Multiple Event Triggers in Linear Covariance Analysis for Spacecraft Rendezvous

Adam Sievers,[*] Renato Zanetti,[†] David C. Woffinden,[‡]

The Charles Stark Draper Laboratory, Houston, Texas, 77058

Linear covariance analysis of both navigation error and trajectory dispersion is a powerful tool for spacecraft rendezvous analysis and design. The introduction of multiple events triggered on state conditions causes discrepancies between the linear covariance results and the theoretical results. This work investigates the causes of the discrepancy and introduces a solution to it. The proposed technique is validated by comparison to the results from Monte Carlo simulations.

I. Introduction

Linear covariance analysis techniques (LinCov)[1,2] generates analytical statistics from a single run similar to sample statistics produced from hundreds or even thousand Monte Carlo runs. Because of its ability to produce results faster than Monte Carlo, LinCov is particularly useful during the trade study phase of the design process. LinCov does not represent errors directly, but requires their statistics as an input. For example, the difference between the true dynamics and the modeled dynamics is rendered through a covariance matrix. Therefore while Monte Carlo techniques aid in sizing the modeling errors, LinCov assumes the errors are given. This assumption, together with the lack of granularity and non-linearity, makes LinCov less accurate than high-fidelity Monte Carlo simulations: another reason why these techniques are more relied on during the early design phase.

In previous work, event triggers for linear covariance analysis were introduced.[3] For state driven events such as initializing a maneuver based on a spacecraft range to a target, the theoretical developments in Ref. 3 provide a way to capture the resulting effects on trajectory dispersions and navigation errors. In Ref. 3 the authors apply a single event trigger to command the final maneuver for a spacecraft docking scenario. When trying to extend the technique to multiple event triggers however, the results from LinCov did not always accurately represent the actual errors anymore.

At orbital rates, few seconds translates into many kilometers traveled. When an event is triggered by means other than time, there is a time delay between the nominal event time and the time at which the event actually happens. This time difference makes the dispersion between the true inertial state and the nominal inertial state grow very large. During the terminal phases of rendezvous and docking, the relative navigation and trajectory errors are of the order of centimeters while the inertial dispersions can be thousand of kilometers because of the event trigger. This large difference causes numerical issues when trying to compute accurate relative dispersions from the inertial dispersions. These numerical issues are made more severe because LinCov carries covariances, which are of the order of the square of the errors.

Possible solutions to this issue include square root filters,[4] carrying relative states instead of inertial states, and higher precision in the numerical computations. While theoretically attractive, square root filters require more computations. A purely relative simulation using Clohessy-Wiltshire equations would introduce unnecessary approximations while carrying a single inertial state and a relative state would add complexity to the simulation. The use of square root filters or relative states would greatly depart from the theory and

[*] Second Lieutenant, US Air Force. Rice University Graduate Student and Draper Laboratory Fellow, 17629 El Camino Real, Suite 470. asievers@draper.com

[†] Senior Member of the Technical Staff, GN&C Manned Space Systems, 17629 El Camino Real, Suite 470. rzanetti@draper.com, Member AIAA

[‡] Senior Member of the Technical Staff, Mission Analysis and Design, 17629 El Camino Real, Suite 470. dwoffinden@draper.com

application of LinCov previously employed, making the use of heritage solutions difficult. A 64 bit floating number will incur in numerical problems during the LinCov simulation of a rendezvous employing two event triggers. While switching to 128 bit floats might seem the most obvious solution, the approach is not always possible using standard commercial software.

The main purpose of this work is to propose a simple solution to the multiple event trigger discrepancy problem observed for orbital rendezvous using a covariance *resetting* technique. LinCov runs employing the solution are compared to the results of a Monte Carlo simulation to demonstrate that the two approaches coincide. Besides establishing and validating the theory to eliminate numerical discrepancies that can emerge when performing multiple event triggers for orbital rendezvous, there are several additional objectives of this paper not addressed in previous research efforts[5] comparing LinCov and Monte Carlo results. These objectives include the 1) implementation of an onboard reduced-state navigation filter to capture a more realistic performance of a properly tuned filter, 2) demonstration of a full rendezvous scenario with a variegate sensor suite and an assortment of targeting algorithms necessary for proximity operations, and 3) quantification of the arrival time uncertainty predictions associated with event triggers.

II. Simulation Models

The rendezvous model used for LinCov carries covariance matrices derived from three distinct collections of state variables: a reference or nominal state $\bar{\mathbf{x}}$, the true or environment state $\mathbf{x}$, and the estimated or onboard state $\hat{\mathbf{x}}$. The environment dispersion is the different between true and nominal state $\delta\mathbf{x}$, and needs to be small in order to achieve mission success. The different between the estimated state and the truth is the navigation error $\delta\mathbf{e}$, which is desired to be small for good navigation performance. Finally, the difference between the true state and the navigation state forms the navigation dispersions $\delta\hat{\mathbf{x}}$.

The components of the nominal state include various chaser, target, and parameter states consisting of

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{\mathbf{r}}_c^{\mathrm{T}} & \bar{\mathbf{v}}_c^{\mathrm{T}} & \bar{\bar{\mathbf{q}}}_c^{\mathrm{T}} & \bar{\boldsymbol{\omega}}_c^{\mathrm{T}} & \bar{\mathbf{r}}_t^{\mathrm{T}} & \bar{\mathbf{v}}_t^{\mathrm{T}} & \bar{\bar{\mathbf{q}}}_t^{\mathrm{T}} & \bar{\boldsymbol{\omega}}_t^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

the inertial position, inertial velocity, inertial to body attitude quaternion, and attitude rate in the body frame for both the chaser and target vehicles. In Equation (1), subscript c represents the chaser, and subscript t the target.

The sensors used are the gyroscope, accelerometer, star tracker, light detection and ranging system (lidar), and global positioning system (GPS) receiver for chaser spacecraft and GPS receiver for the target spacecraft. A detailed description of all the models can be find in Ref. 6.

A. Environment Dynamics Models

The simulation uses a gravity and aerodynamic model to describe the motion of both the chaser and target vehicles around the Earth. The parameters used to calculate the gravitational acceleration can be found in Table 1.[7]

Coefficient	Value
J_2	1.082627×10^{-3}
J_3	-2.532657×10^{-6}
J_4	-1.619622×10^{-6}
μ	3.986005×10^{5} km^3/s^2
$R_\oplus$	6378.136300 km

Table 1. Earth Parameters Utilized in Gravity Model

The model for atmospheric drag is

$$\mathbf{a}_{aero} = -\frac{1}{2}\frac{\rho_{atmos}}{BC} v_{rel}\mathbf{v}_{rel}. \tag{2}$$

This acceleration is a function of atmospheric density (ρ_{atmos}), the vehicle's ballistic coefficient (BC), and the vehicle's velocity relative to the motion of Earth's atmosphere ($\mathbf{v}_{rel}$).[7] The density of the atmosphere at the vehicle's altitude is a difficult quantity to model. The simulation uses a modified exponential method, Babb-Mueller, to determine ρ_{atmos}. The ballistic coefficient ($BC = m/c_D S$), is a function of the vehicle's mass (m), cross-sectional area normal to the velocity vector (S), and coefficient of drag (c_D). Equation (4) is substituted into Equation (3) to calculate the relative velocity. The atmosphere is assumed to be stationary with respect to Earth. In these equations, $\boldsymbol{\omega}_\oplus$ is Earth's rotational velocity vector and r_x and r_y are components of the vehicle's position vector. Further,

$$\mathbf{v}_{rel} = \mathbf{v} - \mathbf{v}_{atmos} \tag{3}$$

$$\mathbf{v}_{atmos} = \boldsymbol{\omega}_\oplus \times \mathbf{r} = \begin{bmatrix} -\omega_\oplus r_y \\ \omega_\oplus r_x \\ 0 \end{bmatrix} \tag{4}$$

B. Rotational Dynamics Models

The simulation utilizes quaternions to represent and propagate attitude.[8] Quaternions are composed of a vector part ($\mathbf{q}_v$), and scalar part (q_s). The orientation of a spacecraft can by represented by three Euler angles, roll about the x-axis, pitch about the y-axis, and yaw about the z-axis. Any sequence of Euler's angles can be described by a single rotation (φ), about a unit vector ($\mathbf{k}$), known as the Euler angle and axis.[9] The inertial-to-body quaternion is written as

$$\bar{\mathbf{q}}_i^b = \begin{bmatrix} q_s \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\varphi}{2}\right) \\ -\sin\left(\frac{\varphi}{2}\right)\mathbf{k} \end{bmatrix}, \tag{5}$$

where the minus sign is present because the Shuttle quaternion convention is used. Quaternions have several advantages in representing attitude. Quaternions require fewer parameters and calculations than direction cosine matrices and avoid singularities that exist when using Euler angles.[10] The attitude is propagated using Equation (6)

$$\dot{\bar{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix} 0 \\ -\boldsymbol{\omega} \end{bmatrix} \otimes \bar{\mathbf{q}}. \tag{6}$$

In this equation, $\otimes$ denotes quaternion multiplication. The equation for propagating attitude rate is derived from Euler's rotational equation of motion or moment equation[11,12]

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}, \tag{7}$$

where $\boldsymbol{\tau}$ is a moment or torque, $\mathbf{J}$ is the inertia matrix, and $\boldsymbol{\omega}$ is the vehicle attitude rate, all quantities are expressed in the vehicle's body-fixed frame. Equation (7) can be rewritten as

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\left(\boldsymbol{\tau} - [\boldsymbol{\omega}\times]\,\mathbf{J}\boldsymbol{\omega}\right), \tag{8}$$

where $\boldsymbol{\tau}$ represents torques acting on the vehicle, which can include external torques caused by perturbations and control torques, and $[\boldsymbol{\omega}\times]$ a skew-symmetric matrix

$$[\boldsymbol{\omega}\times] \equiv \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

External gravitational torques are caused by the nonuniform distribution of vehicle's mass about it's center of mass. The force of gravity acting on the vehicle is different at different locations causing gravity induced torques. The equation for the gravitational torques, $\boldsymbol{\tau}_{grav}$, is developed in Ref 13. Specifically,

$$\boldsymbol{\tau}_{grav} = \int_B \mathbf{r}_{dm} \times \left[-\frac{\mu\left(\mathbf{r} + \mathbf{r}_{dm}\right)dm}{\|\mathbf{r} + \mathbf{r}_{dm}\|^3}\right], \tag{9}$$

where $\mathbf{r}_{dm}$ is the vector from the center of mass to the differential element of mass and $\mathbf{r}$ is the vehicle's position vector in the inertial frame. Equation (9) can be simplified by rotating to the body frame

$$\boldsymbol{\tau}_{grav} = \frac{3\mu}{r^5}\left[\mathbf{r}^b \times \mathbf{J}\mathbf{r}^b\right], \tag{10}$$

where superscript b indicates the body-fixed frame. The external torques caused by the vehicle's center of pressure being located away from its center of mass are easily computed, if the location of the center of pressure with respect to the center of mass is known in body coordinates. The acceleration caused by atmospheric drag is shown in Equation (2) and the aero force is easily obtained by multiplying $\mathbf{a}_{aero}$ by vehicle mass. The torques are

$$\boldsymbol{\tau}_{aero} = \mathbf{r}_{cp}^{b} \times \mathbf{T}_{i}^{b}\mathbf{f}_{aero}. \tag{11}$$

C. Sensor Models

Sensors onboard the chaser spacecraft include an inertial measurement unit (IMU), star tracker, lidar, and GPS receiver. The IMU consists of a gyroscope sensor and accelerometer.[14] The target spacecraft has various sensors onboard but only the GPS receiver is modeled by the simulation. The IMU is used to propagate the estimated chaser states and the discrete sensors are used to update filter states and covariance.

1. Gyroscope

The gyroscope sensor onboard the chaser spacecraft measures the three axis angular velocity of the vehicle. The sensor's output is the integrated angular velocity over the measurement time step.[15]

$$\tilde{\boldsymbol{\omega}}_{IMU} = \mathbf{T}\left(\boldsymbol{\epsilon}_{gyro}\right)\left[\mathbf{I} + \mathbf{D}\left(\mathbf{b}_{sf} + \mathbf{m}_{sf}\right)\right]\boldsymbol{\omega} + \mathbf{b}_{gyro} + \mathbf{m}_{gyro} + \boldsymbol{\eta}_{gyro} \tag{12}$$

$$\Delta\tilde{\boldsymbol{\theta}} = \int_{t}^{t+\Delta t} \tilde{\boldsymbol{\omega}} dt \tag{13}$$

In Equation (12) the measured angular velocity is dependent on the true angular velocity ($\boldsymbol{\omega}$), misalignments ($\mathbf{T}\left(\boldsymbol{\epsilon}_{gyro}\right)$), the scale factor bias and Markov process ($\mathbf{b}_{sf}$ and $\mathbf{m}_{sf}$), the gyro bias and Markov process ($\mathbf{b}_{gyro}$ and $\mathbf{m}_{gyro}$), and the gyroscope measurement noise ($\boldsymbol{\eta}_{gyro}$). The symbol $\mathbf{D}$ denotes the diagonal matrix of the quantities between the parentheses. The gyroscope operates in model replacement mode, the filter does not estimate angular velocity but takes it directly from the gyroscope measurement. This technique is an acceptable practice because of gyroscopes extensive flight heritage and the sensor's highly accurate measurements. The errors present in the gyroscope model are given in Table 2.

Error Type	Variable	1σ Standard Deviation	Units
Misalignment	$\boldsymbol{\epsilon}_{gyro}$	2.0000	arcsec
Bias in roll, pitch, yaw	$\mathbf{b}_{gyro}$	3.6667×10^{-3}	deg/hr
Markov Bias in roll, pitch, yaw	$\mathbf{m}_{gyro}$	3.6667×10^{-3}	deg/hr
Scale Factor in roll, pitch, yaw	$\mathbf{b}_{sf}$	5.0000	ppm
Markov Scale Factor	$\mathbf{m}_{sf}$	5.0000	ppm
Gyroscope Measurement Noise	$\boldsymbol{\eta}_{gyro}$	2.5000×10^{-3}	deg/$\sqrt{\text{hr}}$

Table 2. Gyroscope Model Error Parameters

2. Accelerometer

The accelerometer measures three axis non-gravitational accelerations. The sensor's output is the integrated non-conservative acceleration over the measurement time step.[15] Specifically,

$$\tilde{\mathbf{a}}_{IMU}^{i} = \mathbf{T}\left(\boldsymbol{\epsilon}_{accel}\right)\left[\mathbf{I} + \mathbf{D}\left(\mathbf{b}_{sf} + \mathbf{m}_{sf}\right)\right]\mathbf{a} + \mathbf{b}_{accel} + \mathbf{m}_{accel} + \boldsymbol{\eta}_{accel} \tag{14}$$

$$\Delta\tilde{\mathbf{v}}^{i} = \int_{t}^{t+\Delta t} \tilde{\mathbf{a}}_{IMU}^{i} dt \tag{15}$$

In Equation (14) the measured acceleration is dependent on the true acceleration, misalignments ($\mathbf{T}\left(\boldsymbol{\epsilon}_{accel}\right)$), the scale factor bias and Markov process ($\mathbf{b}_{sf}$ and $\mathbf{m}_{sf}$), the accelerometer bias and Markov process ($\mathbf{b}_{accel}$ and $\mathbf{m}_{accel}$), and the accelerometer measurement noise ($\boldsymbol{\eta}_{accel}$). The errors present in the accelerometer model are given in Table 3.

Error Type	Variable	1σ Standard Deviation	Units
Misalignment	$\boldsymbol{\epsilon}_{accel}$	5.0000	arcsec
Bias in x, y, z	$\mathbf{b}_{accel}$	50.0000	μg
Markov Bias in x, y, z	$\mathbf{m}_{accel}$	50.0000	μg
Scale Factor in x, y, z	$\mathbf{b}_{sf}$	175.0000	ppm
Markov Scale Factor in x, y, z	$\mathbf{m}_{sf}$	175.0000	ppm
Accelerometer Measurement Noise	$\boldsymbol{\eta}_{accel}$	1.5000×10^{-5}	m/s/$\sqrt{\text{s}}$

Table 3. Accelerometer Model Error Parameters

3. Star Tracker

The star tracker provides a measurement to update the chaser orientation. The sensor captures an image of local stars and compares it to a catalog of inertial star locations.[16] The generated measurement is an inertial frame to star tracker frame quaternion. The sensor model is given as

$$\tilde{\bar{\mathbf{q}}}_i^{st} = \bar{\mathbf{q}}\left(\boldsymbol{\eta}_{st}\right) \otimes \bar{\mathbf{q}}\left(\mathbf{b}_{st}\right) \otimes \bar{\mathbf{q}}_b^{st} \otimes \bar{\mathbf{q}}_i^b \ , \tag{16}$$

where $\bar{\mathbf{q}}_{st}^b$ is the body to star tracker quaternion with misalignments, $\bar{\mathbf{q}}_b^i$ is the inertial to body quaternion, and $\bar{\mathbf{q}}\left(\mathbf{b}_{st}\right)$ and $\bar{\mathbf{q}}\left(\boldsymbol{\eta}_{st}\right)$ are quaternions derived from sensor bias and noise.[17] The errors present in the star tracker model are given in Table 4.

Error Type	Variable	1σ Standard Deviation	Units
Misalignment	$\boldsymbol{\epsilon}_{st}$	4.0000	arcsec
Bias in roll, pitch, yaw	$\mathbf{b}_{st}$	3.3333	arcsec
Star Tracker Measurement Noise	$\boldsymbol{\eta}_{st}$		
boresight axis		69.0000	arcsec
cross boresight axis		16.3333	arcsec

Table 4. Star Tracker Model Error Parameters

4. Chaser GPS

The chaser is equipped with a GPS receiver to update position and velocity. The satellites in the GPS constellation broadcast a ranging signal used by the receiver to calculate position and velocity.[18] The sensor model from reference[19] is

$$\tilde{\mathbf{r}}_c = \mathbf{r}_c + \mathbf{m}_{gps_r} + \boldsymbol{\eta}_{gps_r} \ , \tag{17}$$

and

$$\tilde{\mathbf{v}}_c = \mathbf{v}_c + \mathbf{m}_{gps_v} + \boldsymbol{\eta}_{gps_v} . \tag{18}$$

The sensor model includes the chaser position ($\mathbf{r}_c$), the first-order Markov process vector ($\mathbf{m}_{gps_r}$), and GPS measurement noise ($\boldsymbol{\eta}_{gps_r}$) associated with chaser position. The model also includes the chaser velocity ($\mathbf{v}_c$), the first-order Markov process vector ($\mathbf{m}_{gps_v}$), and GPS measurement noise ($\boldsymbol{\eta}_{gps_r}$) associated with chaser velocity. The errors present in the chaser GPS model are given in Table 5.

5. Target GPS

The target is equipped with a GPS receiver to update position and velocity. The sensor model, from reference,[19] is

$$\tilde{\mathbf{r}}_t = \mathbf{r}_t + \mathbf{m}_{tgps_r} + \boldsymbol{\eta}_{tgps_r} \ , \tag{19}$$

and

$$\tilde{\mathbf{v}}_t = \mathbf{v}_t + \mathbf{m}_{tgps_v} + \boldsymbol{\eta}_{tgps_v} . \tag{20}$$

Error Type	Variable	1σ Standard Deviation	Units
Position Bias in x, y, z	$\mathbf{m}_{gps_r}$	5.7735	m
Velocity Bias in x, y, z	$\mathbf{m}_{gps_v}$	5.7735×10^{-2}	m/s
GPS Position Measurement Noise	$\boldsymbol{\eta}_{gps_r}$	5.7735	m
GPS Velocity Measurement Noise	$\boldsymbol{\eta}_{gps_v}$	5.7735×10^{-2}	m/s
GPS Time Constant	τ_{gps}	300	s

Table 5. Chaser GPS Model Error Parameters

The sensor model in Equation (19) includes the target position ($\mathbf{r}_t$), the first-order Markov process vector ($\mathbf{m}_{tgps_r}$), and GPS measurement noise ($\boldsymbol{\eta}_{tgps_r}$) associated with target position. The model also includes the target velocity ($\mathbf{v}_t$), the first-order Markov process vector ($\mathbf{m}_{tgps_v}$), and GPS measurement noise ($\boldsymbol{\eta}_{tgps_r}$) associated with target velocity. The errors present in the target GPS model are given in Table 6.

Error Type	Variable	1σ Standard Deviation	Units
Position Bias in x, y, z	$\mathbf{m}_{tgps_r}$	30.0000	m
Velocity Bias in x, y, z	$\mathbf{m}_{tgps_v}$	6.0000×10^{-2}	m/s
GPS Position Measurement Noise	$\boldsymbol{\eta}_{tgps_r}$	30.0000	m
GPS Velocity Measurement Noise	$\boldsymbol{\eta}_{tgps_v}$	6.0000×10^{-2}	m/s
GPS Time Constant	τ_{tgps}	300	s

Table 6. Target GPS Model Error Parameters

6. *Lidar*

The lidar instrument uses laser light to track the target. The sensor provides a measurement to update chaser position and attitude and target position.[20] The relative position vector in the lidar frame is given by the equation

$$\mathbf{r}^l_{rel} = \mathbf{T}^l_b \mathbf{T}^b_i (\mathbf{r}_t - \mathbf{r}_c) + \mathbf{b}_{lidar} + \boldsymbol{\eta}_{lidar} \ , \tag{21}$$

where $\mathbf{r}_t$ and $\mathbf{r}_c$ are the target and chaser position vectors, $\mathbf{b}_{lidar}$ is the lidar bias, and $\boldsymbol{\eta}_{lidar}$ is the lidar measurement noise. The relative position vector is used to generate the lidar measurements of range (ρ), azimuth (az), and elevation (el). Specifically,

$$\mathbf{r}^l_{rel} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{22}$$

$$\begin{aligned} \rho &= \left\|\mathbf{r}^l_{rel}\right\|^2 \\ az &= \arctan\left(y/x\right) \\ el &= \arcsin\left(z/\rho\right) \end{aligned} \tag{23}$$

The errors present in the lidar model are given in Table 7.

Error Type	Variable	1σ Standard Deviation	Units
Misalignment	$\boldsymbol{\epsilon}_{lidar}$	0.1000	deg
Range Bias	b_ρ	0.1667	m
Azimuth Bias	b_{az}	0.0333	deg
Elevation Bias	b_{el}	0.0333	deg
Lidar Range Measurement Noise	η_ρ	0.0333	m
Lidar Azimuth Measurement Noise	η_{az}	0.0333	deg
Lidar Elevation Measurement Noise	η_{el}	0.0333	deg

Table 7. Lidar Model Error Parameters

D. Actuator Models

Actuators are used to manipulate the spacecraft's translational and rotational velocities. The chaser is equipped with thrusters for both attitude control and translational maneuvers.[21] The error parameters of the actuators discussed are arbitrarily selected and are comparable to values presented in references.[5,19]

1. *Translational Maneuver Thrusters*

The model for impulsive maneuvers is shown in Equation (24).[5] Maneuvers are modeled as impulsive changes in velocity,

$$\Delta \mathbf{v}^i = \mathbf{T}_b^i \left\{ \mathbf{T}\left(\boldsymbol{\epsilon}_{mvr}\right) \left[\mathbf{I} + \mathbf{D}\left(\mathbf{b}_{sf}\right)\right] \hat{\mathbf{T}}_i^b \Delta \hat{\mathbf{v}}_{cmd}^i + \mathbf{b}_{mvr} + \boldsymbol{\eta}_{mvr} \right\} , \tag{24}$$

$$\mathbf{v}^{+\Delta V} = \mathbf{v}^{-\Delta V} + \Delta \mathbf{v}. \tag{25}$$

Table 8 summarizes the translational maneuver uncertainties used in the simulation.

Error Type	Variable	1σ Standard Deviation	Units
Misalignment	$\boldsymbol{\epsilon}_{mvr}$	3.3333×10^{-4}	deg
Bias in x, y, z	$\mathbf{b}_{mvr}$	1.3333×10^{-4}	m/s
Scale Factor Bias in x, y, z	$\mathbf{b}_{sf}$	1.6500×10^{-3}	m/s
Translational Maneuver Noise	η_{mvr}	5.0000×10^{-4}	m/s

Table 8. Translational Maneuver Errors

2. *Reaction Control System*

The reaction control system (RCS) thrusters produce the torques required to alter the chaser's attitude rate. The RCS actuator model is shown in Equation (26).[5]

$$\boldsymbol{\tau}_{ctrl}^b = \mathbf{T}\left(\boldsymbol{\epsilon}_{ctrl}\right) \left[\mathbf{I} + \mathbf{D}\left(\mathbf{b}_{sf}\right) \hat{\boldsymbol{\tau}}_{cmd}^b + \mathbf{b}_{rot} + \boldsymbol{\eta}_{rot} \right] \tag{26}$$

The commanded attitude for both vehicles is set to the target local-vertical local-horizontal attitude, while the commanded attitude rate is set as the orbital rate. Table 9 summarizes the rotational maneuver uncertainties used in the simulation. For more information about spacecraft control see reference.[22]

III. Event Triggers

In LinCov, events generally occur at specific times. However, in many orbital rendezvous missions the terminal phase initiation (TPI) maneuver is triggered when the relative position of the chaser and target vehicles reaches a predetermined value. For this simulation, the TPI maneuver is triggered when the chaser reaches a specific downrange distance from the target. The techniques for applying triggers in LinCov are discussed in reference[3] and are now briefly reviewed.

Error Type	Variable	1σ Standard Deviation	Units
Misalignment	$\boldsymbol{\epsilon}_{ctrl}$	3.3333×10^{-3}	deg
Bias in roll, pitch, yaw	$\mathbf{b}_{rot}$	3.3333×10^{-5}	N m
Scale Factor Bias in roll, pitch, yaw	$\mathbf{b}_{sf}$	3.3333×10^{-3}	N m
Rotational Maneuver Noise	$\boldsymbol{\eta}_{rot}$	1.0000×10^{-10}	N m

Table 9. Rotational Maneuver Errors

An event trigger is a condition represented as a function of the navigation state,

$$\boldsymbol{\Psi}\left[\hat{\mathbf{x}}(t_e)\right] = 0. \tag{27}$$

When the specified condition is achieved, it triggers a specific event that occurs at the true event time (t_e). The trigger condition generally does not occur at the nominal event time ($\bar{t}_e$),

$$\boldsymbol{\Psi}\left[\hat{\mathbf{x}}(\bar{t}_e)\right] \neq 0 \ , \tag{28}$$

however, if the difference between t_e and $\bar{t}_e$ is small, the true and estimated states at t_e can be written as a function of this difference

$$\begin{aligned} \mathbf{x}(t_e) &\approx \mathbf{x}(\bar{t}_e) + \dot{\mathbf{x}}(\bar{t}_e)\left[t_e - \bar{t}_e\right], & (29) \\ \hat{\mathbf{x}}(t_e) &\approx \hat{\mathbf{x}}(\bar{t}_e) + \dot{\hat{\mathbf{x}}}(\bar{t}_e)\left[t_e - \bar{t}_e\right], & (30) \end{aligned}$$

where

$$\begin{aligned} \dot{\mathbf{x}}(\bar{t}_e) &= \dot{\bar{\mathbf{x}}}(\bar{t}_e) + \delta\dot{\mathbf{x}}(\bar{t}_e) \ , & (31) \\ \dot{\hat{\mathbf{x}}}(\bar{t}_e) &= \dot{\bar{\mathbf{x}}}(\bar{t}_e) + \delta\dot{\hat{\mathbf{x}}}(\bar{t}_e). & (32) \end{aligned}$$

Expanding Equation (27) results in

$$0 = \boldsymbol{\Psi}\left[\bar{\mathbf{x}}(\bar{t}_e)\right] + \boldsymbol{\Psi}_x\left[\delta\hat{\mathbf{x}}(\bar{t}_e) + \dot{\bar{\mathbf{x}}}(\bar{t}_e)\delta t_e\right], \tag{33}$$

where

$$\boldsymbol{\Psi}\left[\bar{\mathbf{x}}(\bar{t}_e)\right] = 0; \quad \boldsymbol{\Psi}_x = \left.\frac{\partial\boldsymbol{\Psi}}{\partial\hat{\mathbf{x}}}\right|_{\hat{\mathbf{x}}=\bar{\mathbf{x}}}; \quad \delta t_e = t_e - \bar{t}_e \ . \tag{34}$$

This leads to an expression for δt_e

$$\delta t_e = -\left[\boldsymbol{\Psi}_x\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\boldsymbol{\Psi}_x\delta\hat{\mathbf{x}}(\bar{t}_e). \tag{35}$$

The row vector $\mathbf{s}^T$ is defined as

$$\mathbf{s}^T = -\left[\boldsymbol{\Psi}_x\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\boldsymbol{\Psi}_x. \tag{36}$$

The environment and navigation dispersions at time t_e are given by the equation

$$\delta\mathbf{x}(t_e) = \mathbf{x}(t_e) - \bar{\mathbf{x}}(\bar{t}_e) = \delta\mathbf{x}(\bar{t}_e) + \mathbf{N}\dot{\bar{\mathbf{x}}}(\bar{t}_e)\mathbf{s}^T\delta\hat{\mathbf{x}}(\bar{t}_e) \ , \tag{37}$$

and

$$\delta\hat{\mathbf{x}}(t_e) = \hat{\mathbf{x}}(t_e) - \bar{\mathbf{x}}(\bar{t}_e) = \delta\hat{\mathbf{x}}(\bar{t}_e) + \dot{\bar{\mathbf{x}}}(\bar{t}_e)\mathbf{s}^T\delta\hat{\mathbf{x}}(\bar{t}_e) \ . \tag{38}$$

Let n be the number of true states, m be the number of estimated states, and $\mathbf{N}$ an $n \times m$ matrix that maps the estimated states to the true states, such that the navigation errors can be written as

$$\mathbf{e} = \mathbf{N}^T\delta\mathbf{x} - \delta\hat{\mathbf{x}}, \tag{39}$$

where $\mathbf{N}^T\mathbf{N} = \mathbf{I}_{m\times m}$. Using Equation (37) and Equation (38), the augmented state at the true event time is

$$\delta\mathbf{x}_{aug} = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{N}\dot{\bar{\mathbf{x}}}(\bar{t}_e)\mathbf{s}^T \\ \mathbf{0}_{n\times m} & \mathbf{I}_{m\times m} + \dot{\bar{\mathbf{x}}}(\bar{t}_e)\mathbf{s}^T \end{bmatrix}\delta\mathbf{x}_{aug}(\bar{t}_e). \tag{40}$$

The mean values of the environment and navigation dispersions are zero and the expected value of the augmented state is zero. The augmented covariance matrix is reshaped after an event trigger

$$\mathbf{P}_{aug}^{+s} = \mathbf{I}_s \mathbf{P}_{aug}^{-s} \mathbf{I}_s^T \ , \tag{41}$$

where the shaping matrix ($\mathbf{I}_s$) is

$$\mathbf{I}_s = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{N}\dot{\bar{\mathbf{x}}}(\bar{t}_e)\mathbf{s}^T \\ \mathbf{0}_{n\times m} & \mathbf{I}_{m\times m} + \dot{\bar{\mathbf{x}}}(\bar{t}_e)\mathbf{s}^T \end{bmatrix} . \tag{42}$$

Under certain conditions for orbital rendezvous applications (i.e. large inertial dispersions and very accurate relative navigation errors) numerical instabilities can arise when performing event triggers. This problem is only magnified if multiple events triggers are used. A solution that resolves this issue quickly and reliably utilizes a covariance *resetting* technique introduced in the subsequent section.

A. Resetting Method

Since the relative dispersions of position and velocity are of primary interest for proximity operations, it becomes desirable to reset the augmented state matrix to reduce the inertial dispersions while conserving the relative trajectory dispersions and ensuring the navigation errors remain unchanged. The nominal rendezvous trajectory is given in terms of relative position and velocity. The relative nominal trajectory is not affected by the resetting method and therefore the relative trajectory dispersions are not affected. This method essentially alters the inertial nominal trajectory so that the inertial dispersions are reset without affecting the navigation error. Recall that the environment and navigation dispersions ($\delta\mathbf{x}$ and $\delta\hat{\mathbf{x}}$), are defined as

$$\delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}} \ , \tag{43}$$

and

$$\delta\hat{\mathbf{x}} = \hat{\mathbf{x}} - \bar{\mathbf{x}} \ , \tag{44}$$

where $\mathbf{x}$ is the true state, $\hat{\mathbf{x}}$ is the navigation state, and $\bar{\mathbf{x}}$ is the nominal state. Let the state vectors be written as the chaser state, parameter state, and the target state. The parameter state contains sensors and actuators error states, which are nominally zero and therefore they do not need to be carried in the nominal state.

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \mathbf{x}_c^{\mathrm{T}} & \mathbf{x}_p^{\mathrm{T}} & \mathbf{x}_t^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} & (45) \\ \hat{\mathbf{x}} &= \begin{bmatrix} \hat{\mathbf{x}}_c^{\mathrm{T}} & \hat{\mathbf{x}}_p^{\mathrm{T}} & \hat{\mathbf{x}}_t^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} & (46) \\ \bar{\mathbf{x}} &= \begin{bmatrix} \bar{\mathbf{x}}_c^{\mathrm{T}} & \bar{\mathbf{x}}_p^{\mathrm{T}} & \bar{\mathbf{x}}_t^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} & (47) \end{aligned}$$

One way to change the trajectory dispersions without altering the navigation errors is to update the environment and navigation dispersions, ($\delta\mathbf{x}^+$ and $\delta\hat{\mathbf{x}}^+$), from the original dispersions ($\delta\mathbf{x}^-$ and $\delta\hat{\mathbf{x}}^-$) based on the navigation dispersions,

$$\begin{aligned} \delta\mathbf{x}^+ &= \delta\mathbf{x}^- + \mathbf{N}\mathbf{J}\delta\hat{\mathbf{x}}^- \ , & (48) \\ \delta\hat{\mathbf{x}}^+ &= \delta\hat{\mathbf{x}}^- + \mathbf{J}\delta\hat{\mathbf{x}}^- \ . & (49) \end{aligned}$$

where $\mathbf{J}$ is a matrix to be defined later in this section and $\mathbf{N}$ is defined in Equation (39). Updating the environment and navigation dispersions as outlined in Equation (48) and Equation (49) causes the updated navigation error ($\mathbf{e}^+$) (both inertial and relative) to remain unchanged and equal to the navigation error prior to the update ($\mathbf{e}^-$). That is,

$$\begin{aligned} \mathbf{e}^+ &= \mathbf{N}^T\delta\mathbf{x}^+ - \delta\hat{\mathbf{x}}^+ \\ &= \mathbf{N}^T\left[\delta\mathbf{x}^- + \mathbf{N}\mathbf{J}\delta\hat{\mathbf{x}}^-\right] - \left[\delta\hat{\mathbf{x}}^- + \mathbf{J}\delta\hat{\mathbf{x}}^-\right] \\ &= \mathbf{N}^T\delta\mathbf{x}^- - \delta\hat{\mathbf{x}}^- \\ &= \mathbf{e}^- \end{aligned} \tag{50}$$

The relative position vector is defined in the inertial frame as

$$\mathbf{r}^i_{rel} = \mathbf{r}^i_c - \mathbf{r}^i_t. \tag{51}$$

This expression for $\mathbf{r}^i_{rel}$ can be rewritten in terms of chaser and target position and velocity vectors and $\mathbf{H}_i$, a matrix used to compute the relative position and velocity vectors in the inertial frame from the inertial position and velocity vectors,

$$\mathbf{r}^i_{rel} = \mathbf{H}_i \begin{bmatrix} \mathbf{r}^i_c \\ \mathbf{v}^i_c \\ \mathbf{r}^i_t \\ \mathbf{v}^i_t \end{bmatrix}, \tag{52}$$

where

$$\mathbf{H}_i = \begin{bmatrix} \mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \end{bmatrix}_{3\times 12}. \tag{53}$$

In general, relative states are expressed in the local-vertical local-horizontal (LVLH) frame. Figure 1 illustrates the LVLH coordinate frame.

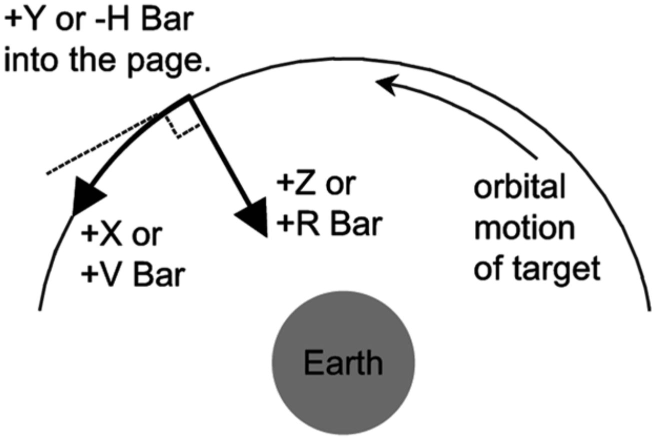


Figure 1. Local-Vertical Local-Horizontal Coordinate Frame[23]

Equation (51) can be recast so that the relative position vector is in the LVLH frame

$$\begin{aligned} \mathbf{r}^{lvlh}_{rel} &= \mathbf{T}^{lvlh}_i \mathbf{r}^i_{rel} \ , \\ &= \mathbf{H}_{lvlh} \begin{bmatrix} \mathbf{r}^i_c \\ \mathbf{v}^i_c \\ \mathbf{r}^i_t \\ \mathbf{v}^i_t \end{bmatrix}, \end{aligned} \tag{54}$$

where

$$\mathbf{H}_{lvlh} = \begin{bmatrix} \mathbf{T}^{lvlh}_i & \mathbf{0} & -\mathbf{T}^{lvlh}_i & \mathbf{0} \end{bmatrix}_{3\times 12}. \tag{55}$$

Note that $\mathbf{H}_{lvlh}$ is a matrix used to compute the relative position and velocity vectors in the LVLH frame from the inertial position and velocity vectors. The expression in Equation (55) neglects an additional term $\mathbf{E}_{lvlh}$ which is now introduced.

In the previous equations, $\mathbf{T}^{lvlh}_i$ is a 3 × 3 matrix to rotate a position or velocity from the inertial frame to LVLH frame. For the remainder of this section, $\mathbf{T}^{lvlh}_i$ will be a 6 × 6 matrix to rotate both the position and velocity of the vehicle at once. Using this new notation and assuming the chaser and target states in Equations (45 - 47) include only position and velocity, $\mathbf{H}_{lvlh}$, before adding the additional term, becomes

$$\mathbf{H}_{lvlh} = \begin{bmatrix} \mathbf{T}^{lvlh}_i & \mathbf{0} & -\mathbf{T}^{lvlh}_i \end{bmatrix} \tag{56}$$

The inertial to LVLH transformation matrix is a function of $\bar{\mathbf{r}}_t$ and $\bar{\mathbf{v}}_t$, the nominal position and velocity of the target. The symbol $\mathbf{E}_{lvlh}$ denotes the partial of Equation (54) with respect to the target position and velocity

$$\mathbf{E}_{lvlh} = \begin{bmatrix} \mathbf{r}_t^T \frac{\partial \mathbf{T}_i^{lvlh}}{\partial \mathbf{r}_t} & \mathbf{r}_t^T \frac{\partial \mathbf{T}_i^{lvlh}}{\partial \mathbf{v}_t} \\ \mathbf{v}_t^T \frac{\partial \mathbf{T}_i^{lvlh}}{\partial \mathbf{r}_t} & \mathbf{v}_t^T \frac{\partial \mathbf{T}_i^{lvlh}}{\partial \mathbf{v}_t} \end{bmatrix} . \tag{57}$$

Adding the $\mathbf{E}_{lvlh}$ to Equation (56) results in

$$\mathbf{H}_{lvlh} = \begin{bmatrix} \mathbf{T}_i^{lvlh} & \mathbf{0} & (-\mathbf{T}_i^{lvlh} + \mathbf{E}_{lvlh}) \end{bmatrix} . \tag{58}$$

Although it is possible to re-set the augmented state matrix in a variety of ways, the objective is to show how it is possible to reset the augmented state matrix such that the relative trajectory and navigation dispersions in the LVLH frame remain unchanged. In other words, it is desirable to satisfy the constraints

$$\mathbf{H}_{lvlh}\mathbf{N}^T \delta\mathbf{x}^+ = \mathbf{H}_{lvlh}\mathbf{N}^T \left[\delta\mathbf{x}^- + \mathbf{N}\mathbf{J}\delta\hat{\mathbf{x}}^-\right] , \tag{59}$$

and

$$\mathbf{H}_{lvlh}\delta\hat{\mathbf{x}}^+ = \mathbf{H}_{lvlh}\left[\delta\hat{\mathbf{x}}^- + \mathbf{J}\delta\hat{\mathbf{x}}^-\right]. \tag{60}$$

Looking at Equation (59) and Equation (60), one way to ensure the constraints are met is to force the **J** matrix to satisfy the constraint

$$\mathbf{H}_{lvlh}\mathbf{J} = \mathbf{0}_{6\times m}. \tag{61}$$

If Equation (61) is satisfied, both the relative environment and navigation dispersions will remain unchanged after the resetting. There are multiple values of **J** that could work. To select a particular **J** matrix, assume it has the form

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{CC} & \mathbf{0} & \mathbf{J}_{CT} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{TC} & \mathbf{0} & \mathbf{J}_{TT} \end{bmatrix} , \tag{62}$$

where J_{CC} is a 6×6 matrix associated with the chaser's inertial position and velocity states and J_{TT} is a 6×6 matrix corresponding to the target's inertial position and velocity states. J_{CT} and J_{TC} are terms relating the chaser and target states to one another. From Equation (61), the elements of **J** must satisfy,

$$\mathbf{T}_i^{lvlh}\mathbf{J}_{CC} + (-\mathbf{T}_i^{lvlh} + \mathbf{E}_{lvlh})\mathbf{J}_{TC} = \mathbf{0}_{6\times 6} , \tag{63}$$

and

$$\mathbf{T}_i^{lvlh}\mathbf{J}_{CT} + (-\mathbf{T}_i^{lvlh} + \mathbf{E}_{lvlh})\mathbf{J}_{TT} = \mathbf{0}_{6\times 6}. \tag{64}$$

Since there are still many possible options for **J**, an additional constraint is added which requires that the target inertial navigation dispersions become zero ($\delta\hat{\mathbf{x}}_t^i = \mathbf{0}$). From Equation (49), this constraint is satisfied if

$$\mathbf{J}_{TT} = -\mathbf{I}_{6\times 6} \tag{65}$$

and

$$\mathbf{J}_{TC} = \quad \mathbf{0}_{6\times 6}. \tag{66}$$

Using these selections and satisfying the constraints in Equation (63) and Equation (64) yields

$$\mathbf{J}_{CC} = \mathbf{0}_{6\times 6} , \tag{67}$$

and

$$\mathbf{J}_{CT} = -\mathbf{I}_{6\times 6} + \mathbf{T}_{lvlh}^i\mathbf{E}_{lvlh}. \tag{68}$$

Combining these results, a **J** matrix that both reduces the target inertial navigation dispersions while maintaining the relative trajectory dispersions and navigation errors is

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & (-\mathbf{I} + \mathbf{T}_{lvlh}^i\mathbf{E}_{lvlh}) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} . \tag{69}$$

The equation for resetting the augmented state matrix has a similar form to reshaping the augmented state matrix due to an event trigger.

$$\mathbf{P}_{aug}^{+j} = \mathbf{I}_J \mathbf{P}_{aug}^{-j} \mathbf{I}_J^T \ , \tag{70}$$

where the covariance resetting matrix $\mathbf{I}_J$ is

$$\mathbf{I}_J = \begin{bmatrix} \mathbf{I}_{n\times n} & -\mathbf{NJ} \\ \mathbf{0}_{m\times n} & \mathbf{I}_{m\times m} - \mathbf{J} \end{bmatrix} . \tag{71}$$

Substituting the expression for $\mathbf{J}$ into the resetting covariance matrix $\mathbf{I}_J$ in Equation (71), the augmented state matrix can be modified using Equation (70) to reduce the inertial trajectory dispersions while maintaining the relative trajectory dispersions and navigation errors.

B. Event Time Dispersion

An analysis of times of arrival is often useful in defining conditions beyond vehicle proximity required for a successful rendezvous. Equation (35), the difference between the true and nominal event times, can be rewritten in terms of the augmented state. The event time variation becomes

$$\begin{aligned} \delta t_e &= \mathbf{0}_{1\times n} \delta\mathbf{x}(\bar{t}_e) + \mathbf{s}^T \delta\hat{\mathbf{x}}(\bar{t}_e) \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{s} \end{bmatrix} \delta\mathbf{x}_{aug}. \end{aligned} \tag{72}$$

The variance of the event times is determined by squaring Equation (72) and taking the expected value. Specifically,

$$\sigma_{t_e}^2 = \begin{bmatrix} \mathbf{0} & \mathbf{s}^T \end{bmatrix} \mathbf{P}_{aug}^{-s} \begin{bmatrix} \mathbf{0} & \mathbf{s}^T \end{bmatrix}^T . \tag{73}$$

IV. Results

This section presents a throughout numerical comperison of the LinCov results with a Monte Carlo simulation. The results from the LinCov run using the resetting method are validated using results from the Monte Carlo simulation consisting of 1000 runs. A thorough comparison of the LinCov and Monte Carlo simulation results is completed. The two different simulations have the same measurement models and error sources. The time of arrival dispersions from LinCov and Monte Carlo are also investigated.

A. Multiple Triggers

Figure 2 shows a rendezvous trajectory with dispersions from LinCov. The target is located at the center of the LVLH frame, (0,0) on the plot. The thick blue line is the LinCov nominal trajectory and colored circles correspond to maneuvers. The red ellipses show the 3σ relative trajectory dispersions.

Figure 3 shows a closer view of the terminal phase of the approach. The results without resetting the inertial dispersions are shown in Figure 4(a). The dispersions are correctly shaped at TPI due to the downrange trigger, but during the last 600 meters of the trajectory the dispersions are very oddly shaped. Figure 4(b) shows the same portion of the trajectory after applying the resetting technique. This plot illustrates that the numerical issues associated with applying multiple event triggers to LinCov is resolved.

B. Trajectory Dispersions

The LinCov and Monte Carlo trajectories are shown in Figure 4. To validate the resetting technique for multiple event triggers, the LinCov results are compared to results of the Monte Carlo simulation. Comparisons of trajectory dispersions, navigation performance, and arrival time uncertainties are discussed. This section also represents a thorough comparison of LinCov results with the results of a Monte Carlo simulation consisting of 1000 runs. Both simulations run for approximately 12,000 seconds and the navigation filter from the Monte Carlo simulation is a realistic representation of a filter that would be included as part of the flight software for an actual mission because it only estimates a fraction of the environment and sensor states. LinCov uses Consider analysis, where only the states that exist in the Monte Carlo navigation filter are used to update the navigation dispersions.[24] Previous comparisons of LinCov and Monte Carlo results

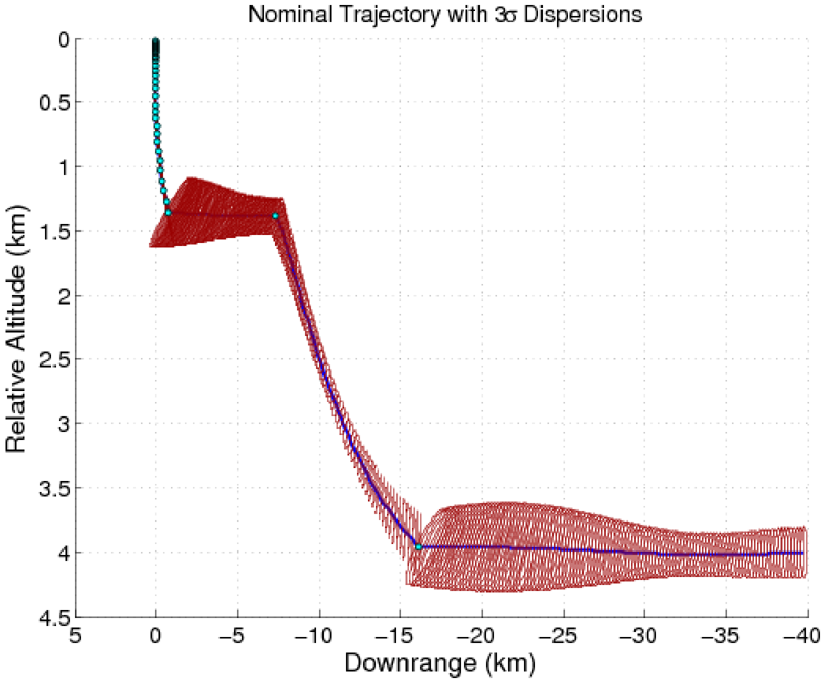
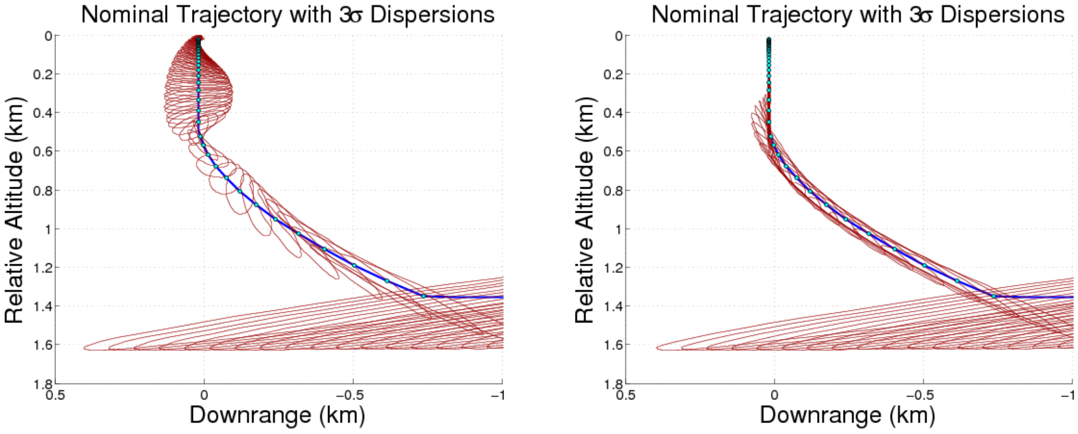


Figure 2. LinCov Nominal Trajectory with 3σ Trajectory Dispersions

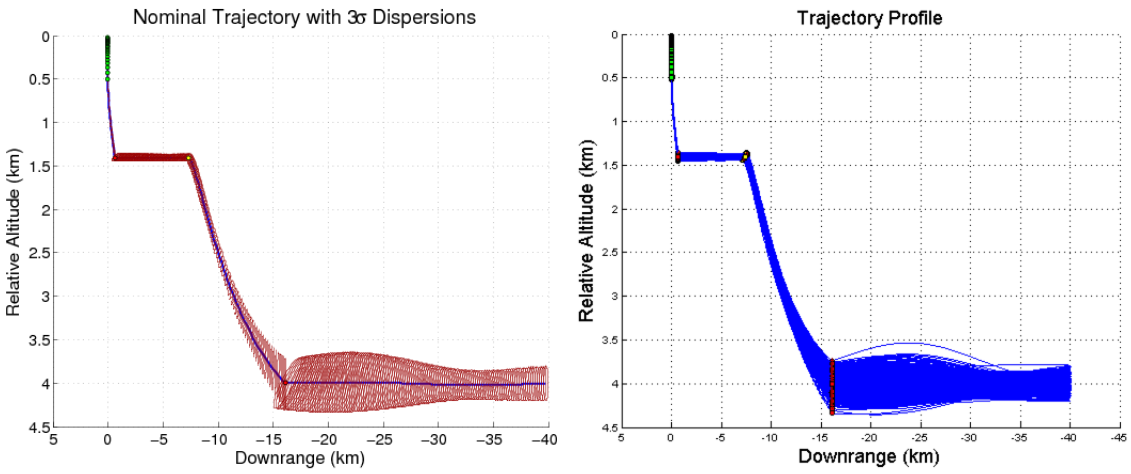


(a) Without Correction

(b) With Correction

Figure 3. Additional Comparison of Trajectory Dispersions

only included the final phase of the trajectory (the V-bar approach) and the environment and navigation models used the same dynamics and the filters estimated all of the true states.

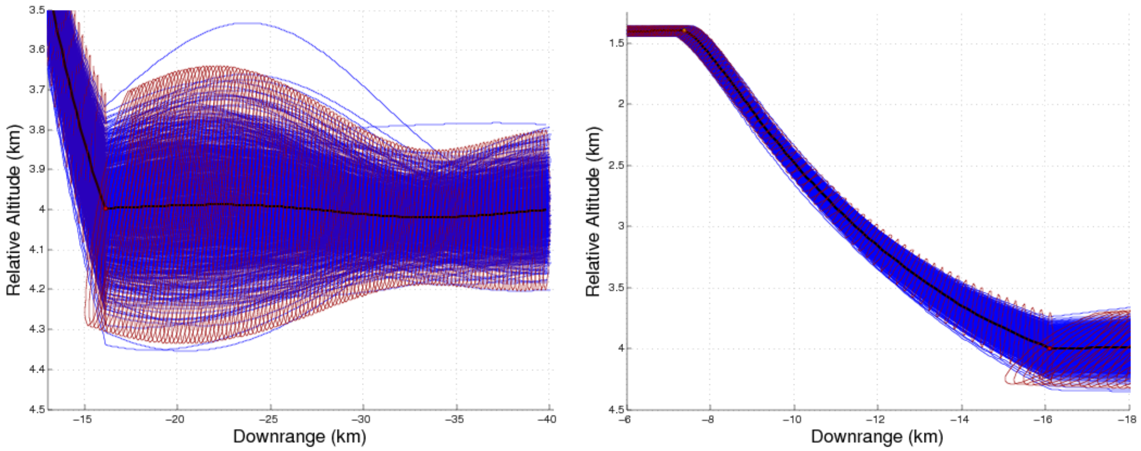


(a) LinCov

(b) Monte Carlo

Figure 4. LinCov Nominal Trajectory with 3σ Trajectory Dispersions

The previous two plots are combined and broken down into five zoomed in views of different sections for better comparison. Figure 5(a) shows the initial coelliptic phase of the trajectory. During this phase of flight, both vehicles are coasting. The 3σ trajectory dispersions from LinCov bound all but three of the Monte Carlo trajectories at any point as expected. At the first maneuver, a downrange trigger is implemented and the dispersions snap vertically. The chaser targets a position to begin a second coellptic orbit using Lambert targeting. Figure 5(b) shows the chasers transition from the first to second coelliptic orbits.

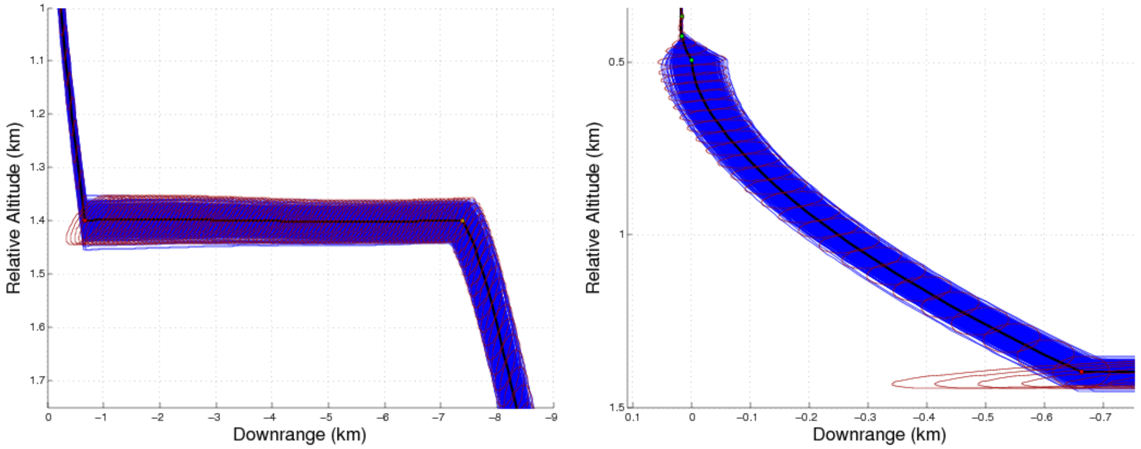


(a) First Coelliptic

(b) Transition to Second Coelliptic

Figure 5. Combined Trajectory Profile

During the first transition phase, the majority of the trajectories are within the 3σ trajectory dispersion ellipses. The four trajectories that exceed the bounds are not a concern because no more than three of the trajectories are out at any point and the trajectories return within the ellipses before the second maneuver occurs. Figure 6(a) shows the second coelliptic of the trajectory. Along this coelliptic nearly all of the trajectories are contained. Again, none of these instances are cause for concern. The dispersions are fairly constant during this section and snap vertical once again at the third maneuver, where another downrange trigger is implemented. The third maneuver serves as the TPI maneuver, propelling the chaser toward the final phase of the trajectory.



(a) Second Coelliptic

(b) Terminal Phase Initiation

Figure 6. Combined Trajectory Profile

Figure 6(b) illustrates the phase of the trajectory that sets up the chaser for the glideslope maneuvers used to move parallel to the relative altitude axis. The final plot of this series, Figure 7, shows the glideslope maneuvers. During the final approach phase of the trajectory, maneuvers occur at two minute intervals. The lidar sensor is active and returns relative measurements for range, azimuth, and elevation from the chaser to target. The dispersion ellipses fit the Monte Carlo trajectories well and show that LinCov can handle multiple event triggers. Figure 7 shows the glideslope approach the chaser uses to reach the target vehicle.

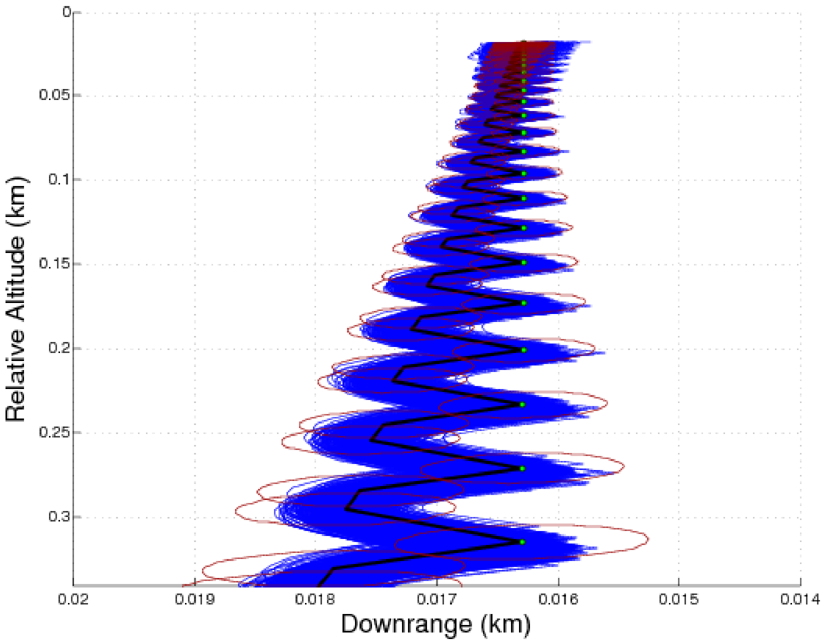


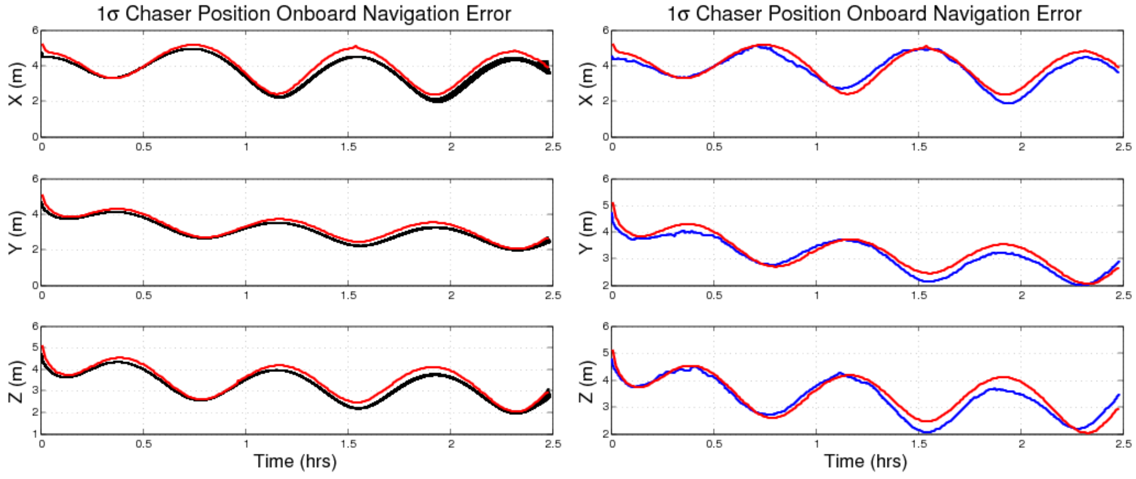
Figure 7. Combined Trajectory Profile - Glideslope

C. Navigation Performance

The navigation performance results presented in this section are unique to similar comparison studies performed in the past.[5] For example, one common technique for LinCov analysis is to assume that the onboard navigation filter states are the same as the true environment states (with the possible exception of angular rate in gyro replacement mode). In this case, no-tuning is needed and the only requirement is consistency between the dispersions noises and the navigation noises. In a more realistic scenario in which the onboard filter has fewer states than the environment, tuning becomes necessary. A perfectly tuned reduced-order filter will perform as if the non-estimated states effect was considered. To capture this affect, a consider update in the LinCov onboard filter is performed. This approach has the advantage of retaining the performance of a perfectly tuned reduced-state filter and the trivial tuning procedure of a full state filter in LinCov.

The Monte Carlo 1σ predicted navigation filter standard deviations and sample standard deviations are compared to onboard standard deviations from LinCov. For the ensuing plots, the black lines are the 1000 predicted standard deviations from the Monte Carlo simulation, the red lines are the onboard predicted standard deviations from LinCov, and the blue lines are the sample standard deviations from the Monte Carlo simulation. As shown in the previous section, the predicted and sample standard deviations from the Monte Carlo simulation are very similar. This comparison between the onboard covariance matrix from LinCov and estimation error covariance matrix from the Monte Carlo simulation is important because the filters used in both simulations are representative of flight software that would be flown aboard the chaser vehicle. The filters do not estimate all of the states that exist in the environment model.

Figure 8 compares the standard deviations for the three components of chaser inertial position navigation error. The initial covariances in both simulations are identical and the difference seen in the starting point of these plots is a result of the simulation running at different time steps. The Monte Carlo simulation runs with a time step of one second and LinCov uses a time step of thirty seconds. This occurs for the majority of comparisons.



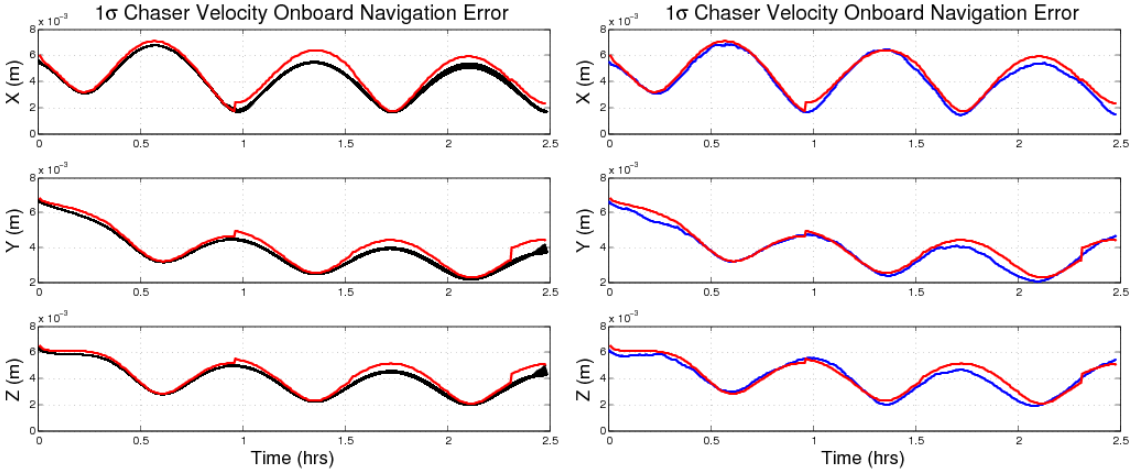
(a) Predicted Standard Deviation

(b) Sample Standard Deviation

Figure 8. Comparison of Chaser Position Navigation Error

Figure 9 shows the comparison for the chaser inertial velocity navigation error. Comparable navigation performance ensures the proper evaluation of inertial and relative targeting algorithms. The first maneuver is calculated using inertial Lambert targeting and the glideslope maneuvers use relative targeting. This comparison also shows that the navigation errors for both inertial and relative states are not affected by the resetting technique.

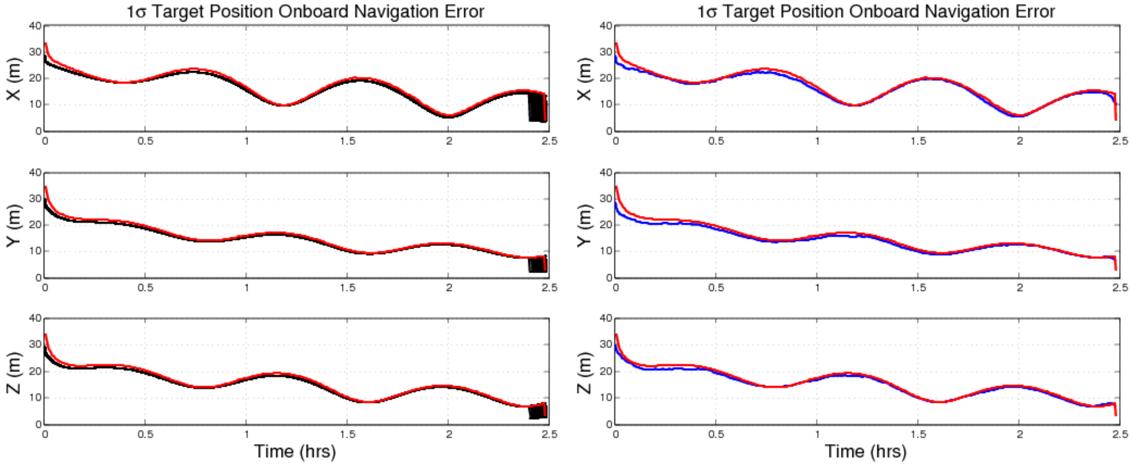
Figure 10 shows the comparison for the target position navigation error and Figure 11 shows the comparison for the target velocity navigation error.



(a) Predicted Standard Deviation

(b) Sample Standard Deviation

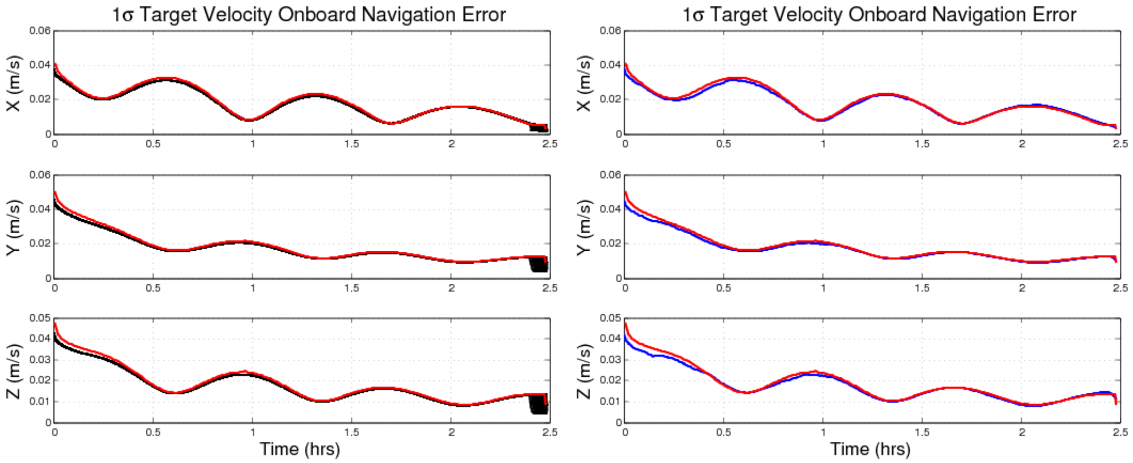
Figure 9. Comparison of Chaser Velocity Navigation Error



(a) Predicted Standard Deviation

(b) Sample Standard Deviation

Figure 10. Comparison of Target Position Navigation Error

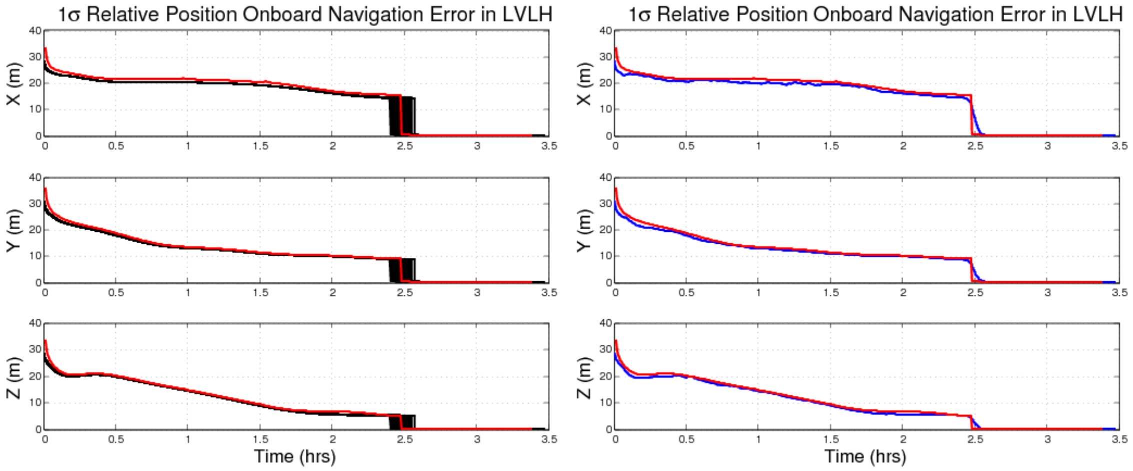


(a) Predicted Standard Deviation

(b) Sample Standard Deviation

Figure 11. Comparison of Target Velocity Navigation Error

Figure 12 shows the comparison for the relative position navigation error.

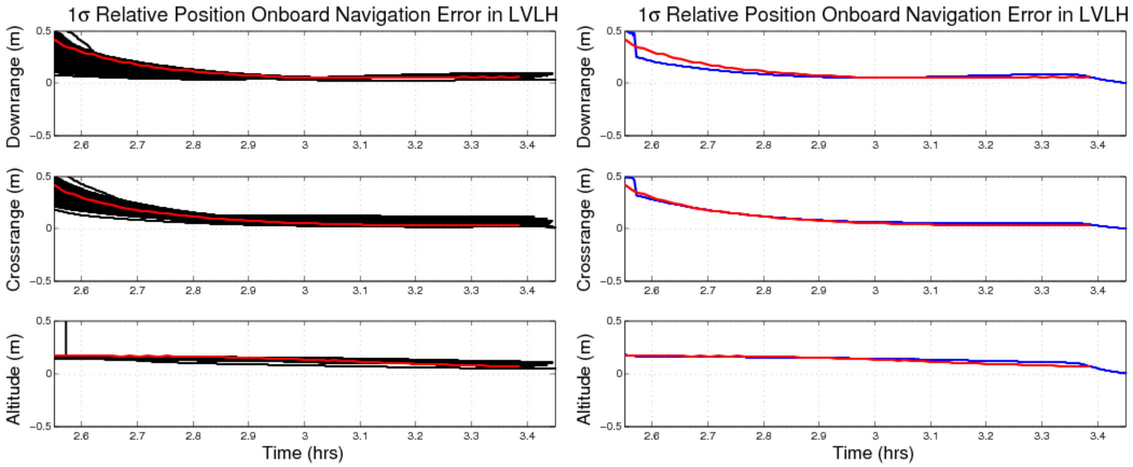


(a) Predicted Standard Deviation

(b) Sample Standard Deviation

Figure 12. Comparison of Relative Position Navigation Error

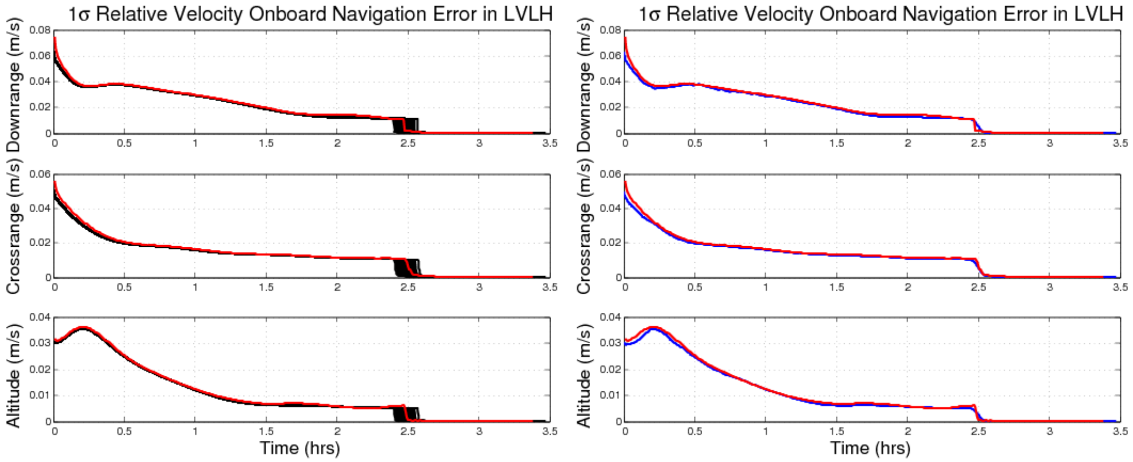
Figure 13 shows a zoomed view of Figure 12 after the lidar sensor turns on.



(a) Predicted Standard Deviation (b) Sample Standard Deviation

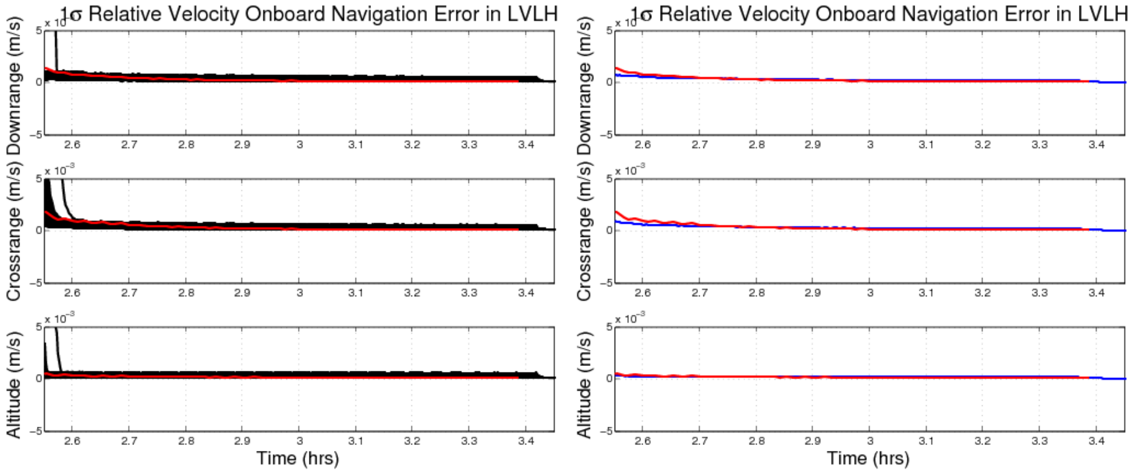
Figure 13. Zoomed View of Comparison of Relative Position Navigation Error

Figure 14 shows the comparison for the relative velocity navigation error and Figure 15 shows a zoomed view of Figure 14 after the lidar sensor turns on.



(a) Predicted Standard Deviation (b) Sample Standard Deviation

Figure 14. Comparison of Relative Velocity Navigation Error



(a) Predicted Standard Deviation

(b) Sample Standard Deviation

Figure 15. Zoomed View of Comparison of Relative Velocity Navigation Error

D. Time Dispersions

This section addresses the time dispersions associated with executing maneuvers based on event triggers rather than time. Time dispersions are calculated for the first maneuver (M1) where a downrange trigger is used and for the second downrange trigger at the third maneuver (M3). A third location, 20 meters relative altitude, is used to calculate the time dispersions at the end of the simulation. In Table 10, the LinCov time dispersions are calculated using Equation (73). This calculation finds the standard deviation of event times as though each trigger resets the simulation time to zero. The time dispersions from LinCov represent the uncertainty between two events, not between a particular event and the beginning of the simulation. The Monte Carlo time dispersions are calculated from the initial simulation time.

Simulation	M1	M3	20 m Altitude
LinCov	63.0258 s	55.0258 s	15.1857 s

Table 10. 1σ LinCov Time Dispersions

The time dispersions in Table 10 are used to calculate the total time dispersions at each event. Specifically,

$$\sigma_t = \sqrt{\sigma_{t_1}^2 + \sigma_{t_2}^2 + \dots}\,, \tag{74}$$

where σ_t is the total 1σ time dispersion, σ_{t_1} is the 1σ time dispersion between the beginning of the simulation and event 1, and σ_{t_2} is the 1σ time dispersion between event 1 and event 2. A comparison of the total time dispersions from each simulation is shown in Table 11. The differences between the two simulation values are 0.0075 seconds (0.0019 %), 0.5771 seconds (0.6850 %), and 1.15169 seconds (1.5169 %). These values are quite close considering the length of the trajectory is over 12,000 seconds.

Simulation	M1	M3	20 m Altitude
LinCov (Total)	63.0258 s	83.6715 s	85.0348 s
Monte Carlo (Total)	63.0183 s	84.2486 s	86.3482 s
$\Delta\sigma_t$	0.0075 s	0.5771 s	1.3098 s

Table 11. 1σ Time Dispersion Comparison

The correlation coefficient is the measure of the degree of linear correlation between two random variables.[24] It ranges from -1 to 1 and is defined as

$$\rho_{X_j X_k} = \frac{\mathrm{Cov}(X_j, X_k)}{\sigma_{X_j}\sigma_{X_k}}. \tag{75}$$

In this equation, $\mathrm{Cov}(X_j, X_k)$ is referred to as the covariance of X_j and X_k. When ρ is equal to zero the random variables are uncorrelated, absolutely positively correlated when ρ is +1, and absolutely negatively correlated when ρ is -1. Let Y be the sum of n random variables

$$Y = X_1 + X_2 + \ldots + X_n. \tag{76}$$

The variance of Y is

$$\sigma_Y^2 = \sum_{j=1}^{n} \sigma_{X_j}^2 + \sum_{j=1}^{n}\sum_{j=1}^{n} \rho_{X_j X_k}\sigma_{X_j}\sigma_{X_k}; \quad j \neq k, \tag{77}$$

where σ_{X_j} and σ_{X_k} are the standard deviations of X_j and X_k.[25] If there are two random variables X_1 and X_2, Equation (77) becomes

$$\sigma_Y^2 = \sigma_{X_1}^2 + \sigma_{X_2}^2 + 2\rho_{X_1 X_2}\sigma_{X_1}\sigma_{X_2}. \tag{78}$$

If X_1 and X_2 are uncorrelated the standard deviation of Y is

$$\sigma_Y = \sqrt{\sigma_{X_1}^2 + \sigma_{X_2}^2}, \tag{79}$$

or the two are absolutely positively correlated

$$\sigma_Y = \sqrt{\sigma_{X_1}^2 + \sigma_{X_2}^2 + 2\sigma_{X_1}\sigma_{X_2}}. \tag{80}$$

The calculation of total time dispersions using Equation (74) assumes that the time dispersions at each event are uncorrelated. The differences between results in Table 11 increase as the trajectory progresses, implying that some degree of correlation exists. Future work could investigate the correlation between the time dispersions at different events.

V. Conclusions

This work used linear covariance analysis and Monte Carlo simulation to apply multiple event triggers to an orbital rendezvous trajectory. Techniques have been presented to overcome the large inertial dispersions that can occur when using the reshaping method alone. The resetting method retains the unmodified relative dispersions, while resetting the inertial dispersions and preventing the numerical issues from improperly distorting the analysis results. To validate these new theoretical concepts, both a linear covariance (LinCov) and Monte Carlo simulation tools implementing two downrange triggers during a full rendezvous scenario are compared and analyzed. The sample statistics generated from 1000 Monte Carlo runs are accurately represented with the LinCov results produced with a single simulation run.

The trajectory dispersion performance comparison shows comparable results from the initial rendezvous phase starting tens of kilometers away to the final approach and docking. The trajectory dispersions reflect the integrated GNC performance of various sensors, targeting algorithms, actuators, environmental forces and torques, and initial condition uncertainties. Under such circumstances, the LinCov results incorporating the proposed multiple event trigger theory still provided similar trajectory dispersion results for the entire aspect of the simulated rendezvous scenario.

The navigation performance comparison demonstrated that the filters provide nearly identical estimates of the true states and that neither the inertial navigation errors nor the relative navigation errors are affected by the resetting method. Several aspects of the navigation performance comparison analysis presented in this paper are unique. First, a reduced-order onboard navigation filter was implemented using consider analysis in LinCov. Second, it addressed not only the final approach phase of the mission but modeled the intricacies of a full rendezvous trajectory for proximity operations.

The time dispersion analysis has provided the uncertainties in the times that events occur. The uncertainty in the time that the first event occurs is close to a minute and just under a minute and a half for the second event. This analysis has established a block of time, two minutes on either side of the nominal time, when the chaser reaches a relative altitude of twenty meters. The time of arrival predictions between the LinCov and the Monte Carlo results differed by less than two percent.

All of the results have shown that the resetting method can be successfully implemented in linear covariance analysis to account for multiple event triggers. They also have shown that LinCov can be used with a high level of confidence.

References

[1] Mayback, P. S., *Stochastic Models, Estimation, And Control Volume 1*, Mathematics in Science and Engineering, Academic Press, Orlando, FL, 1979.

[2] Geller, D. K., "Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning," *Journal of Guidance Control and Dynamics*, Vol. 29, No. 6, November-December 2006, pp. 1404–1414.

[3] Geller, D. K., Rose, M. B., and Woffinden, D. C., "Event Triggeres in Linear Covariance Analysis With Applications to Orbital Rendezvous," *Journal of Guidance Control and Dynamics*, Vol. 32, No. 1, January-February 2009, pp. 102–111.

[4] Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation*, Vol. 128 of *Mathematics in Sciences and Engineering*, Academic Press, 1978.

[5] Woffinden, D. C. and Geller, D. K., "Relative Angles-Only Navigation and Pose Estimation for Autonomous Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, September-October 2007, pp. 1455–1469.

[6] Sievers, A., *Multiple Event Triggers in Linear Covariance Analysis for Spacecraft Rendezvous*, Master's thesis, Rice University, May 2010.

[7] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, Microcosm Press, El Segundo, CA, 2004.

[8] Kuipers, J. B., *Quaternions and Rotation Sequences*, Princeton University Press, Princeton, NJ, 1999.

[9] Grubin, C., "Derivation of Quaternion Scheme via the Euler Axis and Angle," *Journal of Spacecraft and Rockets*, Vol. 7, No. 10, 1970, pp. 1261–1263.

[10] Wertz, J. R., editor, *Spacecraft Attitude Determination and Control*, Springer, 1978.

[11] Hughes, P. C., *Spacecraft Attitude Dynamics*, John Wiley and Sons, 1986.

[12] Wie, B., *Space Vehicle Dynamics and Control*, American Institute of Aeronautics and Astronautics, Reston, VA, 1998.

[13] Kaplan, M. H., *Modern Spacecraft Dynamics and Control*, John Wiley and Sons, 1976.

[14] Titterton, D. H. and Weston, J. L., *Strapdown Inertial Navigation Technology*, American Institute of Aeronautics and Astronautics, Reston, VA, 2004.

[15] Flenniken IV, W. S., Wall, J. H., and Bevly, D. M., "Characterization of Various IMU Error Sources and the Effect on Navigation Performance," *2005 ION GNSS*, Long Beach, CA, 13-16 September 2005.

[16] Larson, W. J. and Wertz, J. R., editors, *Space Mission Analysis and Design*, Microcosm Press, El Segundo, CA, 1999.

[17] Osenar, M. J., *Performance of Automated Feature Tracking Cameras for Lunar Navigation*, Master's thesis, Rice University, April 2007.

[18] Spilker Jr., J. J., Parkinson, B. W., Axelrad, P., and Enge, P., *Global Positioning System: Theory and Application*, American Institute of Aeronautics and Astronautics, Reston, VA, 1996.

[19] Kremer, A. S., *Linear Covariance Analysis Trade Study of Autonomous Navigation Schemes for Cislunar Missions*, Master's thesis, Massachusetts Institute of Technology, June 2007.

[20] Geller, D. K., "Orbital Rendezvous: When Is Autonomy Required?" *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, July-August 2007, pp. 974–981.

[21] Chambers, R. P., "Seven Spacecraft in One - Orion Guidance, Navigation, and Control," *AIAA Space 2008 Conference & Exposition*, San Diego, CA, 9-11 September 2008.

[22] Bryson Jr., A. E., *Control of Spacecraft and Aircraft*, Princeton University Press, Princeton, NJ, 1994.

[23] Goodman, J. L., "History of Space Shuttle Rendezvous and Proximity Operations," *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, September-October 2006, pp. 944–959.

[24] Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier Academic Press, Burlington, MA, 2004.

[25] Elishakoff, I., *Probabilistic Theory of Structures*, Dover Publications, Mineola, NY, 1999.