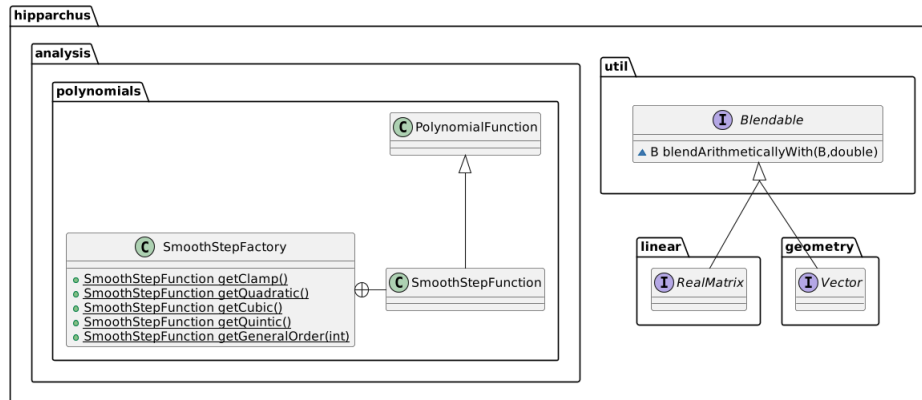


Figure 2: UML diagram of the blending implementation in Hipparchus [Issue 212]

Regarding the implementation in Orekit, a refactoring of the way interpolation was done was deemed necessary. In this light, a `TimeInterpolator` interface was added as a way to remain very modular. Interpolating and blending classes would implement this interface, and the user would always expect the same "interpolate" method signature. A very simplified UML representation of the Orekit implementation is shown below in 3 :

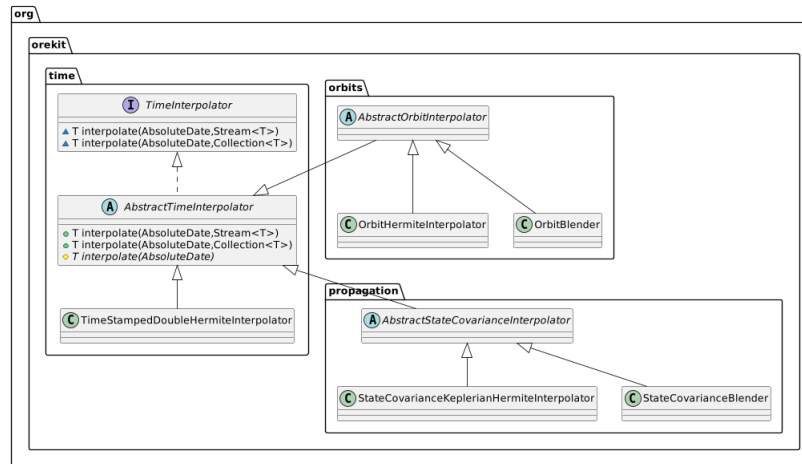


Figure 3: UML diagram of the blending implementation in Orekit [Issue 970]

The `TimeInterpolator` interface is in fact implemented by lots of classes through `AbstractTimeInterpolator` but they are not shown here as they would hinder the diagram readability in addition to be out of context. A "Field" version also exists but it is not mentioned for the same reasons mentioned previously.

³The Hipparchus project is a library of lightweight, self-contained mathematics and statistics components addressing the most common problems not available in the Java programming language. It serves as a foundation to Orekit regarding everything related to mathematics

Tests results

In this section, the test case used here will first be presented and compared to the one in [1]. Then the results computed using a simple keplerian propagation model as well as the full force model will be detailed. In both cases two methods will be compared and the results will be briefly discussed. The "reference data" mentioned later on refers to the data generated with a one-second step, it serves as a basis to evaluate the data obtained through both methods.

Test case

The test case used here is slightly different from the one in [1]. In both cases, the same poorly tracked LEO object evolving in a highly dynamical propagation environment is defined :

Orbit keplerian elements					
a (km)	e	i (°)	ω (°)	Ω (°)	M (°)
7283.18	0.01945	98.35	77.72	-69.6498	53.9737

Table 1: Test case initial orbit in keplerian elements

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

Table 2: Test case initial state in cartesian elements

Reference epoch	22 Nov 2008 19:00:00 UTG		
Position Sigmas in GCRF (X, Y, Z)	98.676 km	420.547 km	366.438 km
Velocity Sigmas in GCRF (X, Y, Z)	0.194 km/s	0.341 km/s	0.430 km/s
Position Correlations in GCRF (XY, XZ, YZ)	-0.999985	0.999983	-0.999997
Velocity Correlations in GCRF (XY, XZ, YZ)	-0.999998	-0.999997	0.999997
Cross-Correlations in GCRF			
(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

Table 3: Test case initial covariance in cartesian elements

Then, results were obtained using the open source low level space mechanics library Orekit [5], this led to slight differences with the test case configuration used in [1] :

- Use of the Dormand-Prince 8(5,3) (i.e. order 8 for the integration and order 6 for error estimation) instead of the Runge-Kutta-Fehlberg 7th order integrator with 8th order error control.
- Use of the IERS2010 EIGEN6S 21x21 ellipsoid and gravity field instead of the WGS84 EGM96 21x21 (although the same ellipsoid and gravity field could have been used if needed).
- Use of the NRLMSISE00 atmosphere instead of the Jacchia-Roberts.
- Finally, the albedo and relativity correction models were not mentioned so the models used here might differ.

Gravity Field	IERS2010 EIGEN6S 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	Isotropic drag with NRLMSISE00		$C_d = 2.2$
SRP	Dual Cone Shadow Model		$C_r = 1$
Eclipsing Bodies	Earth		Moon
Albedo	Knocke Earth Albedo and IR emission [6]		
Relativity Correction	Included [7, equation 3.146]		

Table 4: Full Force Model Parameters used

Despite a slightly different model, Figure 4 can hardly be distinguished from [1, Figure 6] :

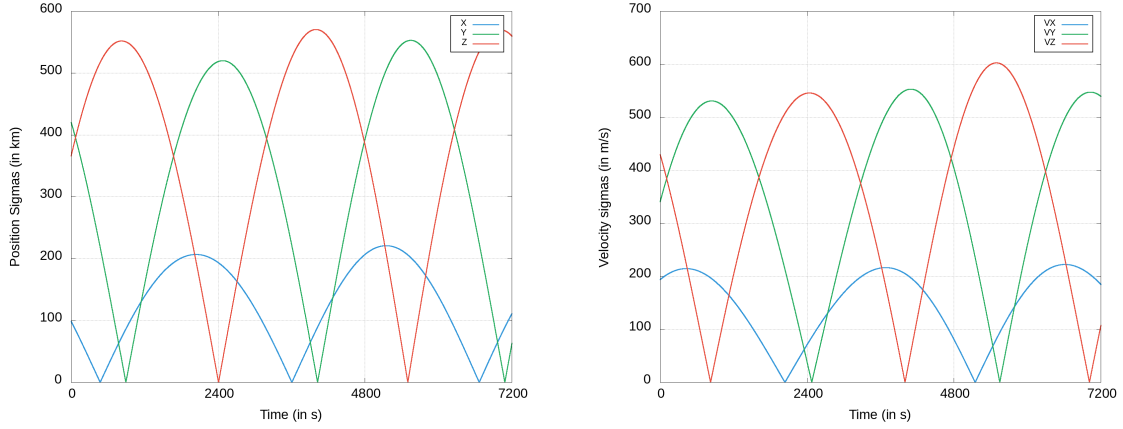


Figure 4: Evolution of Position and Velocity Sigmas in GCRF over 2 hours.

Methodology

Interpolating and blending methods

In [1], it is concluded that the best three methods are the following:

- Quadratic two-body blending.
- Quadratic secular J2 blending.
- Quintic keplerian interpolation (assumed to be hermite interpolation).

This paper will focus on the quadratic two-body blending and quintic keplerian interpolation only as they were easily implementable.

Comparison method

To compare each method with the reference, the absolute and relative Root Mean Square, more commonly referred to as "RMS", of the position and velocity sigmas errors will be used.

They are defined as follows :

$$\begin{aligned}
 AbsoluteError_{RMS} &= \sqrt{\frac{(\sigma_{x_{reference}} - \sigma_{x_{method}})^2 + (\sigma_{y_{reference}} - \sigma_{y_{method}})^2 + (\sigma_{z_{reference}} - \sigma_{z_{method}})^2}{3}} \\
 RelativeError_{RMS} &= \frac{AbsoluteError_{RMS}}{Reference_{RMS}}
 \end{aligned} \tag{3}$$

Keplerian model

The first tests were run using a simple two-body model. As pointed out in [1], no differences are expected with the reference results as the blending method is, by construction, using this very model to interpolate the covariance. Moreover, it is shown in [1] that the covariance in keplerian elements behaves quadratically so any polynomials of order ≥ 2 should capture this evolution exactly.

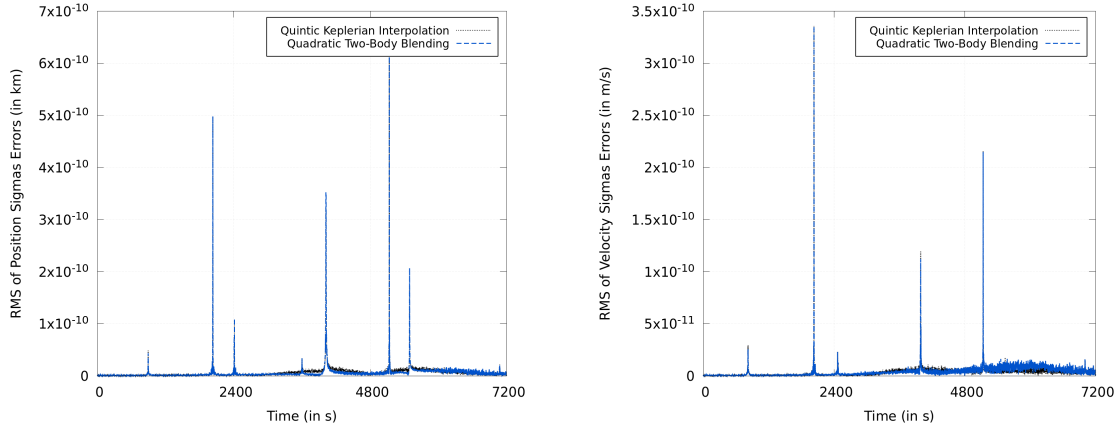


Figure 5: Evolution of Absolute Position and Velocity Sigmas Errors in GCRF over 2 hours with steps of 40mn along Two-Body Reference Orbit.

As shown in Figure 5 and except for some numerical noise, both the quadratic two-body blending and quintic keplerian interpolation methods give exact covariance interpolation as anticipated.

Full force model

In order to reproduce the results of [1, Figure 9], it was first assumed that the "true" state at time t from the reference was known.

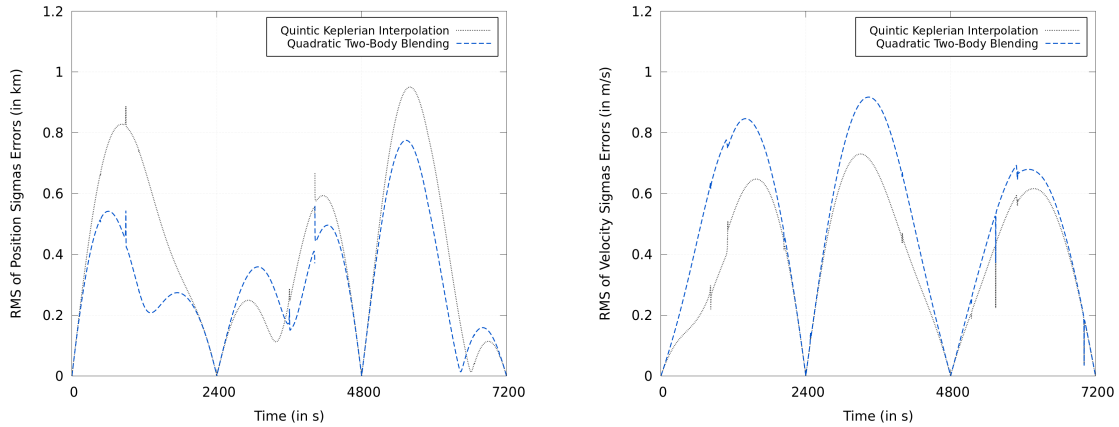


Figure 6: Evolution of Absolute Position and Velocity Sigmas Errors using the "true" State in GCRF over 2 hours with steps of 40mn along Full Force Reference Orbit.

Noticeable differences can be seen between Figure 6 and [1, Figure 9]. However, the order of magnitude remains very similar and even slightly better than [1, Figure 9].

Although the slight differences in models arguably contribute to these discrepancies, they cannot explain such differences observed here. As no explanation could be found and the results nonetheless satisfying, it was decided to pursue the study using an interpolated state instead of the "true" state.

Several methods to interpolate the state were possible given the four tabulated states available (with both derivatives) but after several testing, using three tabulated states as interpolating points gave the best results.

Hence, only the results which interpolate the state with the Hermite interpolation polynomials of degree 8 are shown in Figure 7 :

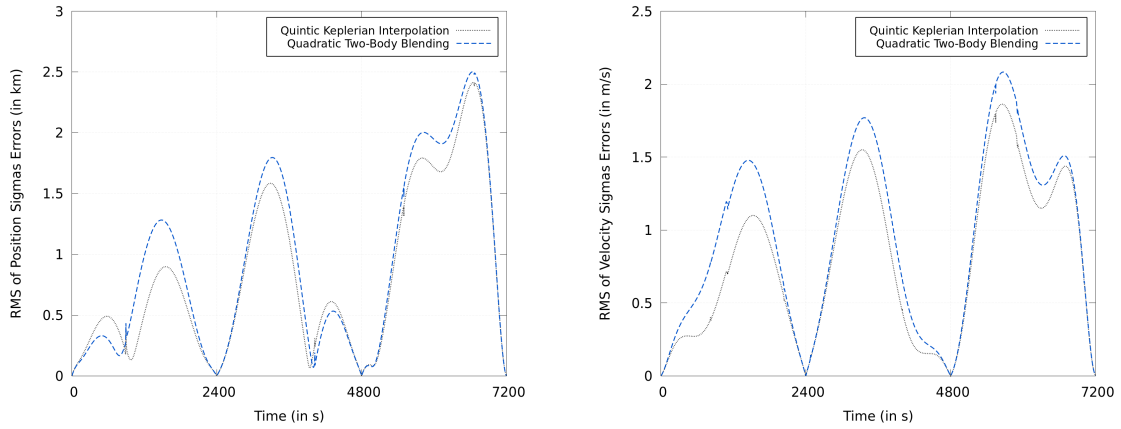


Figure 7: Evolution of Absolute Position and Velocity Sigmas Errors using an Interpolated State (Hermite Interpolation Polynomials of Degree 8) in GCRF over 2 hours with steps of 40mn along Full Force Reference Orbit.

Here, the quintic keplerian interpolation is only slightly better than the quadratic two-body blending but it can be seen that the error is now mainly due to the introduction of an interpolated state.

It was then thought that the concept of blending could be applied to the interpolated state as well, hence the Figure 8 :

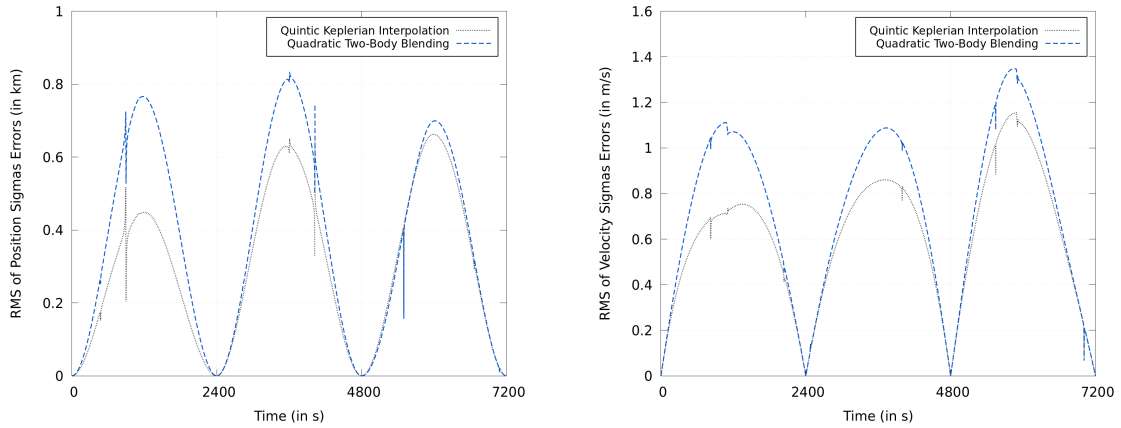


Figure 8: Evolution of Absolute Position and Velocity Sigmas Errors using a Blended State (Quadratic Two-Body Blending) in GCRF over 2 hours with steps of 40mn along Full Force Reference Orbit.

An unintended consequence of using a blended state is that the same results as [1, Figure 9], regarding the quadratic two-body blending method, are found. Moreover, results are much better than when the state was interpolated by polynomials and it can be seen that the quintic keplerian interpolation is superior over the quadratic two-body blending.

It is pointed out to the reader that, although the RMS of position and velocity sigmas errors seems significant in absolute terms, they are in fact almost negligible in relative terms as shown in Figure 9 :

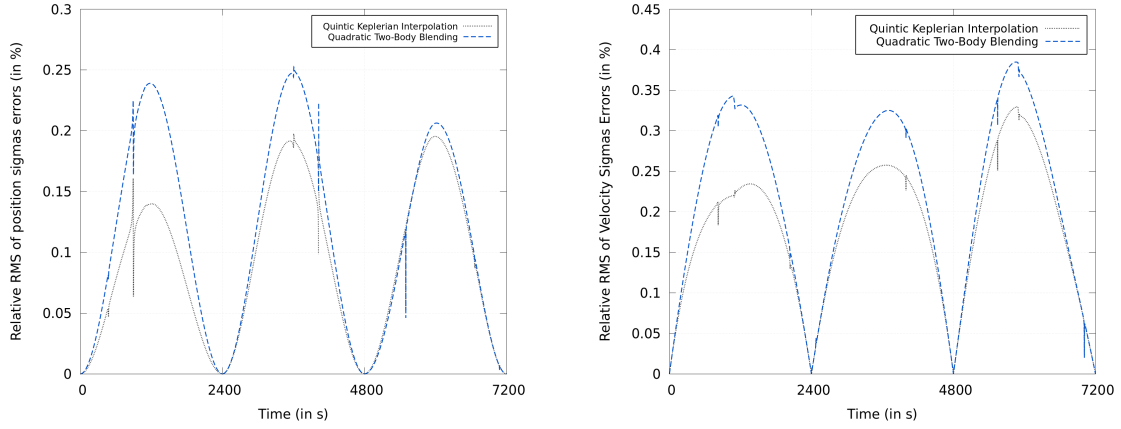


Figure 9: Evolution of Relative Position and Velocity Sigmas Errors using a Blended State (Quadratic Two-Body Blending) in GCRF over 2 hours with steps of 40mn along Full Force Reference Orbit.

Finally, the maximum and average relative RMS in position and velocity sigmas errors were computed using tabulated values with steps of 60, 300, 600, 1200, 1800, 2400 and 3600 s in Figures 10 and 11 :

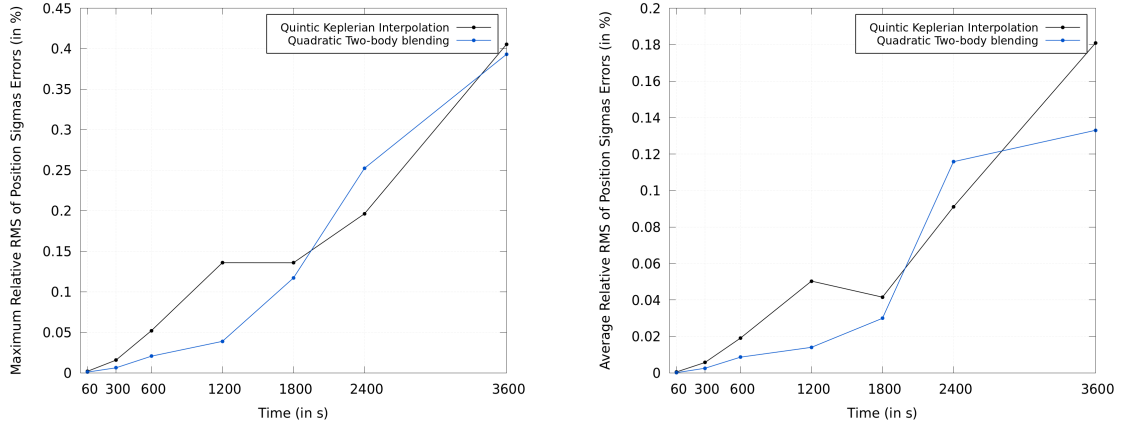


Figure 10: Maximum and Average relative RMS Position Sigmas Errors Using a Blended State (Quadratic Two-Body Blending) in GCRF when Interpolating Covariance with Different Tabulated Steps along Full Force Model Reference Orbit.

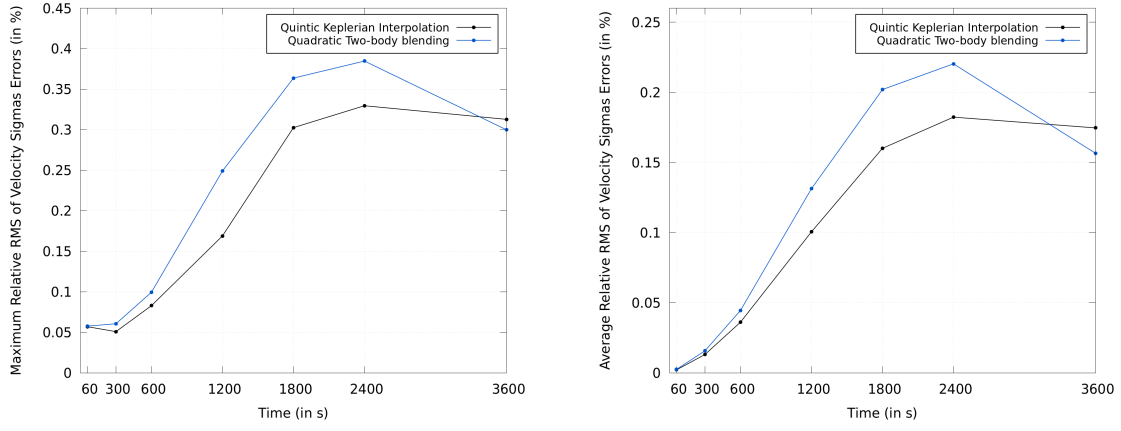


Figure 11: Maximum and Average relative RMS Velocity Sigmas Errors Using a Blended State (Quadratic Two-Body Blending) in GCRF when Interpolating Covariance with Different Tabulated Steps along Full Force Model Reference Orbit.

Similarly to the observations of [1], it can be seen that the quadratic two-body blending generally outperforms the quintic keplerian interpolation for accurate interpolation of the position sigmas, especially for closer step between tabulated values, whereas the other method performs better when interpolating velocity sigmas.

Orbit interpolation

To better understand the effectiveness of blending when applied to orbit interpolation, similar plots to 10 and 11 were made. In addition, it was thought that the Brouwer-Lyddane and Heckstein-Hechler analytical models already available in Orekit could be used to improve the efficiency of orbit blending.

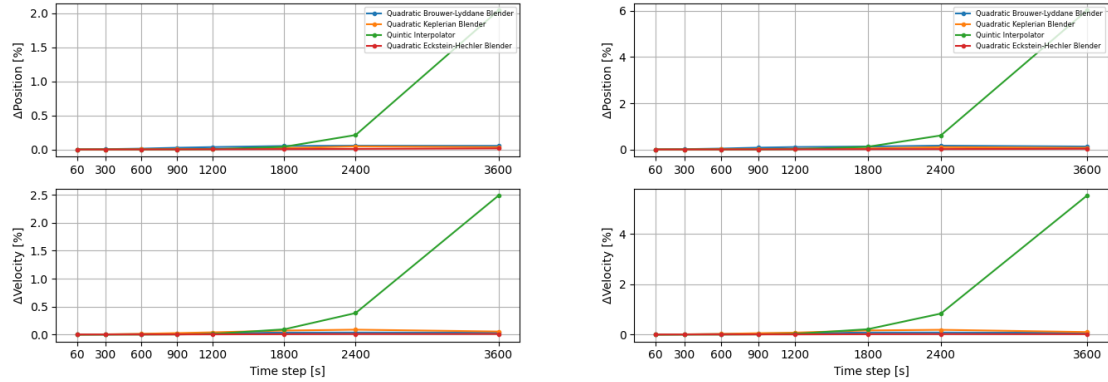


Figure 12: Average (left) and maximum (right) absolute relative position and velocity errors in GCRF when interpolating orbit with different tabulated steps in the Full Force Model environment

Starting from a time step of 1800s between tabulated values, Quintic Hermite interpolation begins to show an exponentially increasing error, whereas all blending methods errors remain relatively constant.

Time step range has then been limited from 60s to 1800s for plot readability.

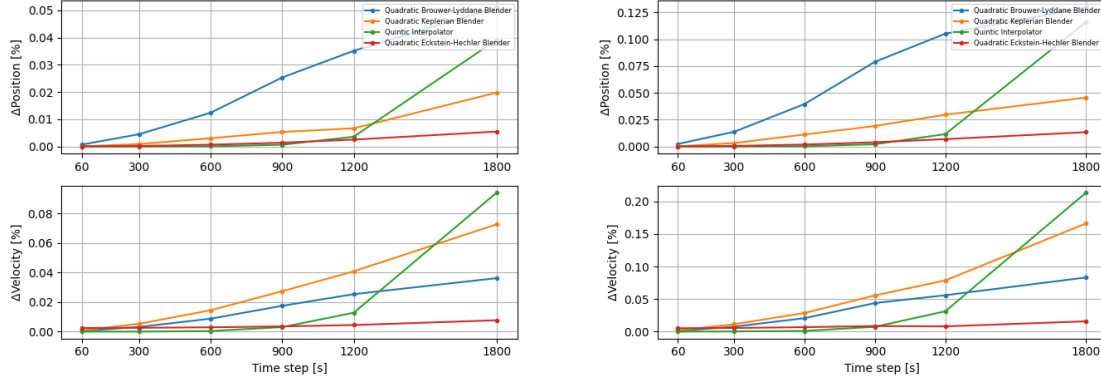


Figure 13: Average (left) and maximum (right) absolute relative position and velocity errors in GCRF when interpolating orbit with different tabulated steps in the Full Force Model environment

After reducing the time step range, it appears that the Quintic Hermite interpolator performs better both in position and velocity, while the time step between tabulated values is smaller than 15mn. This interpolator is then removed so that the other interpolators could be studied in wider time step range.

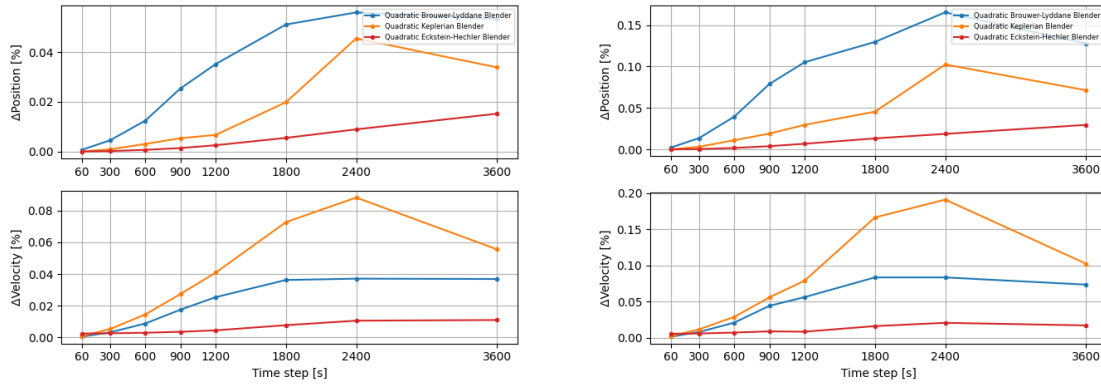


Figure 14: Average (left) and maximum (right) absolute relative position and velocity errors in GCRF when interpolating orbit with different tabulated steps in the Full Force Model environment

It can be seen in 14 that the Eckstein-Hechler blending is more precise both in position and velocity interpolation, while Keplerian blending is more appropriate for position interpolation than Brouwer-Lyddane blending and vice-versa for the velocity.

However, it is reminded to the reader that these results only apply regarding this specific test case as the Eckstein-Hechler is only suited for near circular orbits ($e < 0.1$) inclination neither equatorial (direct or retrograde) nor critical (direct or retrograde).

Conclusion

This paper presented the results obtained with blending and interpolating approaches presented in [1] for computing orbits and covariances between tabulated values. A test case, very similar to the one in [1], has been used and show that both the covariance quadratic two-body blending and quintic keplerian interpolation methods give very satisfying relative RMS of position and velocity sigmas errors, with 40mn steps, of around 0.25% and 0.12%.

Given that the quadratic two-body blending approach preserves the positive definiteness of the covariance matrix and that the usual step between tabulated values should be much smaller than 40mn, this method is the recommended one to the user.

Furthermore, the concept of blending applied to orbit interpolation has also been investigated and it has been shown that, on this specific test case, Quintic Hermite interpolation errors in position and velocity were increasing exponentially when the time step between tabulated values was larger than 15 mn. On the other hand, blending errors using Keplerian, Brouwer-Lyddane and Eckstein-Hechler model were increasing linearly and remained especially precise at large time steps with maximum position/velocity errors of respectively 0.10%/0.19%, 0.17%/0.08% and 0.03%/0.02% when using steps of 40mn. For comparison, using Quintic Hermite interpolation for this time step resulted in a maximum position/velocity of 0.5%/1%. However, further investigation should be done to understand why blending using the Brouwer-Lyddane model is inferior to Keplerian blending in position.

Hence, a combination of orbit quintic Hermite interpolation and covariance quadratic Two-Body blending is recommended to the user when interpolating covariances in a sample where the time step between tabulated values is below a few minutes. A combination of the best compatible analytical model ⁴ for quadratic orbit and covariance Two-Body blending would be suggested when the time step between tabulated values becomes more significant (>10 mn).

References

- [1] Sergei Tanygin. “Efficient covariance interpolation using blending of approximate covariance propagations”. In: *The Journal of the Astronautical Sciences* 61.1 (2014), pp. 107–132.
- [2] David A Vallado. *Fundamentals of astrodynamics and applications*. 4th ed. Microcosm Press, 2013.
- [3] Salvatore Alfano. “Orbital covariance interpolation”. In: (2004).
- [4] David A Vallado. “Covariance transformations for satellite flight dynamics operations”. In: (2004).
- [5] Luc Maisonobe, Véronique Pommier, and Pascal Parraud. “Orekit: An open source library for operational flight dynamics applications”. In: *4th International Conference on Astrodynamics Tools and Techniques*. 2010, pp. 3–6.
- [6] Philip Knocke, J Ries, and B Tapley. “Earth radiation pressure effects on satellites”. In: *Astrodynamics conference*. 1988, p. 4292.
- [7] Oliver Montenbruck, Eberhard Gill, and Fh Lutze. “Satellite orbits: models, methods, and applications”. In: *Appl. Mech. Rev.* 55.2 (2002), B27–B28.

⁴Eckstein-Hechler when compatible with current orbit. Otherwise, Keplerian for position oriented application and Brouwer-Lyddane for velocity oriented application