

COMPARING CONSIDER-COVARIANCE ANALYSIS WITH SIGMA-POINT CONSIDER FILTER AND LINEAR-THEORY CONSIDER FILTER FORMULATIONS[†]

Dr. Michael E. Lisano

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

Introduction

Recent literature in applied estimation theory reflects growing interest in the sigma-point (also called “unscented”) formulation for optimal sequential state estimation, often describing performance comparisons with extended Kalman filters as applied to specific dynamical problems [c.f. 1, 2, 3]. Favorable attributes of sigma-point filters are described as including a lower expected error for nonlinear – even non-differentiable – dynamical systems, and a straightforward formulation not requiring derivation or implementation of any partial derivative Jacobian matrices. These attributes are particularly attractive, e.g. in terms of enabling simplified code architecture and streamlined testing, in the formulation of estimators for nonlinear spaceflight mechanics systems, such as filter software onboard deep-space robotic spacecraft.

As presented in [4], the Sigma-Point Consider Filter (SPCF) algorithm extends the sigma-point filter algorithm to the problem of consider covariance analysis. Considering parameters in a dynamical system, while estimating its state, provides an upper bound on the estimated state covariance, which is viewed as a conservative approach to designing estimators for problems of general guidance, navigation and control. This is because, whether a parameter in the system model is observable or not, error in the knowledge of the value of a non-estimated parameter will increase the actual uncertainty of the estimated state of the system beyond the level formally indicated by the covariance of an estimator that neglects errors or uncertainty in that parameter. The equations for SPCF covariance evolution are obtained in a fashion similar to the derivation approach taken with standard (i.e. linearized or extended) consider-parameterized Kalman filters (c.f. [5]).

While in [4] the SPCF and linear-theory consider filter (LTCF) were applied to an illustrative linear dynamics/linear measurement problem, in the present work examines the SPCF as applied to nonlinear sequential consider covariance analysis, i.e. in the presence of nonlinear dynamics and nonlinear measurements. A simple SPCF for orbit determination, exemplifying an algorithm hosted in the guidance, navigation and control (GN&C) computer processor of a hypothetical robotic spacecraft, was implemented, and compared with an identically-parameterized (standard) extended, consider-parameterized Kalman filter. The onboard filtering scenario examined is a hypothetical spacecraft orbit about a small natural body with imperfectly-known mass. The formulations, relative complexities, and performances of the filters are compared and discussed.

Review of Sigma-Point Consider Filter Algorithm

This paper dispenses with providing formulation of the standard, linear-theory sequential state estimation algorithm, i.e. “extended Kalman filter”, given the numerous references that elaborate on this algorithm, such as the text by Tapley, Schutz and Born [5], as well as the text by Gelb [6] or Crassidis and Junkins [7], among many other excellent sources. Moreover, the reference by Tapley et al provides a chapter uniquely discussing the formulation and usage of consider parameters in a sequential, linear-theory filter – what is termed in this paper a “Linear-Theory Consider Filter” or LTCF. A comparison of the mathematical formulation of the LTCF with that of the SPCF is also given by Lisano [4], and is not repeated here.

[†] The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The SPCF algorithm discussed here was developed to enable sigma-point filters to be used in conditions when a conservative, consider-analysis approach to estimation is needed. As discussed in [4], the SPCF algorithm is obtained in the same manner of partitioning the filter state vector into estimated and considered parameters as used by Tapley, Schutz and Born [5] to obtain the LTCF, but proceeds along the lines of the derivative-free, Sigma-Point Kalman Filter algorithm given in such sources as [3].

The SPCF algorithm is as follows: a $(p \times 1)$ list \mathbf{C} of constant, non-estimated parameters whose errors are to be considered augments the $(n \times 1)$ estimated parameter list \mathbf{X}_{est} , to form a partitioned consider state vector, \mathbf{X}_{cons} :

$$\mathbf{X}_{\text{cons}} \equiv \begin{bmatrix} \mathbf{X}_{\text{est}} \\ \mathbf{C} \end{bmatrix} \quad \begin{matrix} (n \times 1) \\ (p \times 1) \end{matrix} \quad (1)$$

As with the linear-theory consider filter derived in [2], the associated consider covariance \mathbf{P}_{cons} is partitioned according to \mathbf{X}_{cons} , i.e.

$$\mathbf{P}_{\text{cons}} \equiv \begin{bmatrix} \mathbf{P}_{\text{cons}(n \times n)}^{xx} & \mathbf{P}_{(n \times p)}^{xc} \\ \mathbf{P}_{(n \times p)}^{cx} = (\mathbf{P}_{(p \times p)}^{cc})^T & \mathbf{P}_{(p \times p)}^{cc} \end{bmatrix} \quad (2)$$

Now, adapting the UKF formulation approach to computing a-priori sigma points in an unscented filter, first we factorize \mathbf{P}_{cons} (best estimate of covariance from previous epoch t_{k-1}) into $\mathbf{S}_{\text{cons},k-1}(\mathbf{S}_{\text{cons},k-1})^T$. In order to preserve the constancy of \mathbf{P}^{cc} , the lower-triangular square matrix $\mathbf{S}_{\text{cons},k-1}$ is computed using a block-Cholesky decomposition⁷, with the initial pivot starting in the \mathbf{P}^{cc} partition.

$$\mathbf{S}_{\text{cons},k-1} = \begin{bmatrix} \mathbf{S}_{\text{cons},k-1}^{xx} & \mathbf{0}_{(n \times p)} \\ \mathbf{S}_{k-1}^{cx} & \mathbf{S}_{k-1}^{cc} \end{bmatrix} \quad (3)$$

where

$$\mathbf{S}_{k-1(p \times p)}^{cc} \equiv \sqrt{(\lambda_c + p)\mathbf{P}^{cc}} = \text{const.} \quad (4)$$

$$\mathbf{S}_{k-1(p \times n)}^{cx} \equiv (\lambda_c + p) \left(\mathbf{S}_{k-1}^{cc} \right)^{-1} \mathbf{P}_{k-1}^{cx}, \text{ and} \quad (5)$$

$$\mathbf{S}_{k-1(n \times n)}^{xx} \equiv \sqrt{(\lambda_x + n)\mathbf{P}_{\text{cons},k-1}^{xx} - \left(\mathbf{S}_{k-1}^{cx} \right)^T \mathbf{S}_{k-1}^{cx}}. \quad (6)$$

and in which $\lambda_c = 3 - p$, and $\lambda_x = 3 - n$. Equation (5) for \mathbf{S}^{cx} will always be analytic, because sub-matrix \mathbf{S}^{cc} is a square root of constant positive definite matrix \mathbf{P}^{cc} .

As with the linear-theory consider algorithm, the sigma-point consider filter requires that \mathbf{X}_{est} and \mathbf{P}^{xx} be estimated with the standard sequential estimator algorithm, in this case the Sigma-Point Kalman Filter algorithm (also called the ‘‘Unscented Kalman Filter’’ in some references). Reference [4] details the standard (i.e. non-consider) Sigma-Point Filter algorithm, using the same symbology as this paper.

Now, in addition to estimating an n^{th} order vector of states, to compute the additional uncertainty from consider the non-estimating parameters, at epoch t_{k-1} we generate $(2p + 1)$ consider sigma-points, which are a set¹ of samples of system state realizations at its a-priori expected value and also at $2p$ select points on the uncertainty ellipsoid:

$$\mathbf{X}_{\text{cons},k-1,i} = \mathbf{X}_{k-1}^+ + \sigma_{\text{cons},k-1,i}, \quad i = 0, 1, \dots, 2p \quad (7)$$

¹ For parameter list $\mathbf{X} \in \mathbb{R}^L$ belonging to a multivariate probability distribution, $2L + 1$ samples is the minimum needed to capture the first two moments (expected value and covariance) of the distribution.

where $\sigma_{\text{cons},k-1,0} = 0$, and $\sigma_{\text{cons},k-1,i} = \pm \text{columns } (n+1) \text{ through } (n+p) \text{ of } (S_{\text{cons},k-1})^T$, containing the square-root of the dispersed values of consider parameter list \mathbf{C} plus expected variations in a-priori estimated parameter list $\mathbf{X}_{\text{est},k-1}$ due to cross-correlations with \mathbf{C} .

These $(2p+1)$ consider sigma-points are then propagated from epoch t_{k-1} to epoch t_k , using the $((n+p) \times 1)$ vector of nonlinear expressions for the time-evolution of the augmented state vector, e.g. the equations of motion for the dynamical system, augmented with a $(p \times 1)$ list of zeros to maintain the constancy of the consider parameters. In this manner, we obtain the $(2p+1)$ predicted sigma-points at epoch t_k :

$$\chi_{\text{cons},k,i} = f(t_{k-1}, t_k, \chi_{\text{cons},k-1,i}, \mathbf{u}_{k-1}) \quad i = 0, 1, \dots, 2p \quad (8)$$

Per the sigma-point Kalman filter formulation, the predicted mean consider state is computed using the predicted sigma-points as:

$$\mathbf{X}_{\text{cons},k}^- = \frac{1}{p + \lambda_c} \left(\lambda_c \chi_{\text{cons},k,0} + \frac{1}{2} \sum_{i=1}^{2p} \chi_{\text{cons},k,i} \right) \quad (9)$$

Then, the predicted consider covariance is computed as:

$$\mathbf{P}_{\text{cons},k}^- = \begin{bmatrix} \mathbf{P}_{\text{cons},k}^{xx-} & \mathbf{P}_k^{xc-} \\ \mathbf{P}_k^{cx-} & \mathbf{P}^{cc} \end{bmatrix} = \frac{1}{p + \lambda_c} \left\{ \lambda_c (\chi_{\text{cons},k,0} - \mathbf{X}_{\text{cons},k}^-) (\chi_{\text{cons},k,0} - \mathbf{X}_{\text{cons},k}^-)^T + \frac{1}{2} \sum_{i=1}^{2p} (\chi_{\text{cons},k,i} - \mathbf{X}_{\text{cons},k}^-) (\chi_{\text{cons},k,i} - \mathbf{X}_{\text{cons},k}^-)^T \right\} \quad (10)$$

The primary utility of matrix $\mathbf{P}_{\text{cons},k}^-$ is that it yields the $(n \times p)$ sub-matrix \mathbf{P}_k^{xc-} of predicted cross-correlations between consider and estimated parameters. At epochs for which measurements are available, the change due to the measurement information on the cross-correlation sub-matrix is calculated the following way:

$$\mathbf{P}_k^{xc+} = \mathbf{P}_k^{xc-} - \mathbf{K}_k (\mathbf{P}_k^{xy})^T (\mathbf{P}_k^{xx-})^{-1} \mathbf{P}_k^{xc-} - \mathbf{K}_k (\mathbf{P}_k^{cy})^T \quad (11)$$

in which \mathbf{P}_k^{cy} is the cross-correlation matrix between consider-parameter space and measurement space, computed by dispersing the measurement model equation on the $2p+1$ consider-parameter sigma-points and forming a discrete covariance matrix in the same manner as one obtains \mathbf{P}_k^{xy} (c.f. sigma-point filter algorithm description and example in [4]).

The optimal gain \mathbf{K}_k , residual covariance \mathbf{P}_k^{vv} and a-priori state covariance \mathbf{P}_k^{xx-} are all obtained from a standard sigma-point filter (that estimates \mathbf{X}_{est} and neglects errors in \mathbf{C}) operating simultaneously in time with the consider filter algorithm. Finally, the additive uncertainty from considering non-estimated parameters is calculated as:

$$d\mathbf{P}_{\text{cons},k} = \mathbf{P}_k^{xc+} (\mathbf{P}^{cc})^{-1} (\mathbf{P}_k^{xc+})^T \quad (12)$$

and the updated consider covariance matrix, post-measurement, for epoch t_k is given by:

$$P_{\text{cons},k} = \begin{bmatrix} P_{\text{cons}}^{\text{xx}+} & P^{\text{xc}+} \\ P^{\text{cx}+} & P^{\text{cc}} \end{bmatrix} \quad (13)$$

in which the state covariance augmented for consider-parameter uncertainty ($P_{\text{cons},k}^{\text{xx}+}$) is related to the standard state covariance $P_k^{\text{xx}+}$ by:

$$P_{\text{cons},k}^{\text{xx}+} = P_k^{\text{xx}+} + dP_{\text{cons},k} \quad (14)$$

If no observation is available to be processed at time t_k , Equations (12) and (13) are evaluated using $P_k^{\text{xx}-}$ and $P_k^{\text{xc}-}$ instead of $P_k^{\text{xx}+}$ and $P_k^{\text{xc}+}$. This algorithm is recursive, so that after calculating $P_{\text{cons},k}$, if filter processing is to continue, the time index k is set to $(k-1)$, and the steps above are repeated at the next processing epoch, until all measurements have been applied or the desired end time has been reached.

Nonlinear Filtering Scenario: Spacecraft Autonomously Reducing Periapsis at a Small Body

Next, an SPCF and an LTCF are compared, in a simple-but-representative example problem entailing autonomous nonlinear consider-estimation.

As depicted in Figure 1, a hypothetical spacecraft is in an initially-circular orbit about a body having a small mass (e.g. a “spherical asteroid” with radius $R_{\text{body}} = 10$ km and gravitational parameter $GM = 2.5 \times 10^{-10} \text{ GM}_{\text{earth}}$), at a current (apoapsis) altitude of 20 km. The one-way light travel time from Earth to this body is many minutes or even hours long, so the spacecraft is autonomously executing a maneuver to reduce its periapsis altitude h_p from 20 km to just 500 m, e.g. for a close imaging pass.

The onboard consider-covariance algorithm is used in two ways:

- (1) Part of the targeting periapsis-reduction maneuver calculation is to propagate to periapsis time not only the filter position and velocity of the spacecraft, but also the formal uncertainty in position and velocity, considering the error in the value of its imprecisely-known GM (chosen as 7% error for this study) for the asteroid. The periapsis-reduction maneuver will be autonomously flagged as hazardous, and tentatively cancelled, if the predicted position uncertainty at periapsis time exceeds a safety threshold (chosen here to be 200 m radial, 1000 m downtrack, 3-sigma).
- (2) Also, the spacecraft carries a limb sensor that measures the angle δ_{limb} between the local negative vertical direction, and the limb of the spherical small body (with 1 arcsecond 1-sigma limb-locating accuracy). The onboard consider covariance filter logic can process this data, and can also determine, predictively, how much the radial orbit uncertainty would improve if a limb angle measurement were scheduled to happen some pre-selected time (chosen here to be 9000 s) prior to periapsis, to assess the utility of a mid-course corrective maneuver, post-limb-measurement, in a “go/no-go” decision for this hazardous autonomous maneuver.

So, if the limb-measurement-based uncertainty can be quickly (say, within a few wall-clock seconds) determined to comply with the safety threshold, the hazard-flagged periapsis-reduction maneuver can be executed anyway, and a mid-course limb-measurement/clean-up maneuver is autonomously scheduled. Otherwise, the periapsis-reduction maneuver remains hazard-flagged and completely cancelled, and effort is made to improve the onboard navigation state for another orbit revolution, before trying the periapsis-lower maneuver again.

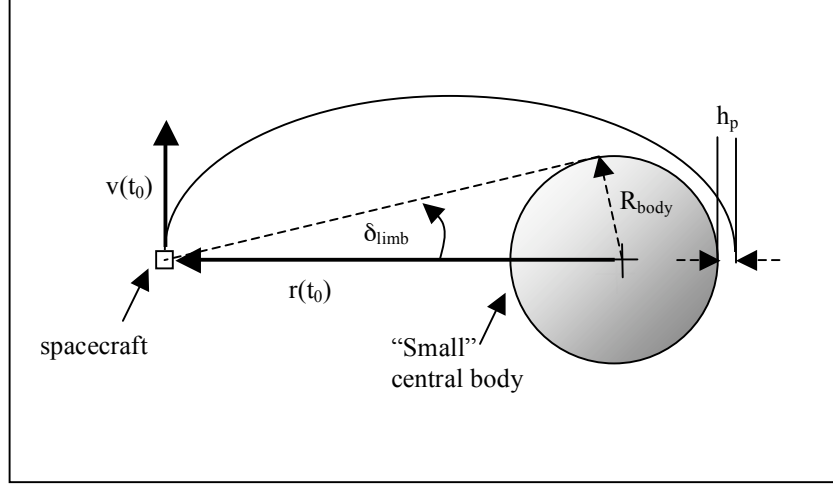


Figure 1: Spacecraft with limb sensor, in elliptical orbit about a small central body with imprecisely-known gravitational parameter GM_{body} .

Continuing with this motivating example, in order to rapidly predict a future translational state of the spacecraft on the post-maneuver descent trajectory (and also simplify the comparison of sigma-point and linear-theory consider filters), the system dynamics are Keplerian orbits, neglecting perturbations such as 3rd body effects (e.g. from the Sun’s gravity), higher-order gravity from the small body, and solar radiation pressure effects are all neglected. In both filter instances, the position and velocity states of the spacecraft are precisely and quickly propagated in time by solving Kepler’s equation for an unperturbed conic.

Filter Formulation and Implementation Aspects

The state vector, including consider parameter GM, is:

$$X^T = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ | \ GM]$$

The non-linear equations of two-body motion, used in both the SPCF and the LTCF, are:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ -GMx/r^3 \\ -GMy/r^3 \\ -GMz/r^3 \\ \hline 0 \end{bmatrix}, \quad r = \sqrt{x^2 + y^2 + z^2}$$

The non-linear equation for the limb angle measurement as a function of state variables, also used in both the SPCF and the LTCF, is:

$$\delta_{\text{limb}} = \sin^{-1} \left(R_{\text{body}} / (x^2 + y^2 + z^2)^{1/2} \right) \quad (16)$$

In the LTCF only, the partial derivatives of the limb angle measurement model with respect to the state vector, including the consider parameter, are:

$$H(t_i) = \begin{bmatrix} h1 & h2 & h3 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

where:

$$\begin{aligned} h1 &= -x R_{\text{body}} / \left\{ (x^2 + y^2 + z^2)^3 (1 - R_{\text{body}} / (x^2 + y^2 + z^2)^{1/2})^{1/2} \right\} \\ h2 &= -y R_{\text{body}} / \left\{ (x^2 + y^2 + z^2)^3 (1 - R_{\text{body}} / (x^2 + y^2 + z^2)^{1/2})^{1/2} \right\} \\ h3 &= -z R_{\text{body}} / \left\{ (x^2 + y^2 + z^2)^3 (1 - R_{\text{body}} / (x^2 + y^2 + z^2)^{1/2})^{1/2} \right\} \end{aligned}$$

Also in the LTCF only, the state transition matrix is propagated using the partial derivatives of the equations of motion with respect to the state vector parameters (including the consider parameter), also called the “plant” matrix, which is:

$$A(X(t), t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ f11 & f12 & f13 & 0 & 0 & 0 & fg1 \\ f21 & f22 & f23 & 0 & 0 & 0 & fg2 \\ f31 & f32 & f33 & 0 & 0 & 0 & fg3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

where:

$$\begin{aligned} f11 &= -GM_{\text{body}} / (x^2 + y^2 + z^2)^{3/2} + 3GM_{\text{body}} x^2 / (x^2 + y^2 + z^2)^{5/2} \\ f12 &= 3GM_{\text{body}} x y / (x^2 + y^2 + z^2)^{5/2} \\ f13 &= 3GM_{\text{body}} x z / (x^2 + y^2 + z^2)^{5/2} \\ f21 &= f12 \\ f22 &= -GM_{\text{body}} / (x^2 + y^2 + z^2)^{3/2} + 3GM_{\text{body}} y^2 / (x^2 + y^2 + z^2)^{5/2} \\ f23 &= 3GM_{\text{body}} y z / (x^2 + y^2 + z^2)^{5/2} \\ f31 &= f13 \\ f32 &= f23 \\ f33 &= -GM_{\text{body}} / (x^2 + y^2 + z^2)^{3/2} + 3GM_{\text{body}} z^2 / (x^2 + y^2 + z^2)^{5/2} \end{aligned}$$

Two versions of the LTCF were implemented, which differed in the manner that the state transition matrix $\Phi(t, t_0)$, was propagated. These two versions had different accuracies in the propagation of the LTCF covariance, because for linear-theory filters, $P(t) = \Phi(t, t_0)P_0\Phi^T(t, t_0) + Q(t)$, where Q is process noise. (Process noise has been set to zero in the present example problem.)

One LTCF version utilized an approximation for state transition propagation, valid for small time-steps and very commonly used in real-time/embedded filter codes, i.e.

$$\Phi(t, t_0) \approx I + A(t_0) \times (t - t_0), \text{ where } I \text{ is the identity matrix.} \quad (19)$$

The other LTCF version integrated the state transition in a fourth-order Runge-Kutta integrator, enabling better accuracies for longer time-steps than the approximate state-transition version, i.e.

$$\Phi(t, t_0) = \int_{t_0}^t \Phi(\tau, t_0) A(X(\tau), \tau) d\tau, \quad \Phi(t_0, t_0) = I \quad (20)$$

To facilitate a direct comparison of implementation aspects, as well as a comparison of runtime performance and accuracy, the SPCF and both LTCF versions were developed in nearly identical fashion, using generic sequential consider-estimation filtering code that provides a uniform application programming interface for sigma-point and linear-theory sequential filter types. All filter types require a-priori time, state, and covariance, as well as process noise and measurement noise matrices, as initialization inputs.

The sigma-point filter interface requires that a user supply the following code:

- (1) a routine to calculate the state at time t , $X(t) = f(t, t_0, X_0, u)$, a function of time t , initial time t_0 , initial state X_0 , and control inputs u , if any, and
- (2) a routine to calculate the measurement model $h(t, X(t))$

The LTCF interface requires the same two input routines as the SPCF interface, plus the following additional user-supplied code:

- (3) a routine to calculate the state transition matrix, $\Phi(t, t_0)$ and
- (4) a routine to calculate the measurement partial derivative matrix $H(t, X(t))$.

As a general note about implementation differences, any linear theory filter (e.g. linear or extended Kalman filter, or LTCF) will require more time and effort to derive, develop, debug, document and maintain than its sigma-point filter counterpart (e.g. unscented Kalman filter or SPCF), given identical parameter spaces, and models of dynamics and measurements, due to the requirement for partial derivatives in the LTCF.

As an example of the relative implementation complexities between the SPCF and LTCF, in development and debugging of the two LTCF versions for this example problem, additional debugging effort was required over the SPCF, to ensure consistency of units (of GM, as well as the position and velocity state variables) between the four matrices f , h , Φ , and H . Conversely, unit consistency needed to be confirmed only between f and h for the SPCF, yielding lower effort to debug and verify the code, with significantly reduced potential for implementation error. This reduction in implementation and verification complexity scales with the dimensionality of the filtering problem being solved.

Results, and Comparison of SPCF and LTCF Performance

In principal, the SPCF algorithm step-size should be limited only by the stepsize required to propagate the state vector (i.e. propagate the equations of motion) with sufficient accuracy. In this simple example problem, the best method to effect this propagation is to compute the Keplerian orbit elements and use Kepler's equation to rapidly predict the state vector for an arbitrarily-large time step. The LTCF can also have its equations of motion propagated in this manner, but as is shown below, covariance propagation considerations necessitate a smaller step-size for the LTCF.

The 3-sigma predicted uncertainties in position and velocity (in radial, downtrack, crosstrack coordinates) from covariance, for the example periapsis lower maneuver problem, are plotted in Figures 2 and 3. In these plots, the results are seen of estimating position and velocity while neglecting a 0.7% error in GM, and also while considering the 0.7% GM error, which show covariances from a six-state (position and velocity) sigma-point Kalman filter as well as the seven-state SPCF described in the preceding section.

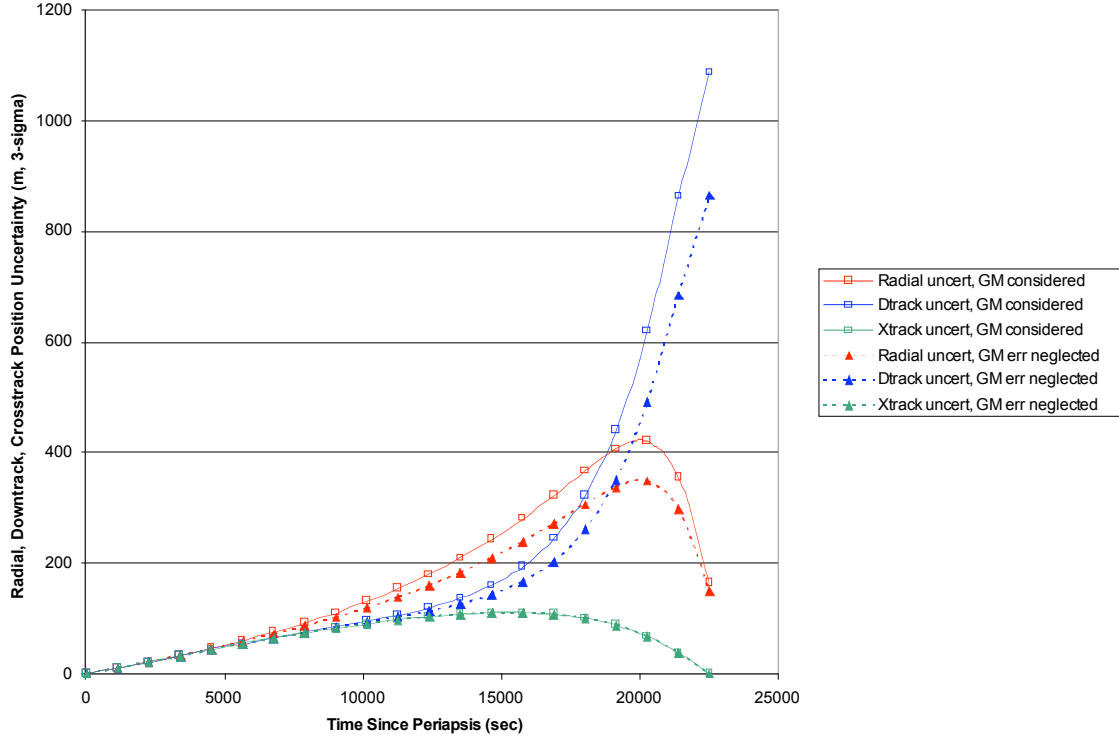


Figure 2: Position Uncertainty (3-Sigma) from SPCF for Keplerian Covariance, No Limb Observations Processed, with 0.7% Gravity Error (3-Sigma) Considered, Compared with Neglecting Gravity Error

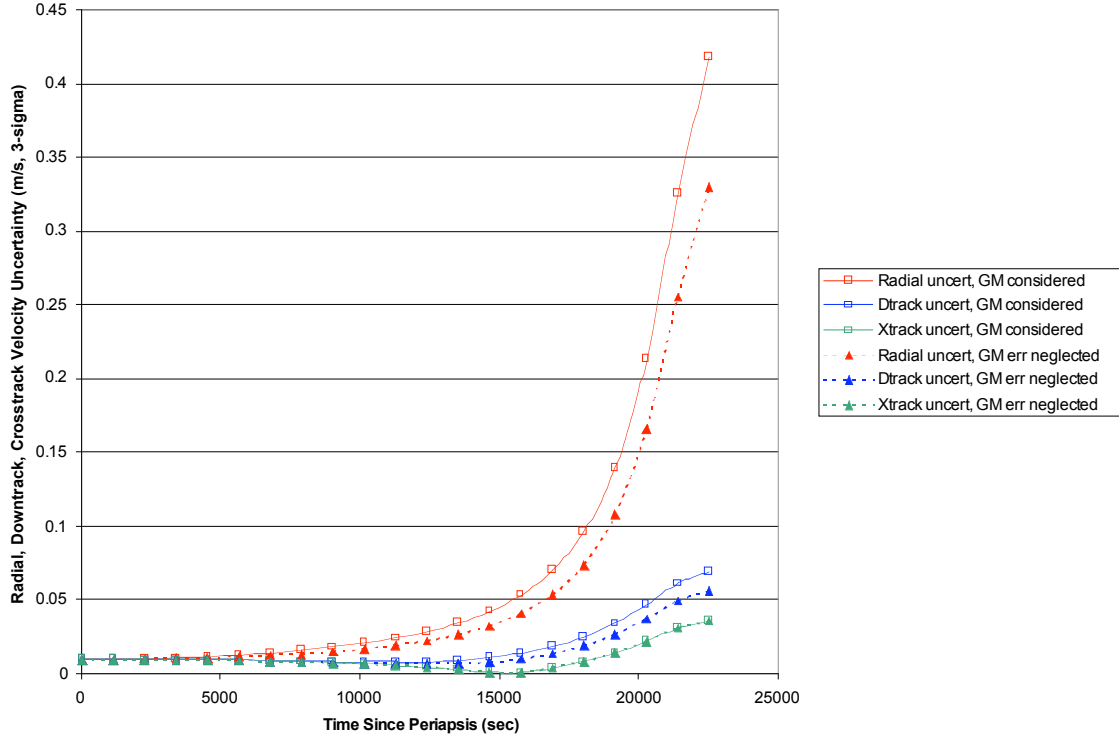


Figure 3: Velocity Uncertainty (3-Sigma) from SPCF for Keplerian Covariance, No Limb Observations Processed, with 0.7% Gravity Error (3-Sigma) Considered, Compared with Neglecting Gravity Error

The effect of considering 0.7% gravity model error on the propagated covariance is readily seen in Figures 1 and 2. In particular, downtrack position 3-sigma uncertainty of the propagated trajectory at periapsis time is approximately 220 m worse than the 870 m “neglect GM error” value, when the gravity model error is considered. The radial velocity 3-sigma uncertainty also increases significantly with GM error considered, by 9 cm/sec, versus the 33 cm/sec “neglect GM error” value. Note that for all plots shown in this paper, including Figures 2 and 3, the timestep for the SPCF covariance propagation was chosen to be 1/10 of the period of the reduced-periapsis orbit, or 4501 seconds².

The experimental method by which the SPCF and two LTCF variants were compared was straightforward. The LTCF covariances were propagated with the same 4501-second timestep as the SPCF, and subsequently the timestep of the two LTCF variants was reduced, to compare differences in output covariance vs. the SPCF covariance as a function of propagation timestep size. Also, overall “wall-clock” propagation time was compared, between the full-timestep SPCF (4501-second timestep) and the two LTCF variants.

The accuracy performances of the “approximate state transition” and “integrated state transition” versions of the LTCF, compared with that of the SPCF, for varying LTCF timestep sizes, are shown in Figures 4 and 5, respectively, as percentage differences in RSS position uncertainty with the SPCF position consider covariance. Figures 4 and 5 show that the periapsis-time (= 22508 seconds) consider-covariance position uncertainty values of the two LTCF versions approach the same values as the 4501-second SPCF as the LTCF propagation timesteps are made smaller and smaller.

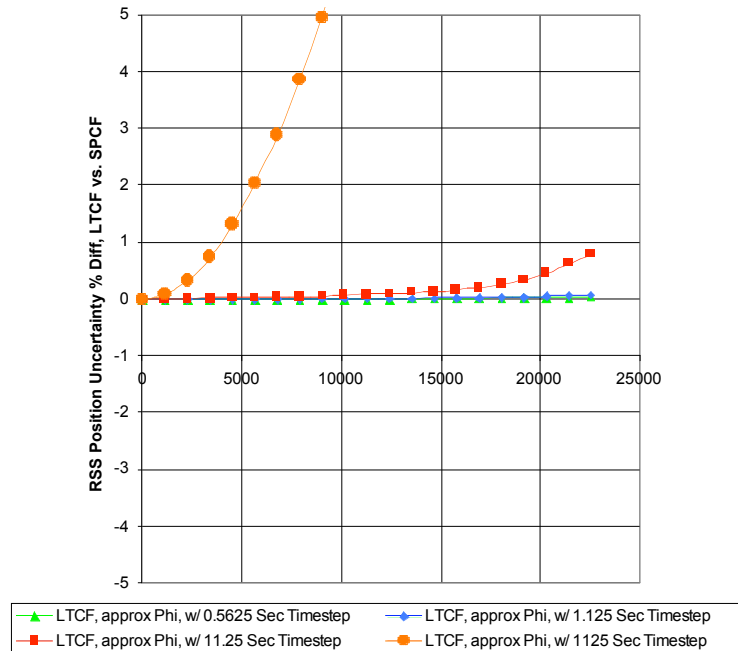


Figure 4: Percentage difference in RSS position uncertainty between LTCF (with approximate propagation of state transition), for various LTCF timestep sizes, and the 4501-second-timestep SPCF.

² It was experimentally determined for this problem that, even if the step-size is chosen to be as large as 1/6 of the orbit period, or 7503 seconds, the SPCF periapsis-time 1-sigma covariance solution was identical within 4 cm in RSS position uncertainty and 1 mm/sec in velocity uncertainty – with similar difference levels in periapsis-time filter output states - to the results obtained using a 4501-second timestep.

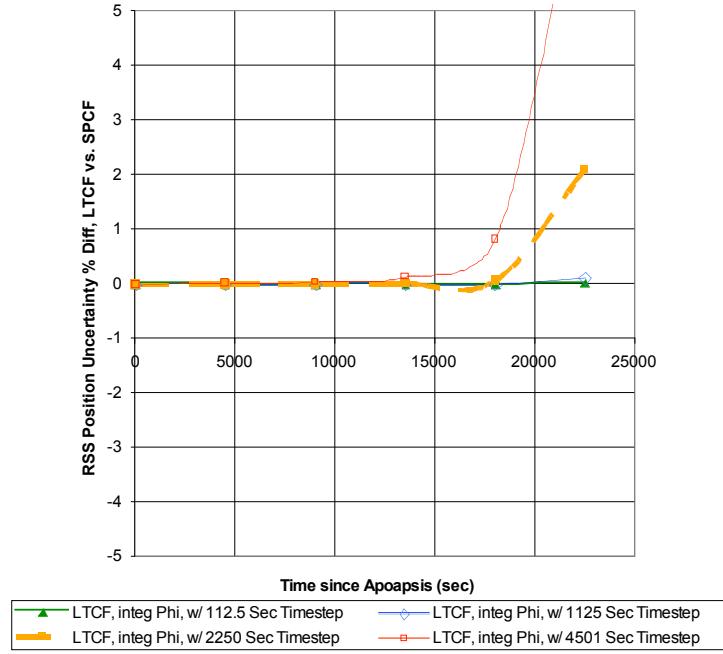


Figure 5: Percentage difference in RSS position uncertainty between LTCF (with integrated propagation of state transition differential equation) for various LTCF timestep sizes, and the 4501-second-stepsizes SPCF.

Reducing the timestep of the approximate-state-transition version of the LTCF to 11.25 seconds or smaller, the periapsis-time RSS position uncertainty is reduced to within 1 percent of the 4501-second-timestep SPCF solution (Figure 4). The integrated-state-transition LTCF, with timestep reduced to 1125 seconds or smaller, drops to well under 1 percent difference with the SPCF, although for twice this timestep (2250 seconds) it exceeds 2 percent difference with the SPCF.

The approximate average run-times of the SPCF and two LTCF variants were benchmarked by running the three filter codes, on the same non-network-shared computer processor, with start-time and end-time recorded using operating-system time-reporting commands, and with all input, output, initialization and post-processing codes excluded from the timing assessment. Each filter was configured to simulate covariance propagation from the initial time to periapsis time, using the largest propagation timestep that yielded a 1-percent or smaller RSS difference with the SPCF in periapsis-time position uncertainty. A thousand runs were made of each filter, and the run-times were averaged. Finally, the averaged run-times for the “approximate state transition” LTCF and the “integrated state transition” LTCF were normalized by the SPCF averaged run-time.

This procedure, in effect, yields for a specified accuracy level a normalized measure of mathematical operations performed for the different filters. The resulting run-time ratios described below for both LTCF versions are greater than 1.0, indicating that to achieve the same accuracy level for this example consideration problem, both versions of the LTCF underperform the SPCF in terms of run-time/throughput performance.

For the present example problem, to achieve periapsis-time position accuracy within 1 percent of the SPCF, the approximate-state-transition version of the LTCF requires a timestep of 11.25 seconds or smaller, or 400 times as many propagation steps as the SPCF. The run-time to propagate and update the covariance over a single time-step using approximate-state-transition LTCF was measured to be 0.245 times that of the SPCF. Thus, the overall approximate-state-transition LTCF runtime to compute the

periapsis-time covariance, with accuracy within 1 percent of the SPCF, was approximately 100 times longer than that of SPCF runtime.

Similarly, for this example problem, to achieve periapsis-time position accuracy within 1 percent of the SPCF, the integrated-state-transition version of the LTCF requires a timestep of 1800 seconds or smaller, or 2.5 times as many propagation steps as the SPCF. The run-time to propagate and update the covariance over a single time-step using integrated-state-transition LTCF was measured to be 0.49 times that of the SPCF. Thus, the overall integrated-state-transition LTCF runtime to compute the periapsis-time covariance, with accuracy within 1 percent of the SPCF, was approximately 1.25 times longer than that of SPCF runtime.

Summarizing the run-time comparison results, for the example nonlinear consider-estimation problem presented here, the integrated-state-transition LTCF has better performance at much larger timesteps than the approximate-state-transition LTCF, and neither version of the LTCF comes within 1 percent the accuracy performance of the SPCF, without incurring runtime/throughput penalties.

The run-time advantage of the SPCF over the LTCF is tempered for problems with lower degrees of nonlinearity, as in these instances the LTCF is able to more closely match the accuracy of the SPCF by taking larger timesteps. For example, if a problem is perfectly linear in its dynamics and measurement model, therefore having no requirement for small timesteps, the present results indicate that the approximate-state-transition LTCF would have much faster run-time than the equivalent SPCF or integrated-state-transition LTCF. Hence, for any nonlinear consider-estimation application where runtime performance is important, an accuracy-and-runtime comparative evaluation should be made similar to the one presented here, to ascertain whether the SPCF or one of the LTCF is better for that particular problem.

The final aspect of consider-filter comparison that is treated here is the accuracy performance of the two LTCF versions and the SPCF, in the presence of a single limb measurement incorporated 9000 seconds prior to periapsis time. Incorporating the limb measurement is seen to reduce the consider-covariance uncertainty in all in-orbit-plane position and velocity components, especially the downtrack position and radial velocity, by 150 m and 7 cm/sec, respectively. The 3-sigma position and velocity consider uncertainties from the approximate-state-transition LTCF are shown in Figures 6 and 7; those of the integrated-state-transition LTCF are shown in Figures 8 and 9; and those of the SPCF are shown in Figures 10 and 11. In each of these figures, the covariance values are shown for several values of propagation timestep size (4501 sec, 1125 sec, 112.5 sec, 11.25 sec, and 1.125 sec).

As seen in Figures 6 through 11, for all of the filters here, the accuracy of the periapsis-time covariance can be controlled by using sufficiently-small propagation timesteps. In Figures 6 and 7, the periapsis-time accuracy of the approximate-state-transition LTCF is seen to settle within a few meters and mm/sec of the same solution for timestep size $dt = 11.25$ seconds or smaller – larger values of dt than this yield very large errors in the periapsis-time covariance, which would be deleterious for the example spacecraft application chosen here.

In Figures 8 and 9, the periapsis-time accuracy of the integrated-state-transition LTCF is seen to settle within a few meters and mm/sec of the same solution for timestep size $dt = 1125$ seconds or smaller. The propagation of the covariance using a rigorously-integrated state transition matrix does yield significant improvement over the approximate-state-transition approach. Nevertheless, due to the mathematical limitations of using a linear-theory estimator on an inherently nonlinear estimation problem, it is seen that larger values of dt than 1125 seconds still yields very large errors in the periapsis-time covariance.

Finally, in Figures 10 and 11, the periapsis-time accuracy of the integrated-state-transition LTCF is seen to settle within several meters and mm/sec of the same solution for all timestep sizes $dt = 4501$ seconds and smaller. The improvement over the integrated-state-transition LTCF approach for the larger stepsizes is because the mathematical limitations of using a linear-theory covariance propagation approach (i.e. the use of a state transition matrix, a linear-theory construct) on an inherently nonlinear estimation problem is not present for the SPCF, when applied to this example problem.

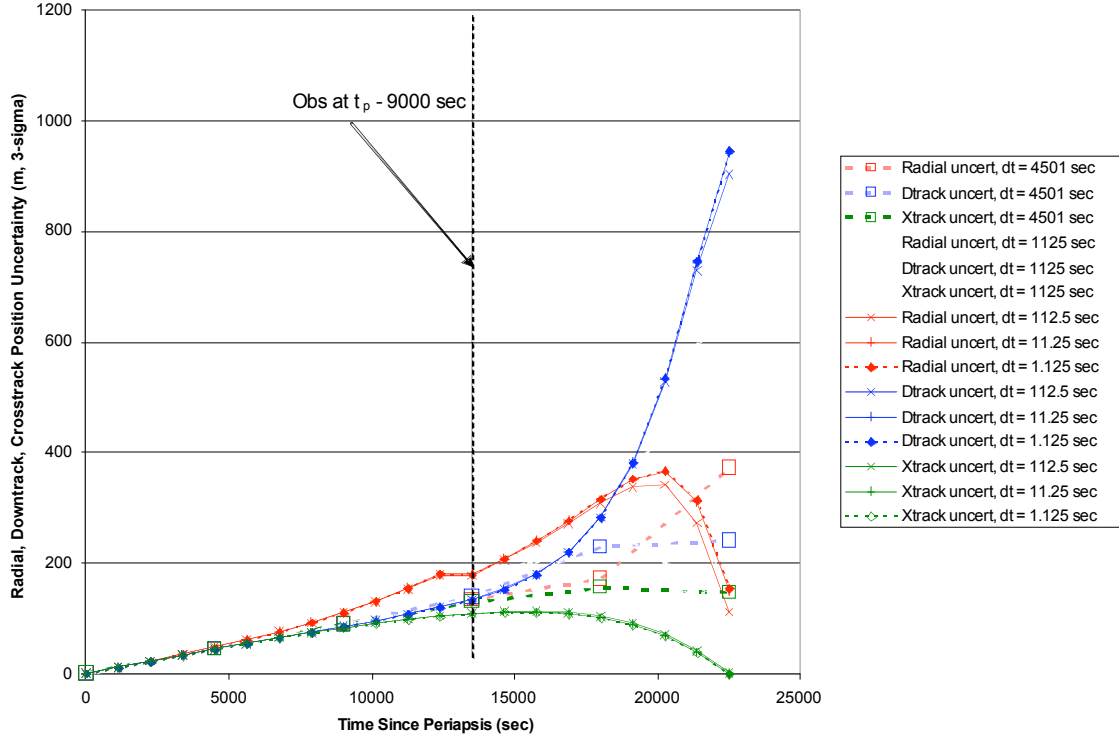


Figure 6: Stepsize-sensitivity of position uncertainty (3-sigma) from LTCF (approximate state transition) with 0.7% gravity error considered, incorporating one limb angle observation at periapsis minus 9000 sec.

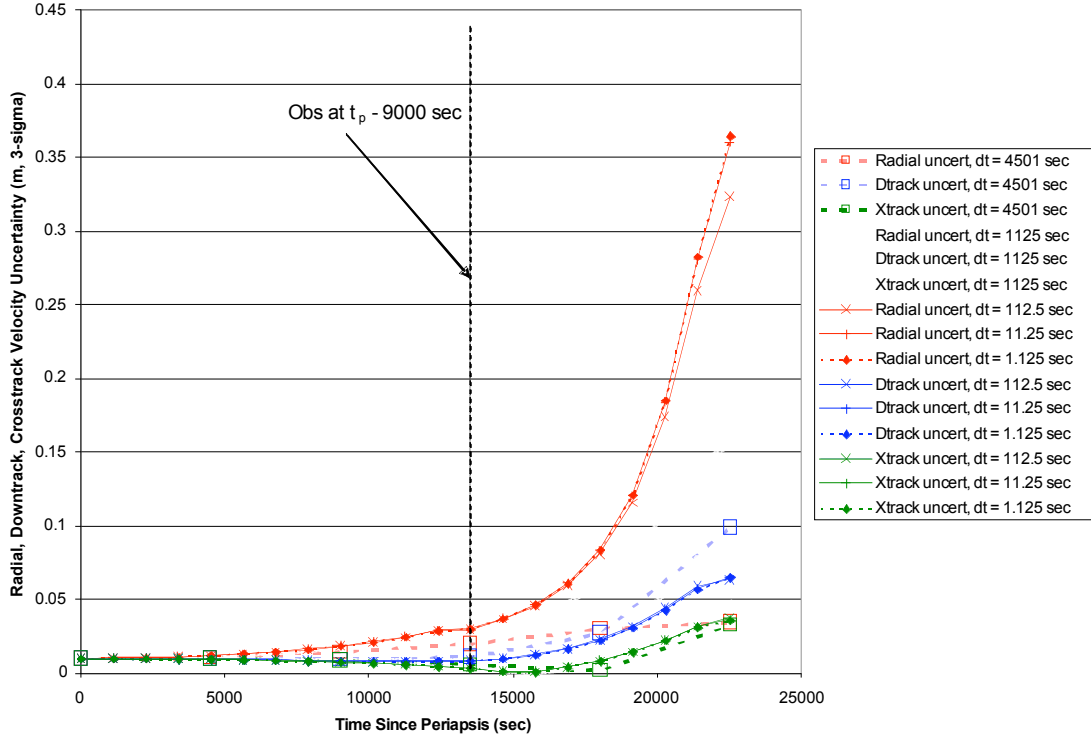


Figure 7: Stepsize-sensitivity of velocity uncertainty (3-sigma) from LTCF (approximate state transition) with 0.7% gravity error considered, incorporating one limb angle observation at periapsis minus 9000 sec.

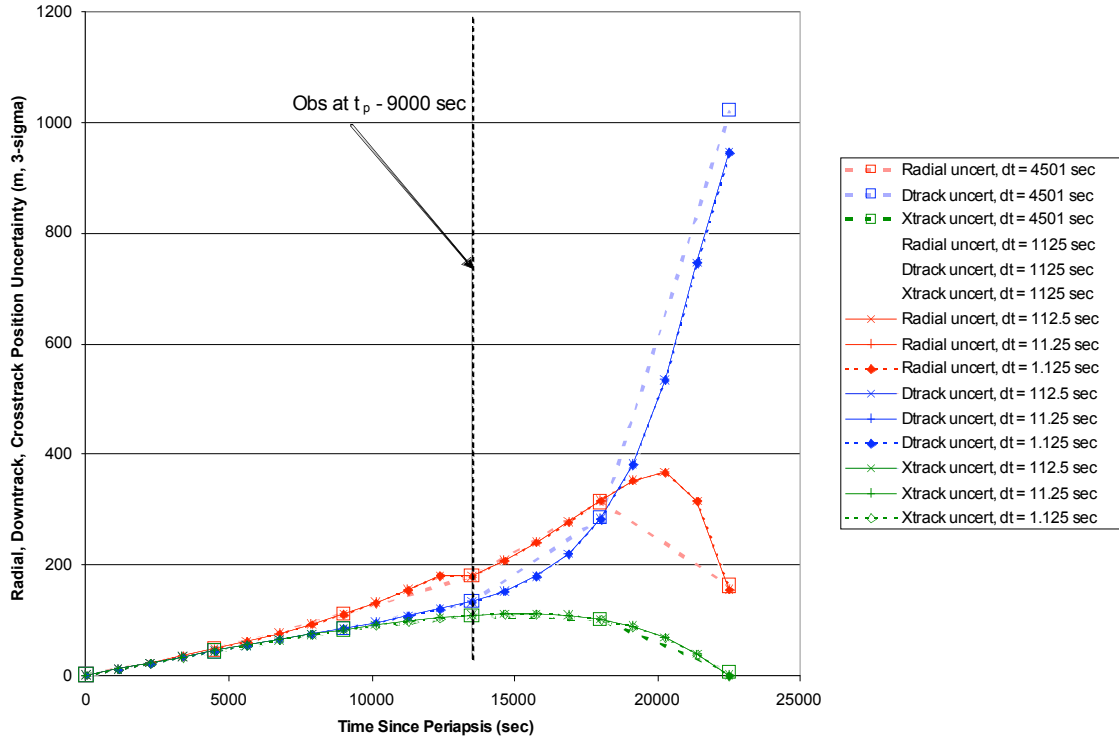


Figure 8: Stepsize-sensitivity of position uncertainty (3-sigma) from LTCF (integrated state transition) with 0.7% gravity error considered, incorporating one limb angle observation at periapsis minus 9000 sec.

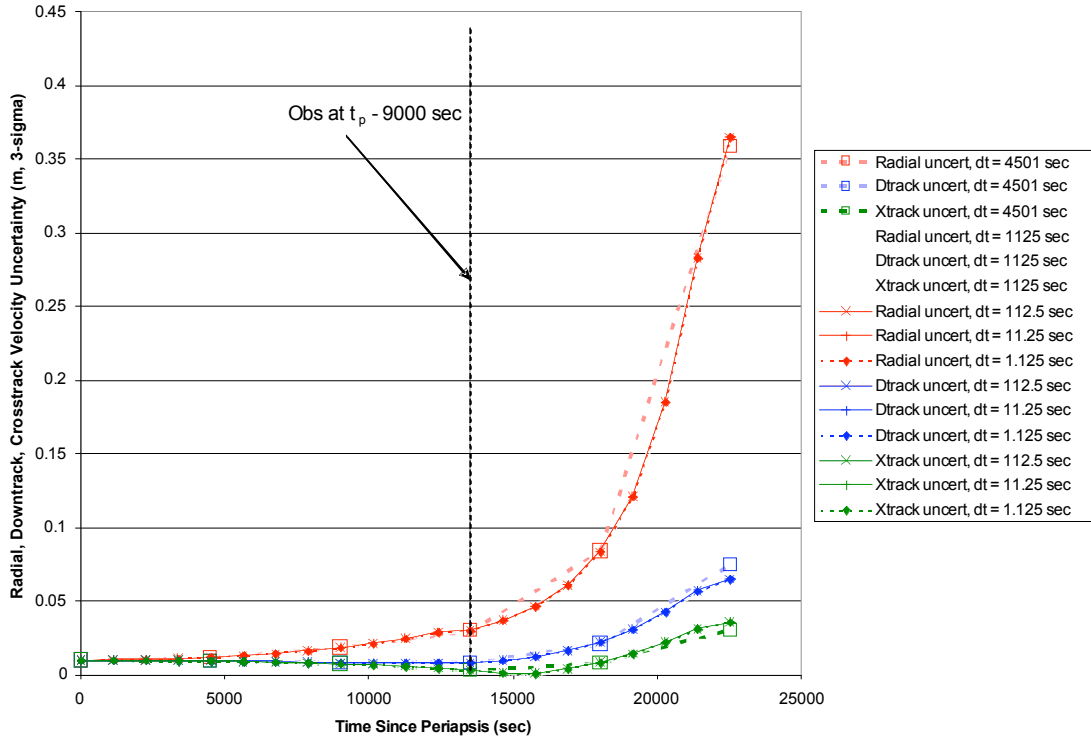


Figure 9: Stepsize-sensitivity of velocity uncertainty (3-sigma) from LTCF (integrated state transition) with 0.7% gravity error considered, incorporating one limb angle observation at periapsis minus 9000 sec.

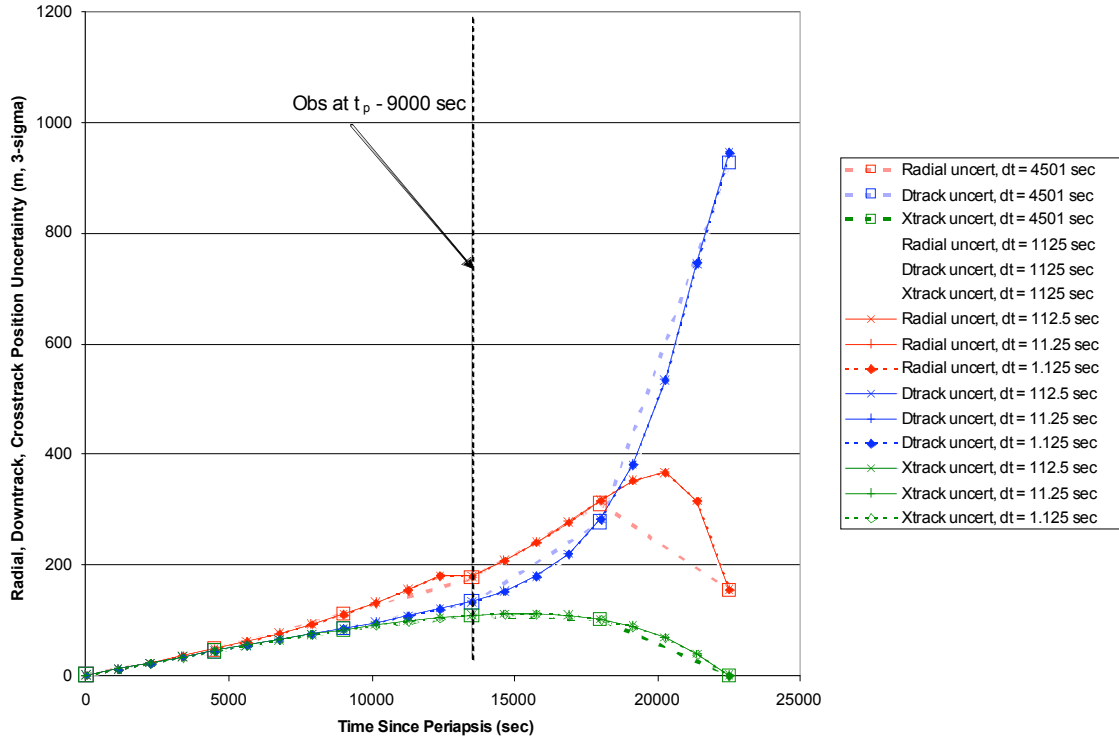


Figure 10: Stepsize-sensitivity of position uncertainty (3-sigma) from SPCF with 0.7% gravity error considered, incorporating one limb angle observation at $t = \text{periapsis} - 9000$ seconds.

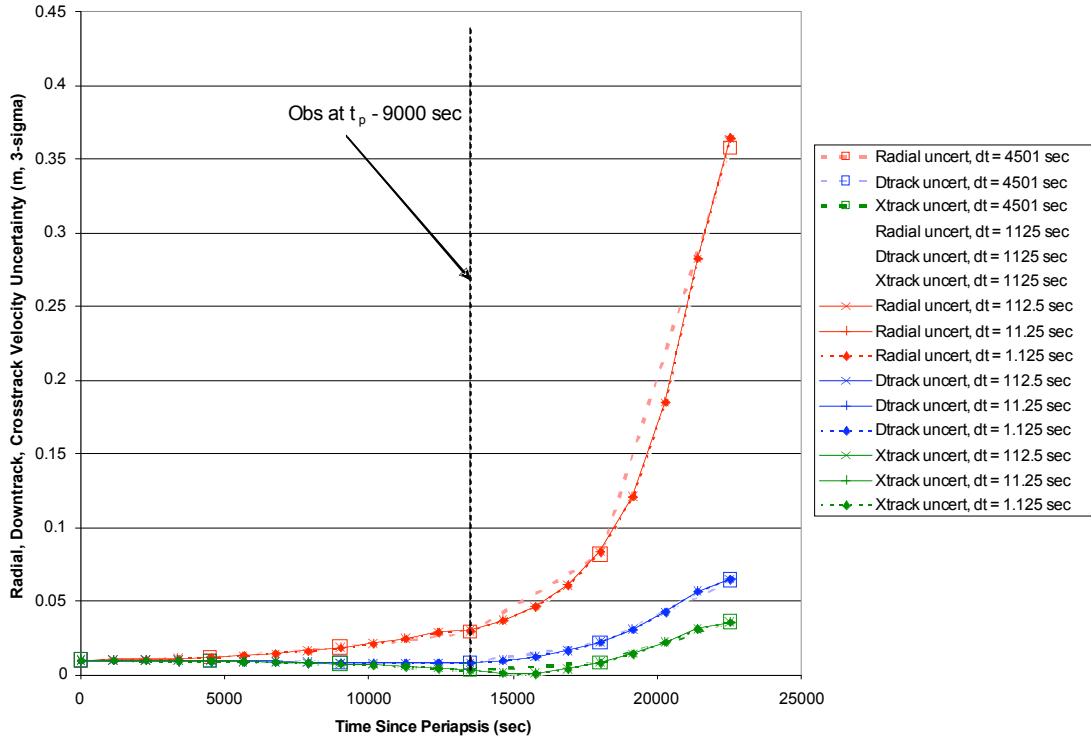


Figure 11: Stepsize-sensitivity of velocity uncertainty (3-sigma) from SPCF with 0.7% gravity error considered, incorporating one limb angle observation at $t = \text{periapsis} - 9000$ seconds.

Conclusions

The present work demonstrates the implementation, and examines the accuracy and throughput performance, of the Sigma-Point Consider Filter (SPCF) algorithm first developed and discussed in [4]. The SPCF was compared with two versions of a standard Linear-Theory Consider Filter (LTCF), with parameterization and models, in an example nonlinear autonomous orbit determination problem entailing consider covariance analysis. Given an existing linear-theory and sigma-point filter-development interface tool such as the “*stochastic.py*” Python-language software module developed at JPL for this research, the SPCF requires less effort to implement than the equivalent LTCF – this should be generally true for any consider-estimation problem because the SPCF does not require the derivation or implementation of any measurement or dynamical partial derivatives. For this example nonlinear problem the SPCF was seen, as expected, to yield high-accuracy covariance results using a propagation timestep size that was larger than a functionally-equivalent LTCF could achieve, even when the LTCF propagated its state transition matrix with no approximation, i.e. via the integral in Equation (20).

Execution runtime was also evaluated for the SPCF, and also for approximate-state-transition and integrated-state-transition versions of the LTCF. The SPCF was found to be faster than either LTCF version. However, because the single-timestep execution runtime of the SPCF is actually measured to be slower than either LTCF version (four times slower than the approximate-state-transition LTCF and two times slower than the integrated-state-transition LTCF), it is noted that the run speed advantages seen here for the SPCF are really due to the large propagation step-size enabled by applying the SPCF nonlinear covariance propagation technique to a nonlinear problem. The runtime advantage over the LTCF is likely to diminish in the presence of smaller nonlinearities. Processing of a single measurement, using different propagation timestep sizes, yielded superior accuracy results for the SPCF at larger timestep sizes, which is consistent with the other results presented here. Based on these accuracy and execution-time results, the Sigma-Point Consider Filter merits consideration for use in nonlinear consider-estimation problems such as the example presented here, particularly those problems in which:

- there is a requirement for statistically-meaningful, realistic and accurate consider covariance values;
- side-stepping the development and testing of partial derivative expressions and code presents an advantage in development cycle schedule and cost; and
- there is a real-time-interval (RTI) usage advantage, or at least no penalty, associated with using a large propagation timestep, e.g. to accurately predict a future value of the covariance in fewer RTI's, or to use a smaller average fraction of the RTI to achieve the filter's calculations.

References:

- [1] Julier, S. and J. Uhlmann. *Unscented Filtering and Nonlinear Estimation*, Proceedings of the IEEE, vol. 92, no. 3, March 2004.
- [2] Karlgaard, C., and Schaub, H. *Comparison of Several Nonlinear Filters for a Benchmark Tracking Problem*. Paper AIAA 2006-6243, AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, CO, Aug. 2006.
- [3] Crassidis, J.L., and Markley, F.L. *Unscented Filtering for Spacecraft Attitude Estimation*, AIAA Journal of Guidance, Control, and Dynamics, Vol. 26, No. 4, July-Aug. 2003, pp. 536-542.
- [4] Lisano, M. *Nonlinear Consider Covariance Analysis Using a Sigma-Point Filter Formulation*. Paper AAS 06-035, 29th Annual AAS Guidance and Control Conference, Breckenridge, CO, Feb. 4-8, 2006.
- [5] Tapley, B., B. Schutz, G. Born. Statistical Orbit Determination. Elsevier Academic Press, 2004. pp. 387-437.
- [6] Gelb, A. Applied Optimal Estimation. M.I.T. Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1974
- [7] Crassidis, J. L., and J. Junkins. Optimal Estimation of Dynamic Systems. Chapman & Hall/CRC Applied Mathematics and Nonlinear Science Series, 2004.