A Systems Approach to Modeling Uncertain ODE Systems

Benjamin W. L. Margolis, Kenneth R. Lyons, Carlos A. Natividad
NASA Ames Research Center, Moffett Field, CA, 94035

Collin E. York, David Woffinden
NASA Johnson Space Center, Houston TX, 77058

Michaela L. Tarpley
Amentum Space Exploration Division, Huntsville, AL 35806

Sagar Bhatt, Stephen R. Steffes
The Charles Stark Draper Laboratory, Inc., Cambridge, MA 02139

NASA needs to perform uncertainty analysis of mission performance. One way to perform uncertainty analysis is to use linear covariance techniques. Uncertainty analysis is an important part of all phases of every NASA mission. This type of analysis helps to discover risks, optimize parameters, and validate future success of a system. In this paper, modeling uncertainty with additive zero-mean white noise on process and system outputs is described from a systems modeling approach. Core block diagram algebra rules for uncertain systems that can be used to generalize linear covariance analysis and systematically implement the covariance equations in NASA's Swift Integrated GNC and Mission Analysis (SIGMA) tool. This framework is applied to a rendezvous scenario with a dual inertial physical model and relative GNC systems, demonstrating its flexibility.

I. Introduction

Incorporating the impacts of system uncertainty such as navigation errors, disturbance accelerations, burn execution biases and misalignments, orbit insertion dispersions, and a host of other influential factors into the design of a trajectory profile to make it robust to these unknown perturbations is becoming more feasible, particularly with the maturation of linear covariance analysis (LCA)[1, 2] for closed-loop guidance, navigation, and control (GNC) systems. LCA is an analytic technique for predicting stochastic behavior of dynamical systems. Previous studies interfaced a genetic algorithm (GA) with a linear covariance analysis tool to provide the initial demonstration of applying this non-traditional approach to robust trajectory optimization. These robust trajectory design techniques were originally developed for rendezvous applications in low Earth orbit[3]. It was then extended to cis-lunar outbound trajectories to a Near Rectilinear Halo Orbit (NRHO)[4] and introduced for a simple rendezvous approach trajectory in the NRHO for mid-course correction placement[5]. Recently, these robust trajectory optimization principles have been applied to solve cislunar transfers to low-lunar orbit[6], NRHO rendezvous and docking[7], angles-only navigation for NRHO rendezvous [8], lunar powered descent and landing[9], along with Mars aerocapture[10] problems. They were also utilized to determine the optimized trajectory correction burn placement for the upcoming Artemis II free-return cis-lunar trajectory profile[11] and exercised to determine the optimized trajectory correction burn placement for NRHO orbit maintenance anticipated for the Artemis III mission[12]. LCA has also been used with a sweeping gradient method [13] to perform robust optimal trajectory design for rendezvous in low-earth orbits and near rectilinear halo orbits[14, 15].Recent developments in techniques [16, 17] generalize some of the modeling concepts and derive the corresponding governing equations.

In this work, we present the software architecture for the Swift Integrated GNC and Mission Analysis (SIGMA) tool. In existing tools, the linear covariance analysis governing equations for particular models were implemented manually, involving significant development effort to derive and code partial derivatives and ensure proper bookkeeping for a variety of scenario configurations. SIGMA takes a different approach, modeling scenarios by connecting individual system models representing physical or GNC subsystems. By leveraging Condor [18] for automatic differentiation and an engineering-focused modeling language, SIGMA is able to generate the functions required to perform covariance analysis. We have found that this reduces the number of configuration-specific special cases and reduces software development for performing analysis.

The rest of the paper is organized as follows. In the next section, we review the scenario models used for analyzing closed-loop GNC systems and review the linear covariance analysis techniques and the core governing equations. Then we re-derive these equations from a block-diagram algebra. Finally, we describe a rendezvous scenario that is implemented in SIGMA.

II. Core Systems and Linear Covariance Analysis Equations of Motion

Typical formulations for linear covariance analysis specify the true (physical or environmental) system dynamics and GNC system dynamics. Here we provide an overview of the governing equations. For additional details see [2], whose notation is reproduced in this section. In the following section, new notation is adopted to reflect the new block-diagram formulation for modeling GNC systems.

A. Physical System

The true state $x(t) \in \mathbb{R}^{n_x}$ is used to model the governing system dynamics, both continuous and discrete sensor measurements, and impulsive state corrections. These models are provided in the context of establishing the core LCA equations. The form of these physical models are given by the true dynamics,

$$\dot{x} = f(x, \hat{u}, t) + w$$

where $\hat{u}(t)$ is a commanded control input and $w(t)$ is zero-mean white noise with covariance $S_w(t)\,\delta(t - t')$.

If continuous measurements such as gyros and accelerometers are utilized, the measured value (e.g., angular rates from gyros and accelerations from accelerometers) are modeled as

$$\tilde{y} = c(x, t) + \eta$$

where $\eta(t)$is zero-mean white noise with covariance $S_\eta(t)\,\delta(t - t')$.

Discrete inertial and non-inertial measurements are modeled as

$$\Delta\tilde{y}_j = \Delta c\left(x_j, t_j\right) + \Delta\eta_j$$
$$\tilde{z}_k = h\left(x_k, t_k\right) + \nu_k$$

respectively, with zero-mean white noise $\Delta\eta_j$ and ν_k with covariances $S_{\Delta\eta}\left(t_j\right)\delta_{jj'}$ and $R_\nu(t_k)\,\delta_{kk'}$.

Impulsive control inputs are modeled as instantaneous corrections according to the mapping,

$$x_j^{+c} = x_j^{-c} + d\left(x_j^{-c}, \Delta\hat{u}_j, t_j\right) + \Delta w_j$$

where $\Delta\hat{u}_j$ is an impulsive control command, Δw_j is a zero-mean correction noise with covariance $S_{\Delta w}\left(t_j\right)\delta_{jj'}$, and the superscripts $-c$ and $+c$ indicate the state value immediately before and after the jth impulsive maneuver.

B. GNC System

The GNC system has an estimated state $\hat{x}(t) \in \mathbb{R}^{n_{\hat{x}}}$ which estimates a mapping of the true state, $\hat{x} \leftarrow\!\shortmid\, m(x)$. For an extended Kalman filter, the prediction or design model dynamics is used to propagate the estimated state according to

$$\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}, \tilde{y}, t)\,.$$

The estimated state is used to calculate the continuous control input,

$$\hat{u} = \hat{g}(\hat{x}, t)\,,$$

which is applied to both the physical and filter models. We note that it is often helpful to distinguish between a commanded control from the control law, $\breve{u} = \hat{g}(\hat{x}, t)$ and the u and $\hat{u}$ that is applied to velocity-like state variables for the physical and estimated state, respectively, according to a physical and estimated error model for the actutator.

The filter's estimated error is represented by the covariance $\hat{P}(t)$ which is governed by the propagation equations,

$$\dot{\hat{P}} = \left[\hat{F}_{\hat{x}} + \hat{F}_{\tilde{y}}\hat{C}_{\hat{x}}\right]\hat{P} + \hat{P}\left[\hat{F}_{\hat{x}} + \hat{F}_{\tilde{y}}\hat{C}_{\hat{x}}\right] + \hat{F}_{\tilde{y}}\hat{S}_\eta\hat{F}_{\tilde{y}}^T + \hat{S}_w$$

where we use upper-case decorated letters with subscripts to denote the Jacobian of the function of the same letter in lower-case with respect to the variable in the subscript, for example $\hat{F}_{\hat{x}} = \partial \hat{f}/\partial \hat{x}$. The symbols of the form $S_{(\cdot)}$ are the covariance matrices for the random variable denoted in the subscript.

When discrete measurement updates are processed by the navigation filter, the state and covariance matrix are instantly updated using the Kalman gain, $\hat{K}$, so that

$$\begin{aligned}
\hat{x}_k^+ &= \hat{x}_k^- + \hat{K}\left(t_k\right)\left[z_k - \hat{h}\left(\hat{x}_k, t_k\right)\right], \\
\hat{P}\left(t_k^+\right) &= \left[I - \hat{K}\left(t_k\right)\hat{H}_{\hat{x}}\left(t_k\right)\right]\hat{P}\left(t_k^-\right)\left[I - \hat{K}\left(t_k\right)\hat{H}_{\hat{x}}\left(t_k\right)\right]^T + \hat{K}\left(t_k\right)\hat{R}_\nu \hat{K}^T\left(t_k\right), \text{ and} \\
\hat{K}\left(t_k\right) &= \hat{P}\left(t_k\right)\hat{H}_{\hat{x}}^T\left(t_k\right)\left[\hat{H}_{\hat{x}}\left(t_k\right)\hat{P}\left(t_k\right)\hat{H}_{\hat{x}}^T\left(t_k\right) + \hat{R}_\nu\left(t_k\right)\right]^{-1},
\end{aligned}$$

where $\hat{h}\left(\hat{x}_k, t_k\right)$ is the predicted measurement model and $\hat{R}_\nu$ is the predicted measurement noise covariance.

Impulsive control commands are also functions of the estimated state,

$$\Delta\hat{u}_j = \Delta\hat{g}\left(\hat{x}_j^{-c}, t_j\right)$$

and applied to the physical impulsive control model and a predicted impulsive control model of the form

$$\hat{x}_j^{+c} = \hat{x}_j^{-c} + \hat{d}\left(\hat{x}_j^{-c}, \Delta\hat{u}_j, \Delta\tilde{y}_j, t_j\right)$$

with corresponding estimated error update

$$\begin{aligned}
\hat{P}\left(t_j^{+c}\right) = &\left[I + \hat{D}_{\hat{x}}\left(t_j\right) + \hat{D}_{\Delta\tilde{y}}\left(t_j\right)\Delta\hat{C}_{\hat{x}}\left(t_j\right)\right]\hat{P}\left(t_j^{-c}\right)\left[I + \hat{D}_{\hat{x}}\left(t_j\right) + \hat{D}_{\Delta\tilde{y}}\left(t_j\right)\Delta\hat{C}_{\hat{x}}\left(t_j\right)\right]^T \\
&+ \hat{D}_{\Delta\tilde{y}}\left(t_j\right)\hat{S}_{\Delta\eta}\hat{D}_{\Delta\tilde{y}}^T\left(t_j\right) + \hat{S}_{\Delta w}\left(t_j\right)
\end{aligned}$$

where $\hat{S}_{\Delta\eta}$ and $\hat{S}_{\Delta w}\left(t_j\right)$ are the predicted discrete measurement noise and impulsive correction noise covariances, respectively.

C. Augmented Covariance Governing Equations

The physical state and estimated state are combined to form the augmented state,

$$X\left(t\right) = \left[x^T\left(t\right), \hat{x}^T\left(t\right)\right]^T \tag{1}$$

which would be evaluated directly with the governing equations above in a Monte Carlo simulation. In LCA, the governing equations above are linearized about the mean state

$$\overline{X} = \left[\overline{x}^T\left(t\right), m^T\left(\overline{x}\right)\right]^T$$

where it is assumed that the mean estimated state can be mapped from the mean physical state. Then the augmented state dispersions are given by

$$\delta X \triangleq X - \overline{X}$$

with augmented state dispersion covariance matrix,

$$C = E\left[\delta X\, \delta X^T\right].$$

By linearizing the dynamics about the mean, the augmented covariance matrix can be propagated by the equation

$$\dot{C} = \mathcal{F}C + C\mathcal{F}^T + \mathcal{G}S_\eta\mathcal{G}^T + \mathcal{W}S_w\mathcal{W}^T \tag{2}$$

where

$$\mathcal{F} = \begin{bmatrix} F_x & F_{\hat{u}}\hat{G}_{\hat{x}} \\ \hat{F}_{\tilde{y}}C_x & \hat{F}_{\hat{x}} + \hat{F}_{\hat{u}}\hat{G}_{\hat{x}} \end{bmatrix}, \quad \mathcal{G} = \begin{bmatrix} 0 \\ \hat{F}_{\tilde{y}} \end{bmatrix}, \text{ and } \quad \mathcal{W} = \begin{bmatrix} I \\ 0 \end{bmatrix}$$

with initial conditions

$$C\left(t_0\right) = \begin{bmatrix} D_{\text{true}} & D_{\text{true}}M_x^T \\ M_x D_{\text{true}} & M_x D_{\text{true}} M_x^T + P_{\text{true}} \end{bmatrix} \tag{3}$$

where D_{true} is the initial dispersion covariance of the physical state about its mean, P_{true} is the initial error covariance of the estimated state from the true state, and M_x is the partial derivative of the navigation state mapping with respect to the true state.

Measurement updates cause instantaneous changes to the augmented covariance matrix of the form

$$C\left(t_k^+\right) = \mathcal{A}_k C\left(t_k^-\right) \mathcal{A}_k^T + \mathcal{B}_k R_\nu\left(t_k\right) \mathcal{B}_k^T$$

where

$$\mathcal{A}_k = \begin{bmatrix} I & 0 \\ \hat{K}\left(t_k\right) H_x\left(t_k\right) & I - \hat{K}\left(t_k\right) \hat{H}_{\hat{x}}\left(t_k\right) \end{bmatrix} \quad \text{and} \quad \mathcal{B}_k = \begin{bmatrix} 0 \\ \hat{K}\left(t_k\right) \end{bmatrix} \quad .$$

Similarly, impulsive corrections cause instantaneous changes to the augmented covariance matrix of the form

$$C\left(t_j^{+c}\right) = \mathcal{D}_j C\left(t_j^{-c}\right) \mathcal{D}_j^T + \mathcal{M}_j S_{\Delta\eta}\left(t_j\right) \mathcal{M}_j^T + \mathcal{N}_j S_{\Delta w}\left(t_j\right) \mathcal{N}_j^T$$

where

$$\mathcal{D}_j = \begin{bmatrix} I + D_x\left(t_j\right) & D_{\Delta\hat{u}}\left(t_j\right) \Delta\hat{G}_{\hat{x}}\left(t_j\right) \\ \hat{D}_{\Delta\bar{y}}\left(t_j\right) \Delta C_x\left(t_j\right) & I + \hat{D}_{\hat{x}}\left(t_j\right) + \hat{D}_{\Delta\hat{u}}\left(t_j\right) \Delta\hat{G}_{\hat{x}}\left(t_j\right) \end{bmatrix}, \quad \mathcal{M}_j = \begin{bmatrix} 0 \\ \hat{D}_{\Delta\bar{y}}\left(t_j\right) \end{bmatrix}, \text{ and } \quad \mathcal{N}_j = \begin{bmatrix} I \\ 0 \end{bmatrix} \quad .$$

In implementation, the mean true state $\overline{x}\left(t\right)$ is propagated with mean dynamics

$$\dot{\overline{x}}\left(t\right) = f\left(\overline{x}, \overline{u}, t\right)$$

where the mean control input is determined by substituting the estimated state $\hat{x}$ with the mapped value of the mean,

$$\overline{u}\left(t\right) = \hat{g}\left(m\left(\overline{x}\right), t\right).$$

Similarly, the mean impulsive control command and update are given by

$$\begin{aligned} \Delta\overline{u}\left(t_j\right) &= \Delta\hat{g}\left(m\left(\overline{x}_j^{-c}\right), t\right) \\ \overline{x}_j^{+c} &= \overline{x}_j^{-c} + d\left(\overline{x}_j^{-c}, \Delta\overline{u}_j, t_j\right). \end{aligned}$$

Then the estimated navigation error $\hat{P}$ and augmented covariance matrix C can follow the governing equations identified above. At measurement updates, which do not affect the physical state, the mapped mean true state $m\left(\overline{x}\right)$ can be used anywhere the estimated state is needed. Additional analysis methods, like stochastic navigation [19] or event triggers [20, 21] are also important to effectively model flight missions.

D. Generalized Augmented-State Covariance Analysis

In [16], a generalized augmented-state covariance analysis (GAUSCOV) technique is formulated to support modeling relevant GNC operations like distinct targeting and maneuver execution events with temporal separation. The core effort is in the expansion of the augmented state (1) to include reference state, arbitrary groupings of estimated state, and static parameters for storing the stochastic result of GNC events like targeting or maneuver execution. In particular, these static GNC parameters enable the evaluation of key performance metrics while modeling impactful operational choices. For example, the estimated impulsive velocity change calculated from the targeting algorithm will differ from the true velocity change experienced by the vehicle due to factors including actuator error and the evolution of difference between the navigation state and true state between measurements; by explicitly formulating the generalized augmented state with these distinct categories, GAUSCOV can be used to evaluate the degree and consequence of these differences. A goal of the effort for this work is to include the additional modeling capability enabled by this generalization.

III. Systems Approach to Implementation (A Block Diagrams Algebra For Uncertain Systems)

Figure 1 shows an abstract block diagram for the components described in the previous section. The block diagram illustrates a core continuous feedback loop:

- dynamics system (true and GNC design model) to continuous sensors (true and GNC design model)

- sensors to GNC system ("filter")
- GNC system to control law (continuous targeting)
- control law to continuous actuator (true and GNC design model)
- actuators (true and GNC design model) to dynamics models (true and GNC design model, respectively)

The block diagram also illustrates several discrete event loops:

- discrete measurement from dynamics model (true and GNC design model) to sensor (true and GNC design model, respectively) to GNC system ("filter") navigation state update
- discrete targeting from GNC system navigation filter
- impulsive actuation/maneuver execution model (true and GNC design model)

To implement this, a systems approach can be used by treating each block as a stand-alone system. For an ith deterministic system, we have state x_i, input u_i, and static parameters p. The initial conditions, state equation, and output equation for this ith system are given by

$$\begin{aligned} x_i\left(t_0\right) &= \xi_i\left(t_0; p\right) \\ \dot{x}_i\left(t\right) &= f_i\left(t, x_i\left(t\right), u_i\left(t\right); p\right) \\ y_i\left(t\right) &= h_i\left(t, x_i\left(t\right), u_i\left(t\right); p\right) \end{aligned}$$

where the parameters after the semicolon remind the reader that the parameters do not vary with time. The ith system may also have arbitrarily many events, such that at the jth event we perform a state update and discrete output

$$\begin{aligned} x_i\left(t_j^+\right) &= \phi_{i,j}\left(t_j^-, x\left(t_j^-\right), u\left(t_j^-\right), \Delta u_{i,j}; p\right) \text{ and} \\ \Delta y_{i,j} &= \Delta h_{i,j}\left(t_j^-, x\left(t_j^-\right), u\left(t_j^-\right), \Delta u_{i,j}; p\right). \end{aligned}$$

Note that the discrete event output $\Delta y_{i,j}$ can be used to model discrete measurements or impulsive thruster outputs, but is ultimately a formal signal that maps into a state update $\phi_{i,j}$ through Δu. Methods for connecting the continuous input and output of deterministic signals are well established [22], and these approaches can be used to model inter-connected events. In the remainder of this section, we define the system equations for uncertain systems and identify the block-diagram algebra rules to derive the linear covariance equations due to system connections.

A. Block Definitions

Modeling dynamics with uncertainty can be achieved by adding a noise term to each of those equations. With zero-mean white noise associated with the initial dispersion σ_i, process w_i, and output v_i, the core equations for an uncertain system become

$$\begin{aligned} x_i\left(t_0\right) &= \xi_i\left(t_0; p\right) + \sigma_i, \\ \dot{x}_i\left(t\right) &= f_i\left(t, x_i, u_i; p\right) + w_i, \text{ and} \\ y_i\left(t\right) &= h_i\left(t, x_i, u_i; p\right) + v_i. \end{aligned}$$

Analogous zero-mean white noise associated with the state update $\omega_{i,j}$ and discrete output $\nu_{i,j}$ for a jth event yield update and event output equations

$$\begin{aligned} x_i\left(t_j^+\right) &= \phi_{i,j}\left(t_j^-, x_i\left(t_j^-\right), u_i\left(t_j^-\right), \Delta u_{i,j}; p\right) + \omega_{i,j} \text{ and} \\ \Delta y_{i,j} &= \Delta h_{i,j}\left(t_j^-, x_i\left(t_j^-\right), u_i\left(t_j^-\right), \Delta u_{i,j}; p\right) + \nu_{i,j}. \end{aligned}$$

B. Uncertain Block Diagram Algebra During Continuous Propagation

1. Interconnections

During continuous propagation, we can consider the "super system", with state $x\left(t\right)$ and output $y\left(t\right)$ consisting of the concatenation of the individual subsystem state $x_i\left(t\right)$ and output $y_i\left(t\right)$. These systems are interconnected through the connection mapping function $g\left(t, y\right)$. The overall system behavior for the super system is defined by



Fig. 1 Block diagram for a typical linear covariance analysis scenario with physical and navigation system models

$$\dot{x}(t) = f(t, x(t), g(t, y); p) + w$$
$$y(t) = h(t, x(t), g(t, y); p) + v$$

2. Linearization

We can linearize the interconnected system about its mean state,

$$\dot{x}(t) = f(t, \overline{x}(t) + \delta x_i(t), g(\overline{y} + \delta y); p) + w$$
$$y(t) = h(t, \overline{x}(t) + \delta x_i(t), g(\overline{y} + \delta y); p) + v$$

producing a mean system

$$\dot{\overline{x}}(t) = f(t, \overline{x}(t), g(\overline{y}(t)); p)$$
$$\overline{y}(t) = h(t, \overline{x}(t), g(\overline{y}(t)); p)$$

with a linearization defined by

$$\delta\dot{x}(t) = \left.\frac{\partial f}{\partial x}\right|_{\overline{x}} \delta x + \left.\frac{\partial f}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}} \delta y + w$$
$$\delta y(t) = \left.\frac{\partial h}{\partial x}\right|_{\overline{x}} \delta x + \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}} \delta y + v$$

Ignoring the difficulty of resolving the potential loop for the mean dynamics (i.e., using a specialized differential-algebraic equation solver if needed), we can solve the output equations in terms of the linearization about the state δx and output noise as

$$\begin{aligned}\delta y(t) &= \left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-1} \left(\left.\frac{\partial h}{\partial x}\right|_{\overline{x}} \delta x + v\right) \\ &= \left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-1} \left.\frac{\partial h}{\partial x}\right|_{\overline{x}} \delta x + \underbrace{\left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-1} v}_{\text{supersystem output noise } \tilde{v}}\end{aligned}$$

This can be substituted into the linearized state dynamics to find

$$\begin{aligned}\delta\dot{x}(t) &= \left.\frac{\partial f}{\partial x}\right|_{\overline{x}} \delta x + \left.\frac{\partial f}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}} \left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-1} \left.\frac{\partial h}{\partial x}\right|_{\overline{x}} \delta x + \underbrace{\left.\frac{\partial f}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}} \tilde{v} + w}_{\text{supersystem process noise } \tilde{w}} \\ &= \left[\left.\frac{\partial f}{\partial x}\right|_{\overline{x}} + \left.\frac{\partial f}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}} \left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-1} \left.\frac{\partial h}{\partial x}\right|_{\overline{x}}\right] \delta x + \tilde{w}\end{aligned}$$

3. Covariance propagation

Applying the linearization to the covariance propagation yields

$$\dot{C} = \mathcal{F}C + C\mathcal{F}^T + S_{\tilde{w}} \tag{4}$$

where

$$\mathcal{F} = \left(\left.\frac{\partial f}{\partial x}\right|_{\overline{x}} + \left.\frac{\partial f}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}} \left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-1} \left.\frac{\partial h}{\partial x}\right|_{\overline{x}}\right) \text{ and} \tag{5}$$

$$S_{\tilde{w}} = S_w + \left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-1} S_v \left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}} \left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-T}. \tag{6}$$

This has the same form as (2) but with the distinct sources of process noise combined into single term that captures the process and output noise as it propagates through the block diagram connections. We also note that the augmented-state covariance Jacobian $\mathcal{F}$ in (5) is calculated through the symbolic differentiation capability exposed by Condor and applied to the system's concatenated state equations with connection maps.

C. Uncertain Block Diagram Algebra for Events

1. Interconnections

As for the continuous case, we assume input and event-input mappings of the form $u_i = g_i(y_k)$ and $\Delta u_{i,j} = \Delta g_{i,j}(\Delta y_{k,j})$ so the overall event is defined by

$$x\left(t_j^+\right) = \phi_j\left(t_j^-, x\left(t_j^-\right), g\left(y\left(t_j^-\right)\right), \Delta g_j\left(\Delta y_j\right); p\right) + \omega_j \text{ and}$$
$$\Delta y_j = \Delta h_j\left(t_j^-, x\left(t_j^-\right), g\left(y\left(t_j^-\right)\right), \Delta g_j\left(\Delta y_j\right); p\right) + \nu_j,$$

where we note that any subsystem may have a null event with no event output and continuity of state across the event, e.g., $x_i\left(t_j^+\right) = x_i\left(t_j^-\right)$.

2. Linearization

This leads to the mean event given by

$$\overline{x}\left(t_j^+\right) = \phi_j\left(t_j^-, \overline{x}\left(t_j^-\right), g\left(\overline{y}\left(t_j^-\right)\right), \Delta g_j\left(\overline{\Delta y_j}\right); p\right) + \omega_j \text{ and}$$
$$\overline{\Delta y_j} = \Delta h_j\left(t_j^-, \overline{x}\left(t_j^-\right), g\left(\overline{y}\left(t_j^-\right)\right), \Delta g_j\left(\overline{\Delta y_j}\right); p\right) + \nu_j.$$

To linearize, we expand the event output first,

$$\delta\Delta y_j = \frac{\partial \Delta h_j}{\partial x}\delta x + \frac{\partial h_j}{\partial u}\frac{\partial g}{\partial y}\delta y + \frac{\partial \Delta h_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\delta\Delta y_j + \nu_j,$$

which we can solve for $\delta\Delta y_j$ as

$$\begin{aligned}\delta\Delta y_j &= \left(I - \frac{\partial \Delta h_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\right)^{-1}\frac{\partial \Delta h_j}{\partial x}\delta x + \left(I - \frac{\partial \Delta h_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\right)^{-1}\frac{\partial \Delta h_j}{\partial u}\frac{\partial g}{\partial y}\delta y + \left(I - \frac{\partial \Delta h_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\right)^{-1}\nu_j \\ &= \left[\left(I - \frac{\partial \Delta h_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\right)^{-1}\frac{\partial \Delta h_j}{\partial x} + \cdots\right]\delta x + \left(I - \frac{\partial \Delta h_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\right)^{-1}\frac{\partial \Delta h_j}{\partial u}\frac{\partial g}{\partial y}\tilde{v} + \left(I - \frac{\partial \Delta h_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\right)^{-1}\nu_j\end{aligned}$$

where the δy term is the same as the continuous case, ultimately a function of δx and v. The linearized update equation is

$$\delta x\left(t_j^+\right) = \frac{\partial \phi_j}{\partial x}\delta x + \frac{\partial \phi_j}{\partial u}\frac{\partial g}{\partial y}\delta y + \frac{\partial \phi_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\delta\Delta y_j + \omega_j.$$

We can substitute for δy and $\delta\overline{\Delta y_j}$ in terms of δx, ν_j, and v if desired. The resulting noise term becomes

$$\delta x\left(t_j^+\right) = \mathcal{A}_j\delta x + \frac{\partial \phi_j}{\partial u}\frac{\partial g}{\partial y}\tilde{v} + \frac{\partial \phi_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\tilde{\nu_j} + \omega_j$$

where

$$\mathcal{A}_j = \frac{\partial \phi_j}{\partial x} + \frac{\partial \phi_j}{\partial u}\frac{\partial g}{\partial y}\left(I - \left.\frac{\partial h}{\partial u}\right|_{\overline{x}}\left.\frac{\partial g}{\partial y}\right|_{\overline{x}}\right)^{-1}\left.\frac{\partial h}{\partial x}\right|_{\overline{x}} + \frac{\partial \phi_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\left(I - \frac{\partial \Delta h_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\right)^{-1}\frac{\partial \Delta h_j}{\partial x}. \tag{7}$$

We also note that the augmented-state covariance update Jacobian $\mathcal{A}_j$ in (7), like the state Jacobian, is calculated through the symbolic differentiation capability exposed by Condor and applied to the event's state update equations with connection maps.

3. Covariance Update

The corresponding covariance update then takes the form

$$C\left(t_j^+\right) = \mathcal{A}_j C\left(t_j^-\right) \mathcal{A}_j^T + S_{\tilde{\omega}_j} \tag{8}$$

where

$$S_{\tilde{\omega}_j} = \frac{\partial \phi_j}{\partial u}\frac{\partial g}{\partial y} S_v \left[\frac{\partial \phi_j}{\partial u}\frac{\partial g}{\partial y}\right]^T + \frac{\partial \phi_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j} S_v \left[\frac{\partial \phi_j}{\partial \Delta u_j}\frac{\partial \Delta g_j}{\partial \Delta y_j}\right]^T + S_{\omega_j}. \tag{9}$$

D. Implementation Details

For a parent system, we have mean dynamics

$$\overline{x}\left(t_0\right) = \xi\left(t_0; p\right), \tag{10}$$

$$\dot{\overline{x}}\left(t\right) = f\left(t, \overline{x}\left(t\right), g\left(\overline{y}\left(t\right)\right); p\right), \text{ and} \tag{11}$$

$$\overline{y}\left(t\right) = h\left(t, \overline{x}\left(t\right), g\left(\overline{y}\left(t\right)\right); p\right), \tag{12}$$

where we assume the algebraic loop caused by the feedthrough of the output equation is resolved analytically or numerically. A jth event has mean event update and event output equations

$$\overline{x}\left(t_j^+\right) = \phi_j\left(t_j^-, \overline{x}\left(t_j^-\right), g\left(\overline{y}\left(t_j^-\right)\right), \Delta g_j\left(\overline{\Delta y_j}\right); p\right) \text{ and} \tag{13}$$

$$\overline{\Delta y_j} = \Delta h_j\left(t_j^-, \overline{x}\left(t_j^-\right), g\left(\overline{y}\left(t_j^-\right)\right), \Delta g_j\left(\overline{\Delta y_j}\right); p\right), \tag{14}$$

where any subsystem that does not participate in the event continues without any discontinuous update or discrete output signal. A top-level parent system modeling a Scenario contains both the physical and GNC subsystems with their connections as well as the specification for when events occur. For an LCA implementation, an ODE system is constructed with the mean initial condition (10), covariance initial condition analogous to (3) and propagated with mean dynamics (11) and covariance dynamics (4). For each event, the mean update (13) and covariance update (8) are performed.

Navigation Systems Navigation system filters can be considered subsystems with assumed governing equations based on the prediction model according to the filter type. For example, a continuous Kalman observer might have dynamics of the form

$$\dot{\hat{x}}\left(t\right) = \hat{f}\left(t, \hat{x}, \hat{u}\right) + K\left(y - \hat{h}\left(t, \hat{x}, \hat{u}\right)\right).$$

For spaceflight applications, extended Kalman filters have measurement updates of the form

$$\hat{x}\left(t^+\right) = \hat{x}\left(t^-\right) + K\left(\Delta y - \Delta\hat{h}\left(t, \hat{x}, \hat{u},\right)\right).$$

Any number of these filters models can be incorporated seamlessly to the systems modeling approach. To match previous covariance analysis implementations, filter systems can be constructed from their prediction models as special blocks with the following unique rules:

1) An estimated error covariance matrix, $\hat{P}_i$, is propagated for the computation of navigation algorithms like the Kalman update, but is propagated independently from the augmented state and augmented state covariance matrix.
2) The prediction model additive noise terms (e.g., process noise, maneuver execution error, measurement noise, etc.) are not used in the system's governing equations directly, and therefore do not drive the corresponding components in the augmented covariance matrix, but instead are used in the computation of navigation algorithms including the estimated error covariance matrix propagation and updates due to measurements or impulsive actuation.
3) For LCA, the mean state of the filter is not propagated, but is instead replaced by a mapping from mean state of the other (physical) systems,

$$\overline{\hat{x}_i} = m_i\left(\overline{x}\right).$$

GNC Parameters To represent static GNC parameters from the GAUSCOV formulation, a special category of state is introduced for Scenario models that the analyst can use as desired. These static GNC parameter states are collected outside of the physical and GNC subsystems with zero time derivative. This category of state can be used for both the true or estimated GNC parameters. For example, a targeting algorithm execution is modeled as an event output, whose value can be stored in a GNC parameter state for the estimated impulse by updating the assigned GNC parameter state to the event output value; the covariance update equation (8) automatically propagates the uncertainty without special consideration. Similarly, a true static GNC parameter can modeled by updating the value to the true impulsive change in velocity from a physical actuator event model.

Monte Carlo Analysis The systems approach to modeling GNC systems supports seamless implementation of Monte Carlo analysis (MCA). The current MCA implementation uses the mean state initialization (10), mean state equation (11) and mean state updates (13) with additive zero-mean white noise. We also note that the estimated state is initialized, propagated, and updated like any physical state for each MCA simulation, unlike in LCA. For initialization and state updates, the additive noise is sampled from the zero-mean gaussian with covariance defined by the initial dispersions and event update augmented noise (9), respectively. For process noise, we apply a zero-order hold sample of the noise (i.e., a piece-wise constant) so that the value added to the time derivative is given by

$$w\left(t\right) = \frac{Z_{t_i}}{\sqrt{t_s}} \qquad \forall t \in \left[t_i, t_{i+1}\right)$$

where Z_i is drawn from a zero-mean gaussian with covariance given by the augmented process noise (6) and t_s is the duration of the zero-order hold, constant for a particular run.

IV. Example Scenario

A. Physical Systems

1. Vehicle center-of-mass

A generic vehicle center of mass system is defined by the position and velocity expressed in an inertial frame,

$$x^T_{\text{vehicle}} = \left[p^T_{\text{vehicle}}, v^T_{\text{vehicle}}\right]$$

with inputs for gravitational acceleration and non-gravitational acceleration expressed in the same inertial frame,

$$u^T_{\text{vehicle}} = \left[g^T_{\text{vehicle}}, a^T_{\text{vehicle}}\right]$$

and state equations

$$\dot{x}^T_{\text{vehicle}} = \left[v^T_{\text{vehicle}}, g^T_{\text{vehicle}} + a^T_{\text{vehicle}} + w^T_{\text{vehicle}}\right],$$

where w_{vehicle} is a spherical process noise. An event with update

$$x^T_{\text{vehicle}}\left(t_e^+\right) = \left[p^T_{\text{vehicle}}\left(t_e^-\right), v^T_{\text{vehicle}}\left(t_e^-\right) + \Delta v^T_e\right]$$

is used to model impulsive maneuvers. In this scenario, two vehicle center-of-mass systems are deployed, denoted by replacing the "vehicle" subscript with "target" and "chaser," respectively.

2. Gravitation

Environmental systems like gravitation are modeled as stateless systems with a closed loop with the vehicle it acts upon. For each vehicle, a gravitation system is defined with an input position for where the gravitational acceleration is being evaluated,

$$u_{\text{grav}} = \left[p_{\text{grav}}\right],$$

and gravitational acceleration output

$$y_{\text{grav}} = \left[g\left(p_{\text{grav}}\right)\right],$$

which is connected to the gravitational acceleration input of the vehicle. For this scenario, we use a point-mass model of the Earth's gravitational acceleration

$$g\left(p_{\text{grav}}\right) = -\mu_{\text{Earth}} \frac{p_{\text{grav}}}{\left|p_{\text{grav}}\right|^3}$$

but SIGMA supports the generation of systems for computing gravitational effects from multi-body and/or spherical harmonics models with minimal user development effort.

3. Actuators

For the chaser vehicle subsystem, we include a thruster model with 3-dimensional error states for bias, scale-factor, and misalignment

$$x_{\text{thruster}}^T = \left[b^T, s^T, \psi^T\right]$$

modeled as Markov state

$$\dot{x}_{\text{thruster}}^T = \left[-b^T/\tau_b + w_b^T, -s^T/\tau_s + w_s^T, -\psi^T/\tau_\psi + w_\psi^T\right].$$

These error states modify the commanded acceleration input

$$u_{\text{thruster}} = [\hat{a}]$$

by

$$y_{\text{thruster}} = (I + [s\backslash]) (I - [\psi \times]) \hat{a} + b + \eta$$

where η represents zero-mean Gaussian execution error. The same error model is applied for impulsive thrust for a given command

$$\Delta u_{\text{thruster}} = [\Delta\hat{v}]$$

with discrete output

$$\Delta y_{\text{thruster}} = (I + [s\backslash]) (I - [\psi \times]) \Delta\hat{v} + b + \Delta\eta$$

4. Sensors

We also model the chaser vehicle as equipped with an S-band sensor measuring the range and range-rate between the two vehicles. This introduces an additional input of the target vehicle's position and velocity to the sensor,

$$u_{\text{Sband}}^T = \left[p_{\text{target}}^T, v_{\text{target}}^T, p_{\text{chaser}}^T, v_{\text{chaser}}^T\right]$$

with discrete measurement outputs

$$\Delta y_{\text{Sband}}^T\left(t_m\right) = \left[\left|p_{\text{target}} - p_{\text{chaser}}\right| + \nu_{\text{range}}, \frac{\left(p_{\text{target}} - p_{\text{chaser}}\right)^T \left(v_{\text{target}} - v_{\text{chaser}}\right)}{\left|p_{\text{target}} - p_{\text{chaser}}\right|} + \nu_{\text{range-rate}}\right]$$

where ν_{range} and $\nu_{\text{range-rate}}$ are zero-mean gaussian measurement noise. We denoted the inputs here as the position and velocity of the target and chaser, which is how the inputs are connected in this scenario. As a generic sensor model the inputs are the position and velocity of the vehicle being ranged and the vehicle that the sensor is placed on.

B. Relative Dynamics GNC System

1. Clohessy-Wiltshire Equations with Thrusters

The relative motion of two orbiting vehicles about central body in nearly circular orbit can be described with the Clohessy-Wiltshire (CW) equations [23]. The CW equations approximate the dynamics of relative position and velocity in the target vehicle's local-vertical local-horizontal (LVLH) frame, which rotates so the z-axis points towards the central body (radial "R-bar" direction), the y-axis points in the opposite direction of the angular momentum vector h ("H-bar"),

and the x-axis completes the right-handed coordinate system, which is along the velocity vector (down-range or "V-bar") for a circular orbit. The relative dynamics are given by

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 2\omega v_z \\ -\omega^2 p_y \\ 3\omega^2 p_z - 2\omega v_x \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ T_x \\ T_y \\ T_z \end{bmatrix}$$

where ω is the orbital angular velocity of the target vehicle, given in terms of the position and velocity of the target vehicle expressed in the same frame by

$$\omega = \frac{p_{\text{target}} \times v_{\text{target}}}{\left|p_{\text{target}}\right|^2}$$

and $u = \left[T_x, T_y, T_z\right]^T$ are accelerations due to thrust in the LVLH frame. With $x = \left[p_x, p_y, p_z, v_x, v_y, v_z\right]^T$, the CW equations can be written in standard matrix form

$$\dot{x} = A_{\text{CW}} x + B_{\text{CW}} u$$

where

$$A_{\text{CW}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2\omega \\ 0 & -\omega^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3\omega^2 & -2\omega & 0 & 0 \end{bmatrix} \quad \text{and} \quad B_{\text{CW}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In addition to the relative translational CW dynamics, the prediction model uses the same thruster and sensor models as the physical system model. The filter's relative state is used for the chaser position and velocity input to the S-band model and zeros are used for the target. The thruster model uses deterministic zeros for the error state. We also note that since the thruster model is attached to the relative dynamics in the LVLH frame, the thrust commands from the CW guidance are used directly in the prediction model but rotated into the inertial frame for commands to the physical system model.

2. Impulsive Targeting with the state transition matrix

Since the CW equations form a linear time invariant system, we can easily compute the state transition matrix from the matrix exponential of A_{CW} and decompose it into quadrants,

$$\Phi_{\text{CW}}\left(t_1, t_2\right) = e^{A_{\text{CW}}\left(t_2 - t_1\right)} = \begin{bmatrix} \Phi_{pp} & \Phi_{pv} \\ \Phi_{vp} & \Phi_{vv} \end{bmatrix}.$$

We can then solve for the change in velocity required to move from the position p_1 and velocity v_1 at time t_1 to a desired position p_d at time t_2,

$$\Delta v = \Phi_{pv}^{-1}\left(p_d - \Phi_{pp} p_1\right) - v_1.$$

This commanded impulsive acceleration is used directly for the filter model prediction update and rotated to the inertial frame for the physical model. For both the physical chaser model and relative chaser filter model, the impulsive thrust is connected to the actuator to model the maneuver execution error.

We can model targeting at a time t_0 before the maneuver execution by propagating from the position p_0 and velocity v_0

$$\begin{bmatrix} p_1 \\ v_1 \end{bmatrix} = \Phi_{\text{CW}}\left(t_0, t_1\right) \begin{bmatrix} p_0 \\ v_0 \end{bmatrix}.$$

3. Continuous thrust with feedback control

For the continuous thrusting we will use a feedback controller of the form

$$u(t) = -K_p\left(\hat{p}(t) - p_r(t)\right) - K_v\left(\hat{v}(t) - v_r(t)\right)$$

where K_p and K_v are constant feedback gain matrices and $p_r(t)$ and $v_r(t)$ are desired reference trajectories. To choose the feedback gains, we design a linear quadratic regulator (LQR) for the CW equations. LQR is an optimal feedback controller that solves the quadratic optimal control problem

$$\begin{aligned}
\underset{K}{\text{minimize}} \quad & J = \int_{t_0}^{t_f} \left[x(t) - x_r(t)\right]^T Q \left[x(t) - x_r(t)\right] + \left[u(t) - u_r(t)\right]^T R \left[u(t) - u_r(t)\right] dt \\
\text{subject to} \quad & \dot{x}(t) = A(t)\,x(t) + B(t)\,u(t) \\
& u(t) = -K(t)\left(x(t) - x_r(t)\right)
\end{aligned}$$

where Q and R are constant weighting matrices for tuning system performance. This system can be solved by forming a backwards differential Riccati equation, a matrix ODE given by

$$\begin{aligned}
-\dot{S} &= A^T(t)\,S(t) + S(t)\,A - S(t)\,B(t)\,R^{-1}B^T(t)\,S(t) + Q \\
S\left(t_f\right) &= 0
\end{aligned}$$

and forming the time-varying feedback gains as

$$K(t) = R^{-1}B^T S(t). \tag{15}$$

For constant state and input Jacobians $A(t) = A$ and $B(t) = B$, the differential Riccati equation quickly approaches a steady state S_0 which satisfies the algebraic Riccati equation

$$0 = A^T S_0 + S_0 A - S_0 B R^{-1} B S_0 + Q.$$

This means that as $t_f \to \infty$, the optimal feedback gain (15) is constant with $S(t) = S_0$.

C. Scenario Operations

This scenario is primarily a demonstration of the flexibility of this modeling approach and test-case for initial evaluation of the computational correctness of the SIGMA tool. In this Scenario, a non-maneuvering target vehicle begins in circular low Earth orbit with a maneuvering chaser vehicle in a coelliptic orbit 2 km behind and 1 km below the target vehicle. Every 5 minutes, the chaser will perform a measurement using the onboard S-band sensor. The chaser vehicle will perform a transfer maneuver to 500 meters in front of the target modeled as an impulsive thrust 9 minutes after scenario initiation, with targeting performed 3 minutes before maneuver execution. Upon completion of the transfer maneuver, the vehicle will perform continuous thrust using the LQR controller to approach docking.

D. Results

In this subsection, we show simulation results from a 200-sample MCA with 60 second noise sampling and LCA. Figure 2 shows the position of the chaser vehicle relative to the target in the LVLH frame from the MCA simulations along with the mean and 3σ ellipse from LCA. Due to process noise and initial condition dispersions in velocity, the position dispersions increase as the chaser drifts on its coelliptic orbit. After maneuver execution, trajectory dispersions are reduced as the vehicle approaches the targeted position. Upon achieving the docking axis, the continuous feedback control to approach the targeting vehicle continues to reduce the position dispersions. As expected, the vast majority of the MCA samples fit within the LCA bounds. An equivalent plot for the estimated state is shown Figure 3, where we can observe behavior similar to the physical case.

Figure 4a shows the 3σ RSS of the dispersion and error covariances throughout the trajectory for both LCA and MCA. In Figure 4b, the dispersions and error covariances during the finite burn are highlighted, showing the true relative dispersions decrease as the measurement improves the navigation state accuracy. Overall the MCA and LCA behavior agree well despite the difference in modeling for the estimated state. Figures 5 and 6 show the per-axis dispersions from the LCA mean for the physical state and estimated state, respectively. Figure 7 shows the per-axis dispersions of the estimated error.

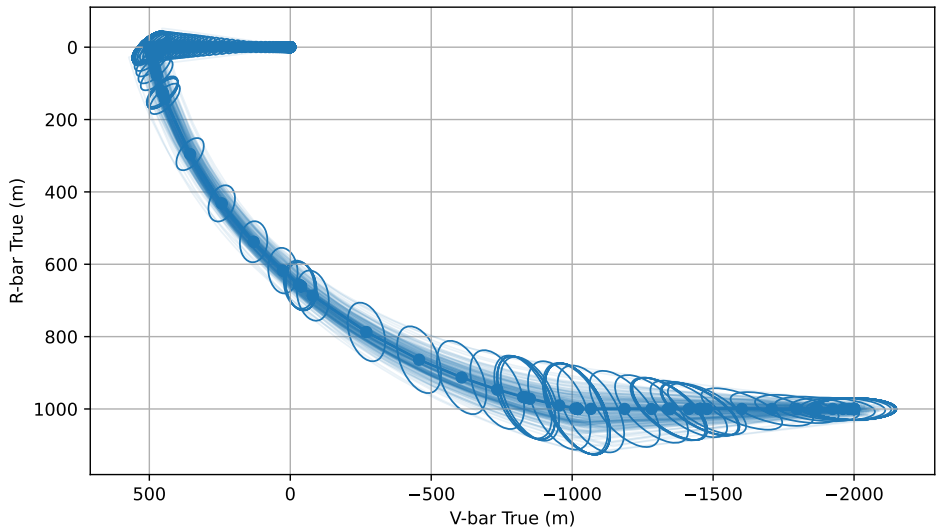


Fig. 2 Chaser position relative to the target in the local vertical/local horizontal reference frame with linear covariance analysis mean and 3σ confidence ellipse and Monte Carlo analysis samples

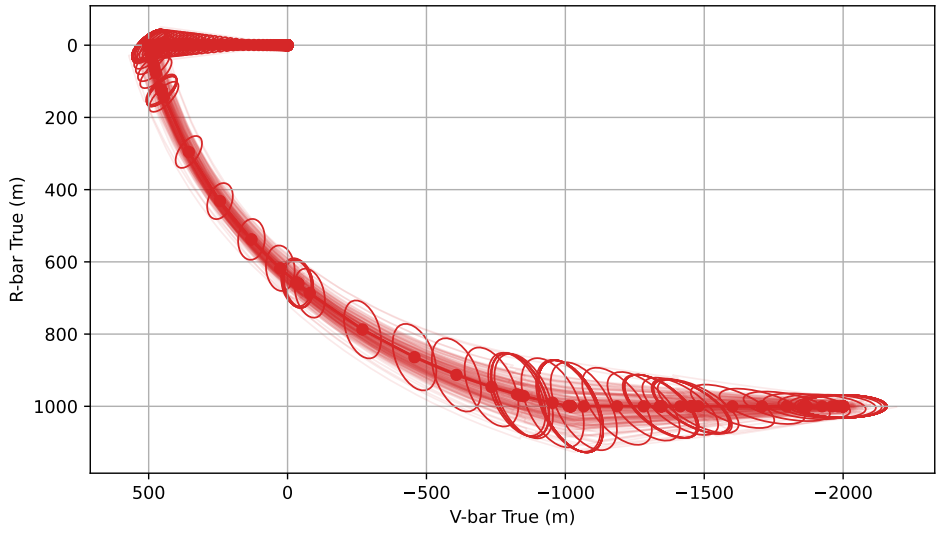
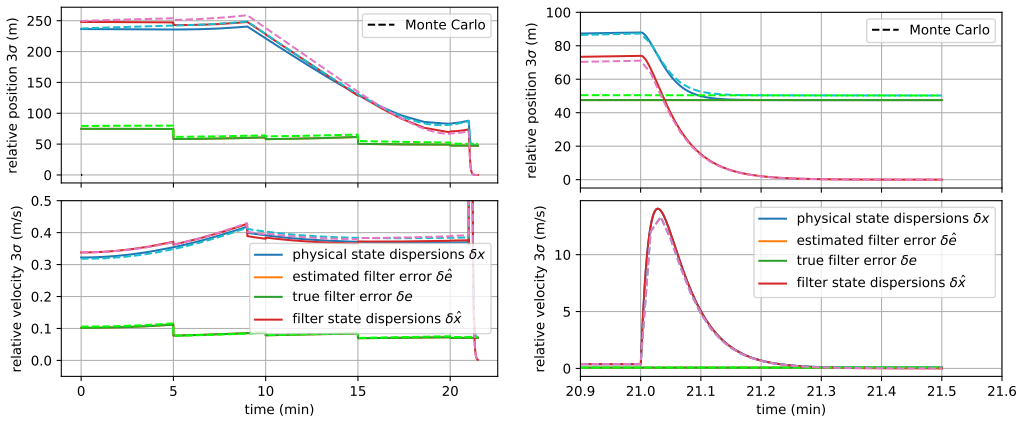
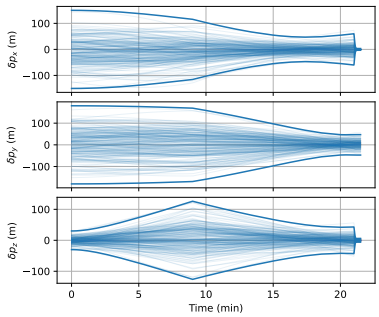


Fig. 3 Estimated chaser position relative to the estimated target in the local vertical/local horizontal reference frame with linear covariance analysis mean and 3σ confidence ellipse and Monte Carlo analysis samples

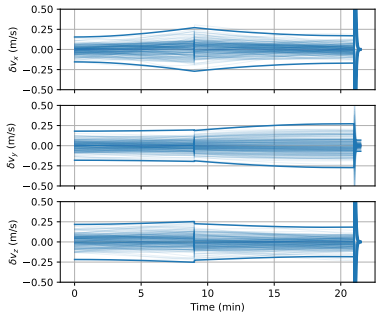


(a) overall trajectory (b) during continuous thrust

Fig. 4 3σ RSS relative dispersions during scenario

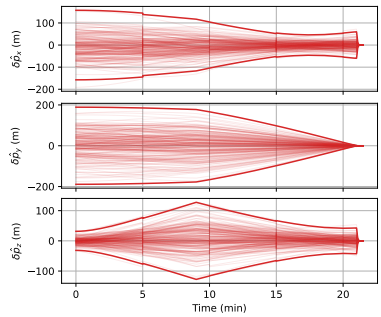


(a) position

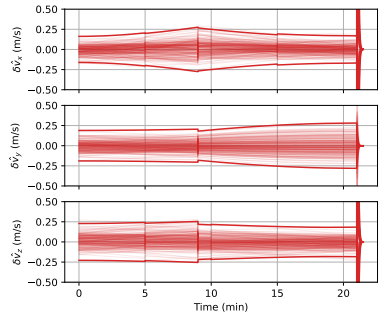


(b) velocity

Fig. 5 Physical state dispersions

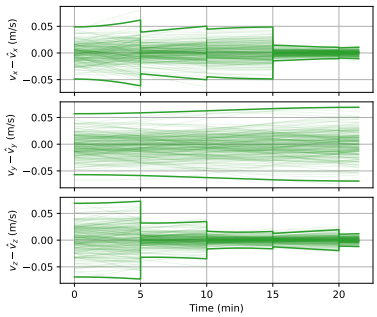


(a) position

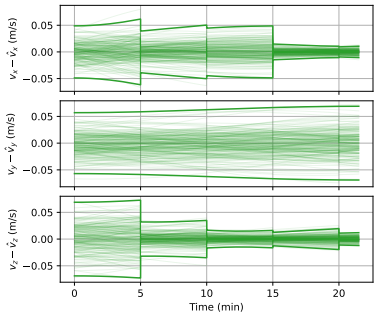


(b) velocity

Fig. 6 Estimated state dispersions



(a) position



(b) velocity

Fig. 7 Estimated state error

V. Conclusion

In this work, we derived the core block diagram algebra rules for uncertain systems that can be used to generalize linear covariance analysis and systematically implement the covariance equations. This architecture has been implemented in SIGMA, the Swift Integrated GNC and Mission Analysis tool. We have demonstrated its flexibility by analyzing a rendezvous scenario with dual-inertial physical model and relative GNC systems, and analyzed the scenario with both linear covariance analysis and Monte Carlo analysis. In addition to the flexible uncertain ODE system modeling architecture, the development team has implemented many relevant models for orbital trajectory analysis, including sensors, actuators, gravitation, and targeting, as well as features to enhance the user experience like the visualization tools that generated the figures in this paper. SIGMA shows promise for analysis and design of future NASA missions.

Acknowledgements

This work was supported by Systems Analysis leadership at NASA Ames Research Center, EG6 at NASA Johnson Space Center, NASA Engineering and Safety Center, and the HLS.

The work presented here was funded, in part, under NASA's ESSCA Contract Number: 80MSFC18C0011. This work was coauthored by employees of The Charles Stark Draper Laboratory, Inc. under Contract No. 80JSC021DA005 with the National Aeronautics and Space Administration.

The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, or allow others to do so, for United States Government purposes. All other rights are reserved by the copyright owner.

References

[1] Maybeck, P. S., *Stochastic models, estimation, and control*, Vol. 1, Academic Press, New York, 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, 2006, pp. 1404–1414.

[3] Jin, K., Geller, D. K., and Luo, J., "Robust Trajectory Design for Rendezvous and Proximity Operations with Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 4, 2020, pp. 741–753.

[4] Geller, D. K., Shuster, S., Woffinden, D., and Bieniawski, S., "Robust Cislunar Trajectory Optimization via Midcourse Correction and Optical Navigation Scheduling," *44th Annual AAS Guidance, Navigaiton and Control Conference*, AAS 22-065, Breckenridge, CO, 2022.

[5] Woffinden, D., Shuster, S., and Geller, S., David Kand Bieniawski, "Robust Trajectory Optimization and GN&C Performance Analysis For NRHO Rendezvous," *2022 AAS/AIAA Astrodynamics Specialist Conference*, 22-564, Charlotte, North Carolina, 2022.

[6] Geller, D., Woffinden, D., and Bieniawski, S., "Sensitivity of Optimal Midcourse Correction Scheduling for Robust Cislunar Trajectory Design," AAS 23-061, Breckenridge, CO, 2023.

[7] Goulet, T., Woffinden, D., Collins, N., and Andrews, B., "Robust Trajectory Design for Rendezvous in a Near Rectilinear Halo Orbit," AAS 23-066, Breckenridge, CO, 2023.

[8] Cavesmith, T., Woffinden, D., and Collins, N., "Angles-Only Robust Trajectory Optimization for NRHO Rendezvous," AAS 24-168, Breckenridge, CO, 2024.

[9] Calkins, G., Woffinden, D., and Putnam, Z., "Robust Trajectory Optimization for Guided Powered Descent and Landing," *2022 AAS/AIAA Astrodynamics Specialist Conference*, AAS 22-660, Charlotte, NC, 2022.

[10] Joshi, J., Woffinden, D., and Putnam, Z., "End-to-End Mars Aerocapture Analysis Using Linear Covariance Techniques and Robust Trajectory Optimization," *2022 AAS/AIAA Astrodynamics Specialist Conference*, AAS 22-678, Charlotte, NC, 2022.

[11] Woffinden, D., Eckman, R., and Robinson, S., "Optimized Trajectory Correction Burn Placement for the NASA Artemis II Mission," AAS 23-062, Breckenridge, CO, 2023.

[12] Woffinden, D., and Barton, B., "Optimized Trajectory Correction Burn Placement for NRHO Orbit Maintenance," *33rd AAS/AIAA Space Flight Mechanics Meeting*, AAS 23-364, Austin, TX, 2023.

[13] Margolis, B. W. L., "A Sweeping Gradient Method for Ordinary Differential Equations with Events," *Journal of Optimization Theory and Applications*, Vol. 199, No. 2, 2023, pp. 600–638. doi:10.1007/s10957-023-02303-3.

[14] Margolis, B. W., and Woffinden, D., "Robust trajectory optimization techniques using a sweeping gradient method and linear covariance analysis," *AAS/AIAA Astrodynamics Specialist Conference*, 2024.

[15] Margolis, B. W. L., and Woffinden, D., "Co-Optimization of Navigation System Requirements and Trajectory Design Using a Sweeping Gradient Method and Linear Covariance Analysis," *AIAA and AAS Astrodynamics Specialist Conference*, Broomfield, CO, 2024.

[16] York, C., Geller, D., and Woffinden, D., "Generalized Augmented-State Covariance Analysis for Spaceflight," *47th Annual American Astronautical Society (AAS) Guidance, Navigation and Control (GN&C) Conference*, 2025.

[17] Geller, D., Woffinden, D., and York, C., "Generalized Reference Targeting for Spaceflight," *47th Annual American Astronautical Society (AAS) Guidance, Navigation and Control (GN&C) Conference*, American Astronomical Society, 2025.

[18] Margolis, B. W. L., and Lyons, K. R., "Condor, a mathematical modeling framework for engineers with deadlines," *Journal of Open Source Software*, Vol. 10, No. 113, 2025, p. 8859. doi:10.21105/joss.08859.

[19] Woffinden, D., and Breger, L., "Automated Derivation and Verification of Navigation Requirements for On-orbit Rendezvous," AIAA 2013-4964, Boston, MA, 2013.

[20] Geller, D. K., Rose, M. B., and Woffinden, D. C., "Event Triggers in Linear Covariance Analysis with Applications to Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 102–111. doi:10.2514/1.36834.

[21] Zanetti, R., Woffinden, D. C., and Sievers, A., "Multiple Event Triggers in Linear Covariance Analysis for Spacecraft Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 353–366. doi:10.2514/1.54965.

[22] W. L. Margolis, B., "SimuPy: A Python framework for modeling and simulating dynamical systems," *The Journal of Open Source Software*, Vol. 2, No. 17, 2017, p. 396. doi:10.21105/joss.00396.

[23] Clohessy, W. H., and Wiltshire, R., "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aero/Space Sciencess*, Vol. 27, 1960, pp. 653—658.