Modeling Stochastic Process Uncertainties in Spacecraft Dynamics: A New Capability in the Basilisk Framework

Juan Garcia-Bonilla
University of Colorado Boulder
429 UCB, 3775 Discovery Drive
Boulder, Colorado 80303 USA
juan@garciabonilla.com

Hanspeter Schaub
University of Colorado Boulder
429 UCB, 3775 Discovery Drive
Boulder, Colorado 80303 USA

***Abstract*—**Spacecraft are subject to dynamic uncertainties that cannot be adequately represented by static random parameters, including fluctuations in atmospheric density, solar flux variability, and thrust noise. These effects are more naturally modeled as stochastic processes governed by stochastic differential equations (SDEs), but mainstream aerospace and robotics simulation frameworks typically realize uncertainty through discrete-time noise injection while relying on deterministic Ordinary Differential Equation solvers. This paper introduces a modular, model-based framework for SDE-driven uncertainty modeling that allows users to assign continuous-time stochastic dynamics to arbitrary simulation parameters with minimal changes to existing simulation graphs. The reference implementation is provided in the open-source Basilisk astrodynamics toolkit, which is extended to support stochastic states driven by additive and multiplicative noise, explicit management of drift and diffusion contributions, and numerical integrators for Itô-type SDEs. Representative processes, including Ornstein-Uhlenbeck and higher-order Gauss-Markov models, are reviewed along with practical parameter interpretations and estimation considerations relevant to flight data. Design options for integrating SDE propagation into model-based toolkits are analyzed in terms of computational efficiency, usability, and flexibility, and the chosen implementation strategy in Basilisk is described. A de-orbit case study with stochastic atmospheric density demonstrates the operational impact of continuous-time process modeling: the assumed correlation time fundamentally changes the predicted dispersion in orbital lifetime, while treating density uncertainty as a constant random bias can substantially mischaracterize re-entry uncertainty. By embedding SDE-based stochastic process propagation into a widely used astrodynamics framework, this work increases simulation fidelity and provides a reusable capability for mission design, operations analysis, and autonomy validation under time-varying uncertainties.

TABLE OF CONTENTS

979-8-3315-7360-7/26/$31.00 ©2026 IEEE

1. INTRODUCTION

Accurate modeling of spacecraft dynamics is essential for mission planning, guidance and navigation algorithm design, autonomy validation, and space traffic management. Errors in dynamical models directly impact predictions of orbital evolution, proximity operations, and end-of-life disposal, all of which carry operational and safety implications. A critical challenge in this context is the treatment of uncertainty. While many uncertainties, such as initial state errors, sensor biases, or constant model parameters, can be represented as static random variables or deterministically varying parameters, others evolve dynamically and cannot be adequately captured by static models.

Several sources of uncertainty in astrodynamics exhibit intrinsically stochastic, time-varying behavior. Atmospheric density fluctuations in low Earth orbit [1, 2], variations in solar flux [3, 4], and thrust noise from propulsion systems [5, 6] are notable examples. Unlike static uncertainties, these effects evolve randomly over time, influencing spacecraft motion in ways that cannot be captured by fixed parameter offsets. Modeling such phenomena as stochastic processes, described through Stochastic Differential Equations (SDEs), provides a more realistic description of their behavior and enables more accurate prediction of their long-term impact on spacecraft dynamics.

Despite the prevalence of such stochastic effects, current simulation tools do not natively support the continuous-time integration of SDEs. Widely used platforms in robotics and aerospace offer only partial mechanisms for uncertainty modeling, for example perturbing sensor or actuator signals with random variables, injecting disturbances through plugins, or sampling random values at fixed intervals and holding them constant between integrator steps. Frameworks such as Gazebo [7], MuJoCo [8], Isaac Sim [9], JPL DARTS [10], Chrono [11], and Modelica [12] [13] allow noise injection, and Drake [14] provides structured random input ports, but all rely on deterministic Ordinary Differential Equation (ODE) solvers underneath. Similarly, spaceflight simulators such as GMAT [15], STK [16], FreeFlyer [17], NASA 42 [18], and Basilisk [19] support configurable noise models but lack stochastic integrators. As a result, noise is sampled at sensor or controller update rates and then treated as a constant value within each integration stage rather than being modeled as a diffusion term that evolves continuously in time.

In all of these tools, stochastic effects are therefore realized only through sampled perturbations or stateful processes evaluated at discrete update intervals. The underlying state propagation relies on deterministic ODE or Differential Algebraic Equation solvers that treat such inputs as constant within each integration step. None of the platforms listed include support for continuous-time diffusion models, numerical SDE

solvers, or stochastic state propagation of the form

$$\mathrm{d}\boldsymbol{X} = \boldsymbol{f}(t, \boldsymbol{X})\mathrm{d}t + \sum_{i=1}^{n} \boldsymbol{g_i}(t, \boldsymbol{X})\mathrm{d}W_i \tag{1}$$

This paper addresses this gap by presenting a modular approach for incorporating stochastic uncertainties into model-based astrodynamics simulation frameworks through SDEs. The method allows users to assign stochastic dynamics to arbitrary parameters in a simulation, supports both additive and multiplicative noise structures, and interfaces clearly with SDE numerical integrators. The approach is designed to integrate with model-based architectures [20], enabling deterministic states to be replaced with stochastic counterparts without altering the surrounding framework. While the reference implementation is provided in the open-source Basilisk toolkit [19], the principles are general and can be applied in other model-based simulation environments.

To ground the approach, the paper provides a brief review of stochastic process modeling, including the Ornstein-Uhlenbeck and higher-order Gauss-Markov processes, which are frequently used in aerospace applications [21, 22]. Practical aspects such as parameter selection, interpretation, and estimation from data are also discussed [23]. The framework is then demonstrated through a case study of a spacecraft in low Earth orbit subject to uncertainty in atmospheric density. Results from Monte Carlo analysis are compared between two treatments: a static random-density model and a stochastic Ornstein-Uhlenbeck process. The comparison highlights the differences in uncertainty growth, underscoring the importance of modeling time-dependent stochastic effects for accurate prediction of orbital behavior.

This paper makes three contributions. First, it formulates a general, model-based interface for representing time-varying uncertainties as stochastic states governed by stochastic differential equations, allowing continuous-time uncertainty to be attached to arbitrary simulation parameters using additive or multiplicative noise structures. Second, it analyzes alternative architectural strategies for integrating drift and diffusion evaluations into model-based dynamics toolkits and documents the implementation selected for the Basilisk framework, which balances computational efficiency with user workflow simplicity. Third, it demonstrates the practical impact of continuous-time stochastic uncertainty modeling through a de-orbit case study in which atmospheric density is modeled as an Ornstein-Uhlenbeck process; the results show that the assumed correlation time fundamentally alters predicted orbital lifetime dispersion, and that treating uncertainty as a constant random bias can significantly mischaracterize re-entry uncertainty.

The remainder of this paper is organized as follows. Section 2 introduces the formulation of stochastic differential equations. Section 3 discusses stochastic processes relevant to astrodynamics, with emphasis on the Ornstein-Uhlenbeck and Gauss-Markov families. Section 4 examines sources of stochastic uncertainty in spacecraft dynamics, including atmospheric variability, solar flux, thruster performance, and unmodeled perturbations. Section 5 presents the implementation of SDE-based propagation within the Basilisk toolkit and compares alternative approaches. Section 6 demonstrates the framework through a de-orbit case study with stochastic atmospheric density. Conclusions are provided in Section 7.

2. STOCHASTIC DIFFERENTIAL EQUATIONS

From Deterministic to Stochastic Dynamics

The evolution of a deterministic dynamical system is governed by an ODE:

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(t, \boldsymbol{x})\mathrm{d}t \tag{2}$$

where $\boldsymbol{f}(t, \boldsymbol{x})$ specifies the rate of change of the state vector $\boldsymbol{x}$ at time t.

When stochastic forces drive the dynamics, the system is instead described by a Stochastic Differential Equation (SDE):

$$\mathrm{d}\boldsymbol{X} = \boldsymbol{f}(t, \boldsymbol{X})\mathrm{d}t + \sum_{i=1}^{n} \boldsymbol{g_i}(t, \boldsymbol{X})\mathrm{d}W_i \tag{3}$$

where W_i are independent Wiener processes (Brownian motions). Here, $\boldsymbol{f}$ is the *drift term*, while each $\boldsymbol{g_i}$ is a *diffusion term* that modulates the effect of stochastic perturbations.

The solution to an ODE is a deterministic trajectory $\boldsymbol{x}(t)$, while the solution to an SDE is a stochastic process $\boldsymbol{X}(t)$ that represents an ensemble of possible trajectories. Consequently, analysis of stochastic systems often focuses on statistical moments (e.g., mean, variance) or on their probability density function (pdf) at a given time.

Discretization: Euler and Euler-Maruyama

A useful approximation for ODEs is the Euler method, which for a small timestep Δt gives:

$$\Delta x = f(t, x)\Delta t \tag{4}$$

The stochastic analogue is the Euler-Maruyama scheme:

$$\Delta X = f(t, X)\Delta t + g(t, X)\Delta W \quad \Delta W \sim \mathcal{N}(0, \Delta t) \tag{5}$$

where ΔW is a normally distributed increment with variance proportional to Δt. Thus, over a small timestep, the change in X has mean $f(t, X)\Delta t$ and standard deviation $g(t, X)\sqrt{\Delta t}$. The Euler-Maruyama scheme is the simplest practical tool for simulating SDEs numerically; higher-order integrators are also available [24–26].

Random ODEs and Their Equivalence to SDEs

An alternative formulation found in the literature is the Random Ordinary Differential Equation (RODE), where stochasticity enters through auxiliary processes:

$$\mathrm{d}\boldsymbol{X} = \boldsymbol{f_x}(t, \boldsymbol{X}, \boldsymbol{Y})\mathrm{d}t \tag{6}$$

with $\boldsymbol{Y}$ governed by an SDE:

$$\mathrm{d}\boldsymbol{Y} = \boldsymbol{f_Y}(t, \boldsymbol{Y})\mathrm{d}t + \sum_{i=1}^{n} \boldsymbol{g_{Y,i}}(t, \boldsymbol{Y})\mathrm{d}W_i \tag{7}$$

Although RODEs may appear more general, they can always be recast as SDEs by augmenting the state vector:

$$\mathrm{d}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{Y}\end{bmatrix} = \begin{bmatrix}\boldsymbol{f_x}(t, \boldsymbol{X}, \boldsymbol{Y})\\ \boldsymbol{f_Y}(t, \boldsymbol{Y})\end{bmatrix}\mathrm{d}t + \sum_{i=1}^{n} \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{g_{Y,i}}(t, \boldsymbol{Y})\end{bmatrix}\mathrm{d}W_i \tag{8}$$

For this reason, and because SDEs are more widely studied in the literature, this paper expresses stochastic dynamics in the SDE framework.

Example: Stochastic Spring-Damper System

As an example, consider a spring-damper system subject to a stochastic forcing term:

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + c\frac{\mathrm{d}x}{\mathrm{d}t} + kx = F_x(t) \tag{9}$$

where $F_x(t)$ is a stochastic force described by an SDE. A first-order SDE representation of this system is:

$$\mathrm{d}\begin{bmatrix} x \\ v \\ F_x \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m}(-cv - kx + F_x) \\ f_{F_x}(\cdot) \end{bmatrix}\mathrm{d}t + \sum_{i=1}^{n}\begin{bmatrix} 0 \\ 0 \\ g_{F_x,i}(\cdot) \end{bmatrix}\mathrm{d}W_i \tag{10}$$

This illustrates how physical systems with stochastic perturbations can be consistently modeled within the SDE framework.

3. Relevant Stochastic Processes

Forces as White Noise

A first idea for modeling stochastic perturbations is to include a noise term acting directly on acceleration. Returning to the spring-damper system:

$$\mathrm{d}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m}(-cv - kx) \end{bmatrix}\mathrm{d}t + \begin{bmatrix} 0 \\ \sigma \end{bmatrix}\mathrm{d}W, \tag{11}$$

with constant diffusion σ.

Recovering the effective perturbation force (from Equation 10) gives

$$\frac{1}{m}F_x\mathrm{d}t = \sigma\mathrm{d}W, \tag{12}$$

which implies F_x is proportional to the time derivative of a Wiener process. This corresponds to a white-noise signal with infinite variance, zero correlation time, and infinite bandwidth. Physically, such a force would exhibit instantaneous fluctuations, which is not realistic.

Introducing the Force as a State

To avoid this, the force can be modeled as an additional state. Simplifying Equation 10 for constant diffusion σ:

$$\mathrm{d}\begin{bmatrix} x \\ v \\ F_x \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m}(-cv - kx + F_x) \\ 0 \end{bmatrix}\mathrm{d}t + \begin{bmatrix} 0 \\ 0 \\ \sigma \end{bmatrix}\mathrm{d}W \tag{13}$$

The perturbation is now a force state, but its variance grows unbounded with time since it is a pure Brownian motion. This means that, as time progresses, the perturbation force might reach larger and larger values. In many cases, however, one instead wants a bounded process with a stationary distribution, such that the perturbation magnitude does not continue growing indefinitely.

The Ornstein-Uhlenbeck Process

A simple bounded alternative is the Ornstein-Uhlenbeck (OU) process:

$$\mathrm{d}x = -\theta(x - \mu)\mathrm{d}t + \sigma\mathrm{d}W \tag{14}$$

For $\sigma = 0$, this reduces to an exponential decay towards μ with rate θ. With $\sigma > 0$, the drift term pulls the process towards μ, while the diffusion perturbs it away. Unlike Brownian motion, the OU process has a stationary variance.

Figure 1 highlights the differences between the perturbation force models considered so far. Brownian motion has variance that grows without bound, while the OU process converges to a stationary distribution.

The OU process can be re-parameterized in terms of correlation time τ and stationary standard deviation σ_{st}:

$$\mathrm{d}x = -\frac{1}{\tau}(x - \mu)\mathrm{d}t + \sqrt{\frac{2}{\tau}}\sigma_{\text{st}}\mathrm{d}W \tag{15}$$

Assuming $x(t_0) = 0$, the mean, standard deviation, and autocorrelation of such processes are

$$\mathrm{E}[x(t)] = \mu(1 - e^{-t/\tau}) \tag{16}$$

$$\mathrm{Std}[x(t)] = \sigma_{\text{st}}\sqrt{1 - e^{-2t/\tau}} \tag{17}$$

$$\rho(\Delta t) = e^{-|\Delta t|/\tau} \tag{18}$$

Figure 2 illustrates the effect of the correlation time τ on the rate of variation of a perturbation force. Small τ produces rapid fluctuations, while large τ yields slower changes. The inset plots reveal the self-similarity of these processes: the segment of the $\tau = 0.1$ s trajectory over $t \in [0, 1]$ s resembles the $\tau = 1$ s trajectory over $[0, 10]$ s. Similarly, the inset of the $\tau = 1$ s resembles the $\tau = 10$ s trajectory over the full duration.

Time-Varying OU Processes

In many applications, the mean or variance evolves with time. For example, atmospheric density along an elliptical orbit varies predictably with altitude. One can generalize the OU process with time-varying parameters:

$$\mathrm{d}x = -\frac{1}{\tau(t)}(x - \mu(t))\mathrm{d}t + \sqrt{\frac{2}{\tau(t)}}\sigma_{\text{st}}(t)\mathrm{d}W \tag{19}$$

Alternatively, a normalized OU process $\kappa(t)$ with mean zero and unit variance can be scaled:

$$x(t) = \mu(t) + \sigma_{\text{st}}(t)\kappa(t) \tag{20}$$

$$\mathrm{d}\kappa = -\frac{1}{\tau(t)}\kappa\mathrm{d}t + \sqrt{\frac{2}{\tau(t)}}\mathrm{d}W \tag{21}$$

As Figure 3 illustrates, the first formulation retains inertia, so parameter changes are absorbed gradually. The scaled formulation adapts instantly. The appropriate choice depends on application context.

Higher-Order Gauss-Markov Processes

White noise and OU processes are zeroth- and first-order Gauss-Markov processes. Higher orders allow more flexible autocorrelation structures. In particular, the underdamped second-order Gauss-Markov (SOGM) process introduces oscillatory correlations, useful in astrodynamics where perturbations may align with orbital periods [22].

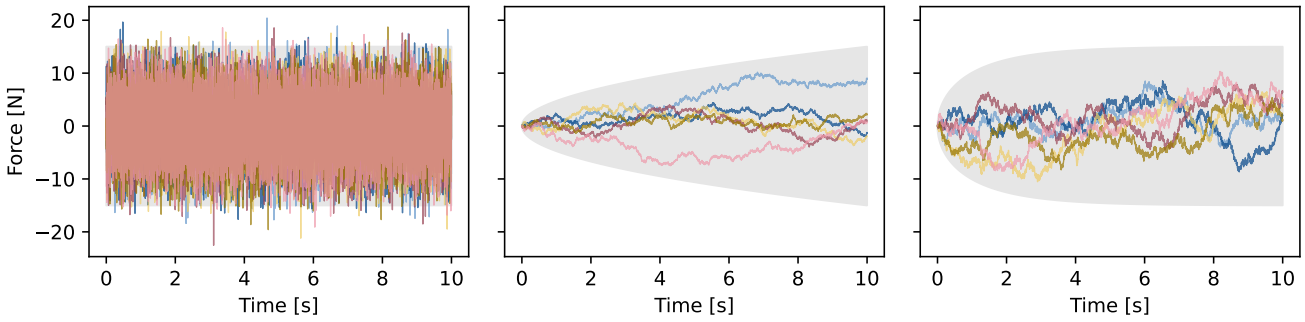


Figure 1: Comparison of perturbation forces modeled as (left to right): white noise on acceleration, Brownian motion, and an OU process. Shaded areas indicate 3σ bounds.

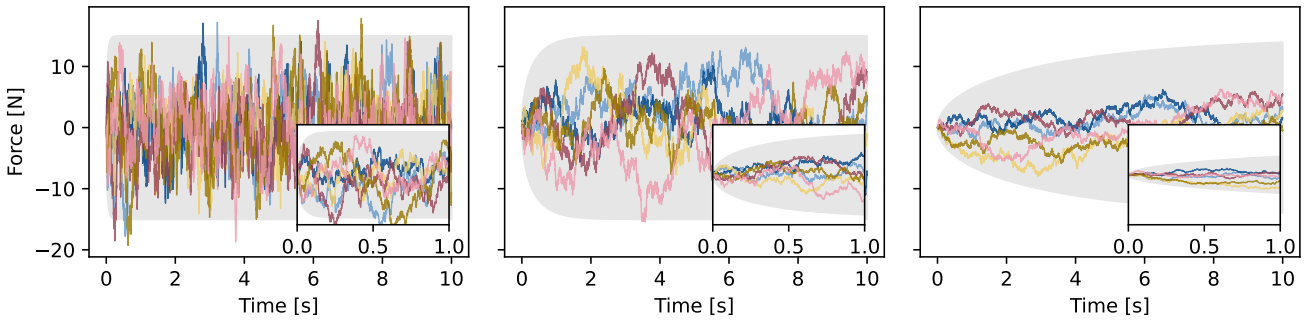


Figure 2: Ornstein-Uhlenbeck processes with identical stationary standard deviation but different correlation times $\tau = 0.1$ s, 1 s, $10s$ (left to right). Insets show the interval $t \in [0, 1]$ s.

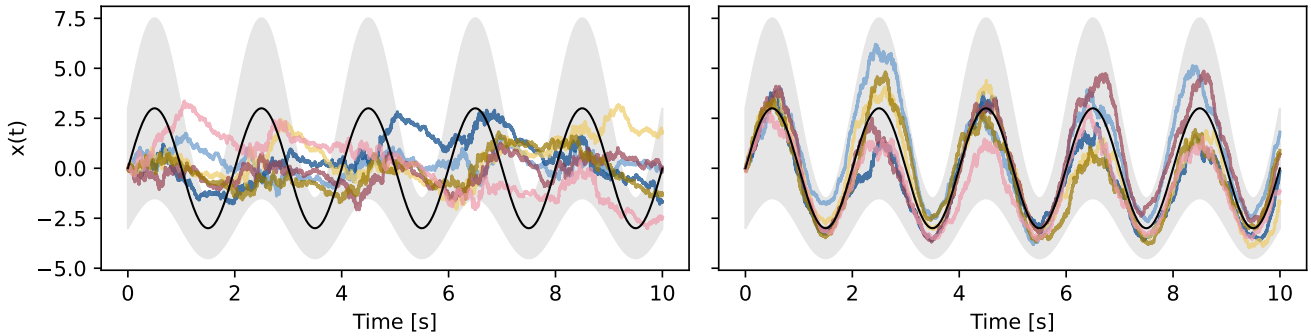


Figure 3: OU processes with time-varying parameters. Left: process given by Equation 19; right: process given by Equations 20-21. The sinusoidal black line indicates $\mu(t)$ and shaded regions show $3\sigma_{\text{st}}(t)$ bounds.

A first-order SDE representation of SOGM is

$$\mathrm{d}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -2\zeta\omega_n x_2 - \omega_n^2(x_1 - \mu) \end{bmatrix} \mathrm{d}t + \begin{bmatrix} 0 \\ \sigma \end{bmatrix} \mathrm{d}W \quad (22)$$

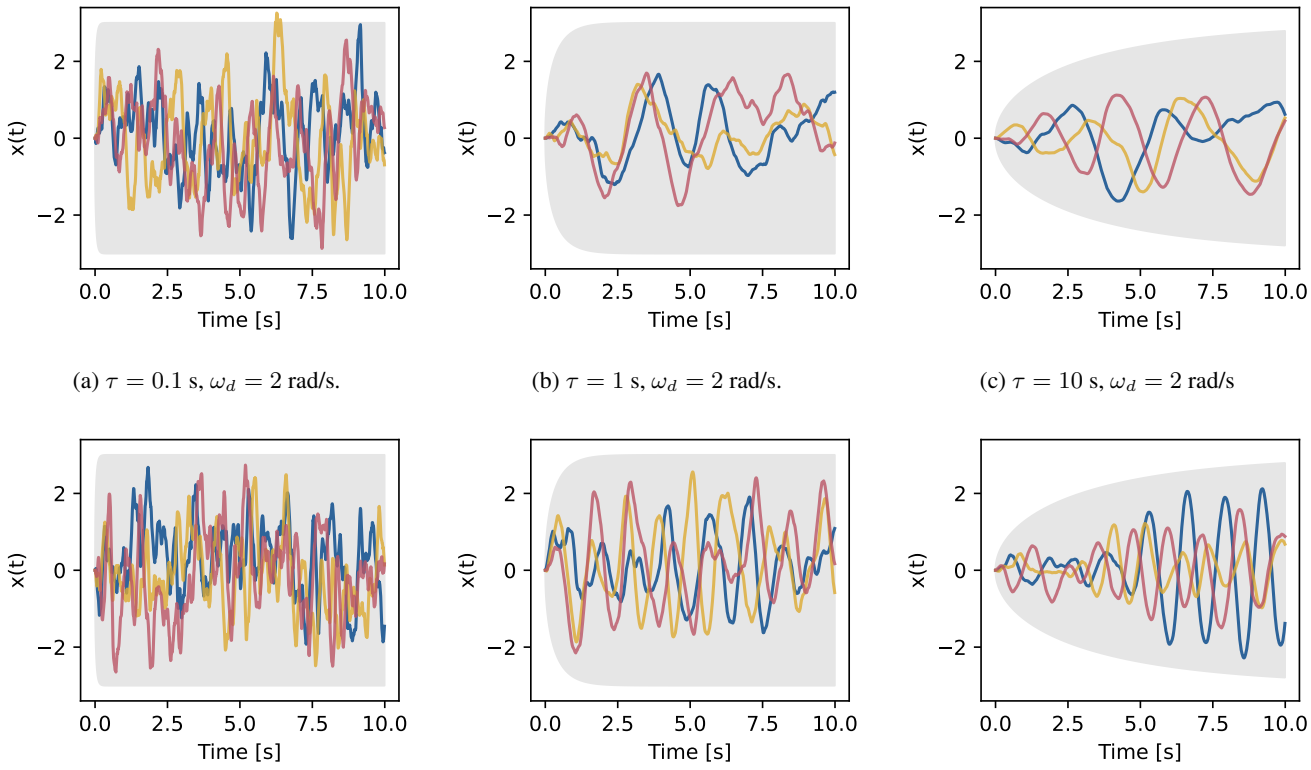
Rewriting with a correlation time $\tau = (\zeta\omega_n)^{-1}$, damped oscillation frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$, and stationary standard deviation $\sigma_{\text{st}} = \sigma/[2(\zeta\omega_n^3)^{1/2}]$ gives

$$\mathrm{d}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{2}{\tau}x_2 - \left(\frac{1}{\tau^2} + \omega_d^2\right)(x_1 - \mu) \end{bmatrix} \mathrm{d}t + \begin{bmatrix} 0 \\ \sigma \end{bmatrix} \mathrm{d}W \quad (23)$$

$$\sigma = 2\sigma_{\text{st}}\sqrt{\frac{1}{\tau}\left(\frac{1}{\tau^2} + \omega_d^2\right)} \quad (24)$$

The parameters μ and σ_{st} play similar roles as in the OU process. Figure 4 shows how τ and ω_d shape the dynamics. The correlation time τ controls how quickly memory of past values decays: smaller τ produces faster variations, while larger τ yields slower evolution. The frequency parameter ω_d sets the oscillation rate. Oscillations are pronounced when τ is large, but for small τ the rapid loss of correlation suppresses visible oscillatory behavior.

OU and SOGM processes have clear limiting cases: $\tau \to 0$



(a) $\tau = 0.1$ s, $\omega_d = 2$ rad/s. (b) $\tau = 1$ s, $\omega_d = 2$ rad/s. (c) $\tau = 10$ s, $\omega_d = 2$ rad/s

(d) $\tau = 0.1$ s, $\omega_d = 5$ rad/s. (e) $\tau = 1$ s, $\omega_d = 5$ rad/s. (f) $\tau = 10$ s, $\omega_d = 5$ rad/s

Figure 4: Second-order Gauss-Markov processes with different τ and ω_d.

yields white noise, $\tau \to \infty$ yields a constant random value, and $\omega_d = 0$ reduces the SOGM to a critically damped OU-like process.

Parameter Estimation

In practice, one often fits a stochastic process to discrete data. This requires choosing a model family and estimating parameters. Maximum likelihood estimation (MLE) is common, posed as [23]

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) \tag{25}$$

with log-likelihood

$$\ell(\boldsymbol{\theta}) = -\sum_{k=0}^{T} \log p\big(\mathbf{x}(t_{k+1}) \mid \mathbf{x}(t_k), \boldsymbol{\theta}\big) \tag{26}$$

where $\mathbf{x}(t_k)$ are observations and $p(\cdot)$ are transition densities.

For the OU process with $\mu = 0$, the transition density is closed form, yielding explicit MLE formulas [23]:

$$\theta_{\text{ML}} = -\frac{1}{\Delta t} \log\left[\frac{\sum_{k=0}^{T-1} x(t_k)\, x(t_{k+1})}{\sum_{k=0}^{T-1} x(t_k)^2}\right] \tag{27}$$

$$\sigma_{\text{ML}} = \frac{1}{T}\left(\frac{2\theta_{\text{ML}}}{1 - e^{-2\theta_{\text{ML}}\Delta t}}\right) \times \sum_{k=0}^{T-1} \left(x(t_{k+1}) - e^{-\theta_{\text{ML}}\Delta t} x(t_k)\right)^2 \tag{28}$$

Parameter estimation methods for linear and generic SDEs are reviewed in References 23, 27, 28.

4. Stochastic Uncertainties in Astrodynamics

This section reviews sources of uncertainty in perturbation forces acting on spacecraft and how they may be represented stochastically.

Atmospheric Behavior

In low planetary orbits, atmospheric drag is often the dominant perturbation. For re-entry trajectories, atmospheric forces strongly govern dynamics. These forces depend on local density and wind, so uncertainties in either translate into uncertainties in drag. Because thermospheric behavior is influenced by complex dynamics and external drivers such as solar and geomagnetic activity, empirical models cannot fully capture its variability [1, 2].

Accelerometer data from satellites has been widely used to estimate local density and winds experienced by spacecraft [2]. These studies reveal chaotic short-term variability that models such as HWM07 and NRLMSISE-00 fail to reproduce. Figure 5, for example, shows density and wind estimates from GRACE-B over a three-hour interval compared with model predictions. The measurements exhibit both higher variability and significant deviations from the model.

Longer-term studies using density estimates derived from Two-Line Element (TLE) data show similar discrepancies.

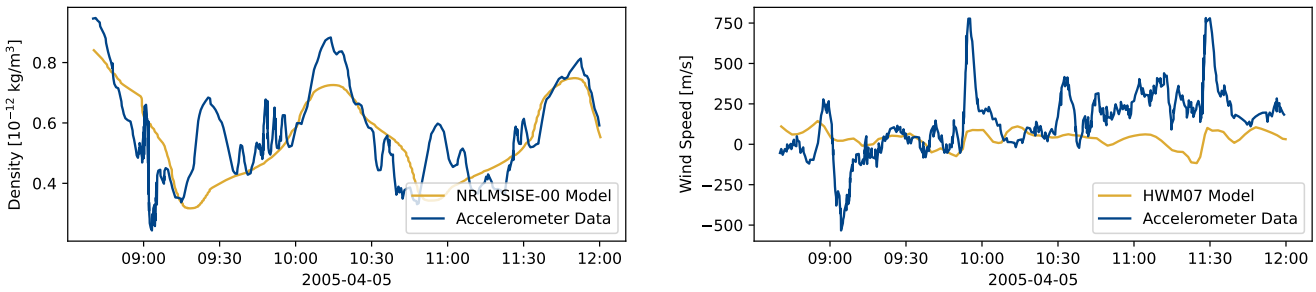


Figure 5: Time series of density and wind estimated from accelerometer data for GRACE-B compared with HWM07/NRLMSISE-00. Data from Reference 2.

Figure 6 presents local density for CHAMP from both the NRLMSISE-00 model and TLE-based estimates, along with the residual time series. The residuals highlight the stochastic nature of model errors.

Such data enables stochastic modeling of atmospheric density and wind by fitting processes to residuals between models and measurements. However, this approach assumes knowledge of model inputs such as solar flux (F10.7) and geomagnetic indices (AP), which are themselves uncertain. Figure 7 shows their historical variability. While long-term trends exist, short-term behavior is highly irregular. For instance, solar flares and geomagnetic storms can cause abrupt 30-100% density increases on timescales of minutes to hours, while diurnal, multi-day, and solar cycle variations introduce changes ranging from tens of percent to an order of magnitude [1].

Given these uncertainties, one may either fit stochastic processes directly to density/wind data, or model F10.7 and AP as stochastic inputs to atmospheric models.

Solar Flux

Solar radiation pressure significantly perturbs satellites and interplanetary spacecraft, particularly those with large reflective surfaces such as antennas or solar sails. Its magnitude depends on the Total Solar Irradiance (TSI), which varies due to solar physics [3]. While solar activity follows an 11-year cycle, short-term variability remains unpredictable [4].

Figure 8 shows TSI data and fitted Ornstein-Uhlenbeck processes. Estimated parameters are $\tau = 4.004$ days and $\sigma_{\text{st}} = 0.097$ W/m^2 for the 2-year window, and $\tau = 0.375$ days and $\sigma_{\text{st}} = 0.035$ W/m^2 for the 90-day window. Outliers corresponding to strong solar events were excluded. Stochastic modeling of TSI using OU processes thus provides a tractable way to approximate short-term variability.

Thruster Performance

Rocket engine performance depends on complex physical processes that affect thrust and propellant consumption. For electric propulsion, thrust noise is an important concern in high-precision applications. Sources of this noise include electromagnetic interference, vibrations, and environmental interactions, which generate small-amplitude, high-frequency fluctuations [5]. Figure 9 shows experimental Hall thruster thrust data with clear stochastic features. Similar behavior has been reported for colloid thrusters in flight [6].

Monopropellant thrusters also exhibit stochastic thrust variations due to transient effects in feed pressure, valve timing, flow resistance, and catalyst behavior. Choi and Bach [31] observed oscillations in a high-test peroxide thruster under pulsed operation, linked to surges in the propellant feed system.

Prescribed Motion

In astrodynamics, it is common to prescribe either translational or rotational motion depending on the focus of the analysis. When translational dynamics are of interest, attitude control is often assumed to be perfect, and the spacecraft is modeled as following a commanded orientation profile while its orbital trajectory is propagated. Conversely, when attitude dynamics are the primary concern, the orbital path may be prescribed and only the rotational equations of motion are integrated. In some studies, both the position and attitude of the spacecraft body are prescribed in order to isolate the behavior of a subsystem, such as a robotic manipulator or an antenna deployment mechanism, without the added complexity of propagating the full rigid-body dynamics.

The classical assumption underlying prescribed motion is that the spacecraft follows the commanded trajectory exactly, with no error in position or orientation. In practice, this assumption is rarely valid. Uncertainties in actuation, sensing, and environmental forcing introduce deviations between the commanded and realized states. For example, solar sails are notoriously difficult to control because of their large, flexible structures and their sensitivity to environmental perturbations. Reference 32 documents this challenge, reporting significant and fluctuating discrepancies between commanded and actual control angles during the LightSail 2 technology demonstration mission.

When the fidelity of prescribed motion affects system-level behavior, a stochastic framework provides a more realistic description. Random processes, such as Ornstein-Uhlenbeck models, can be used to represent variability in commanded profiles, allowing analyses to account for imperfect control performance without the need to propagate full coupled position-attitude dynamics [33].

Unknown Unknowns

Stochastic processes are also useful for modeling perturbations that are unmodeled or only partially understood. In orbit determination, Myers and Tapley [21] introduced Gauss-Markov processes to represent unknown accelerations. Leonard et al. [22] applied second-order Gauss-Markov models to capture the effect of unmodeled higher-order gravity harmonics. For many bodies, particularly small bodies,

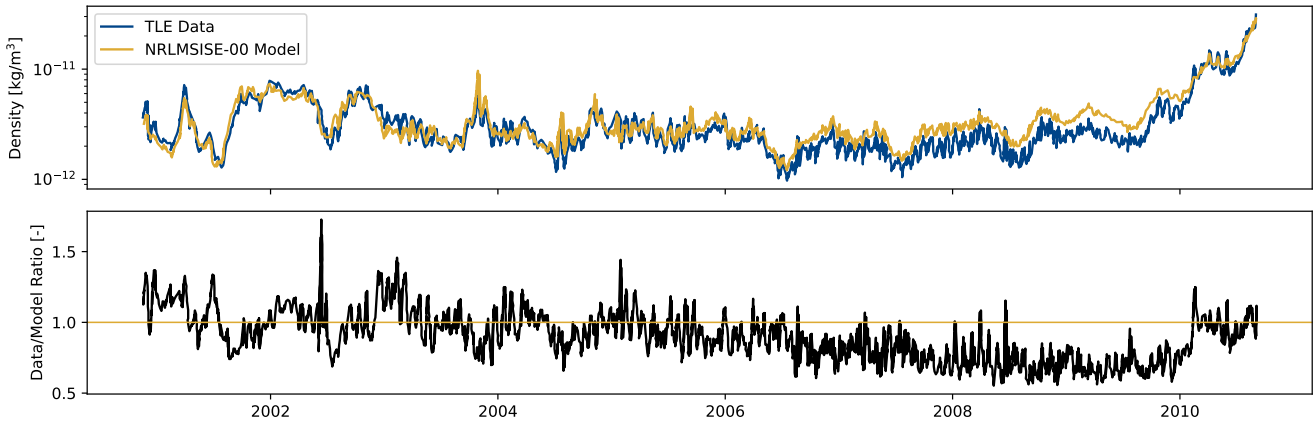
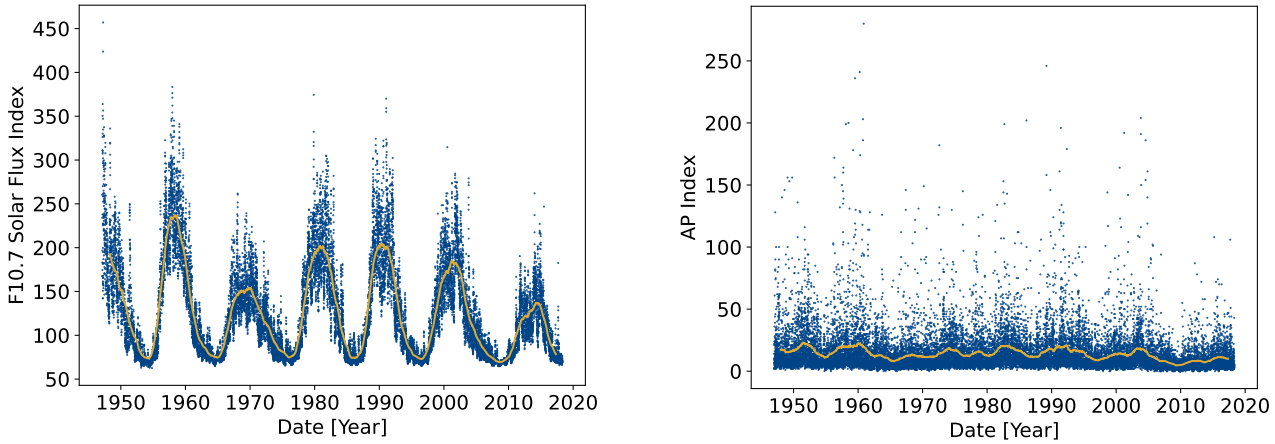


Figure 6: Local density for CHAMP from NRLMSISE-00 and TLE estimates. Residuals illustrate stochastic model error. Data from Reference 2.



(a) Solar flux index (F10.7).

(b) Geomagnetic index (AP).

Figure 7: Daily F10.7 and AP indices with 2-year moving average (yellow). Data from the National Centers for Environmental Information (NCEI) [29].

gravity fields remain poorly characterized. In such cases, stochastic processes can be used to represent the uncertain gravity environment itself, enabling missions and guidance, navigation, and control (GNC) schemes to be designed with robustness against these unknowns.

Other examples include stochastic modeling of solar radiation pressure for objects with unknown attitude states [34], spacecraft outgassing [35], dust or particle impacts in cometary or cryovolcanic environments [36, 37], and structural effects such as vibrations, slosh, magnetic torques, thruster leakage, or gas impingement [38–40].

5. Implementation in Dynamics Toolkits

Most dynamics simulation toolkits, including those used in astrodynamics, do not natively support states whose evolution is governed by SDEs. This section introduces a strategy for extending such tools to handle stochastic dynamics. The approach builds on the model-based simulation framework proposed in Reference 20, which is briefly reviewed here.

Model-Based Paradigm for ODEs

The fundamental task of a dynamics simulator is to evaluate the system's governing equations of motion:

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(t, \boldsymbol{x})\mathrm{d}t \tag{29}$$

where the state vector $\boldsymbol{x}$ depends on the system being simulated. For a point-mass spacecraft, $\boldsymbol{x}$ may contain only position and velocity. For a rigid-body, it additionally includes attitude and angular velocity. More complex spacecraft (e.g., with articulated panels, robotic manipulators, or fuel tanks) require further states such as joint angles, actuator dynamics, or propellant mass.

The main role of the simulation toolkit is to evaluate $\boldsymbol{f}(t, \boldsymbol{x})$ efficiently and accurately. In addition to performance, usability is crucial: the framework should be easy to understand, extend, and reuse. To this end, Reference 20 proposed a model-based architecture, implemented in the Basilisk astrodynamics toolkit [19], similar in philosophy to MATLAB's Simulink [41] or JPL's DARTS [10].

In this paradigm, the computation of $\boldsymbol{f}$ is decomposed into self-contained *models* that exchange data through standard-

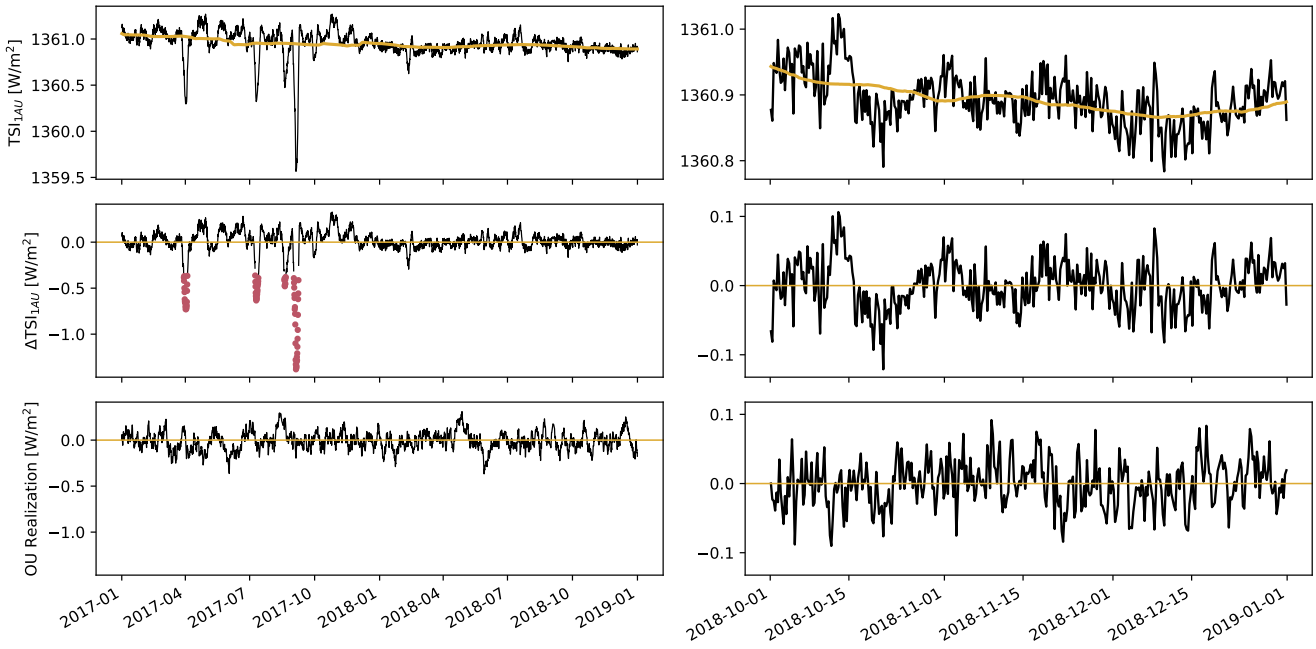


Figure 8: Top: Total Solar Irradiance (TSI) at 1 AU with moving averages. Middle: deviations from moving average, with outliers in red. Bottom: realizations of OU processes fitted to $\Delta\mathrm{TSI}_{1AU}$. Data from Reference 30.

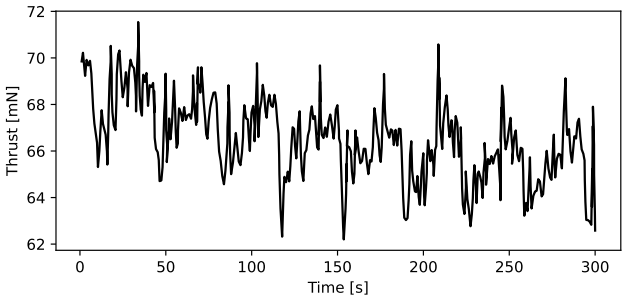


Figure 9: Experimental thrust data from a Hall thruster. Data from Reference 5.

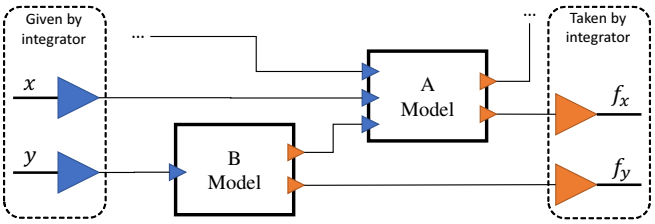


Figure 10: Model-based paradigm for ODE simulation. Each model (black square) exchanges inputs (blue) and outputs (orange) via messages. Connections to omitted models are indicated by "...".

ized messages. For instance, a "guidance model" outputs a desired joint angle; a "PID model" takes this reference and the current angle to output a voltage; a "DC motor model" converts the voltage into torque; and a "multibody dynamics model" converts torque into joint acceleration. Each model computes only its local contribution, while the simulator collects all contributions to form $\mathrm{d}\boldsymbol{x}/\mathrm{d}t$.

In addition to physical states (position, attitude, joint angles, etc.), models may introduce auxiliary states. For example, an analog PID controller requires an integral-error continuous state. The framework in Reference 20 allows models to declare such auxiliary states, receive their values as inputs, and return their derivatives for integration.

Figure 10 illustrates this approach for

$$\mathrm{d}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_x(\cdot) \\ f_y(\cdot) \end{bmatrix} \mathrm{d}t \tag{30}$$

where f_x and f_y are the time derivatives of the states x and y respectively.

Model A declares and updates x, model B similarly handles y, and both can exchange data. At each integrator step, the toolkit supplies the current state values, calls each model to compute the derivatives, and assembles $\mathrm{d}\boldsymbol{x}/\mathrm{d}t$ for numerical integration.

Extending to SDEs

The above framework addresses deterministic states. Extending it to SDEs requires handling both drift and diffusion terms:

$$\mathrm{d}\boldsymbol{X} = f(t, \boldsymbol{X})\mathrm{d}t + \sum_i g_i(t, \boldsymbol{X})\mathrm{d}W_i. \tag{31}$$

Thus, models must not only declare their drift contributions but also specify how each state is influenced by one or more noise sources (diffusion contributions). By default, noise sources are independent; e.g., an IMU bias and a perturbing force would evolve under separate Wiener processes. However, for generality, the framework must allow multiple states to share a noise source.

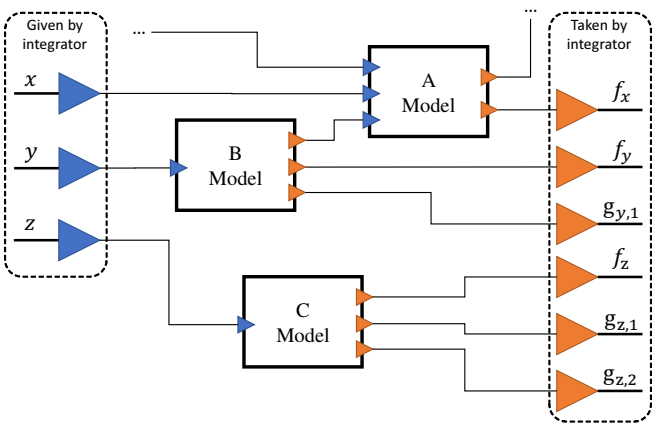


Figure 11: Extension of the model-based paradigm to SDEs. Models provide both drift and diffusion contributions.

	a	b	c
I	1	2	3
II	4	5	6
III	-	-	7

Table 1: Implementation approaches. Rows: drift/diffusion list structure. Columns: model reuse strategy. "-" indicates infeasible combinations.

Figure 11 illustrates a simulation of

$$\mathrm{d}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \mathrm{d}t + \begin{bmatrix} 0 \\ g_{y,1} \\ g_{z,1} \end{bmatrix} \mathrm{d}W_1 + \begin{bmatrix} 0 \\ 0 \\ g_{z,2} \end{bmatrix} \mathrm{d}W_2. \qquad (32)$$

Here, y and z share noise source W_1, while only z is also driven by W_2.

For some stochastic integrators (e.g., Euler-Maruyama), it suffices to compute drift and diffusion simultaneously. For others, drift and diffusion may need to be evaluated at different states or times [24–26]. Evaluating both together can waste computation, especially if computing the drift is expensive and only diffusion is required.

Implementation Options

Three strategies can be used to manage drift and diffusion evaluations:

(I) Maintain independent model lists for drift and for each noise source.
(II) Use one list for drift and a single list for all diffusions.
(III) Use a single list for both drift and diffusion.

Option I is the most computationally efficient; Option III the least.

Another design choice concerns whether models are reused:

(a) Separate models per list.
(b) Shared models, with different functions for drift and diffusions.
(c) Shared models, with one function reused for all lists.

Options (a) and (b) are more efficient than (c).

Table 1 summarizes all combinations. Figures 12a-12g illustrate them.

Implementation Approaches

The seven approaches summarized in Table 1 and illustrated in Figure 12 represent different ways of organizing drift and diffusion evaluations in a model-based simulation framework. Each balances three competing factors: computational efficiency, usability for the end user, and flexibility in simulation design.

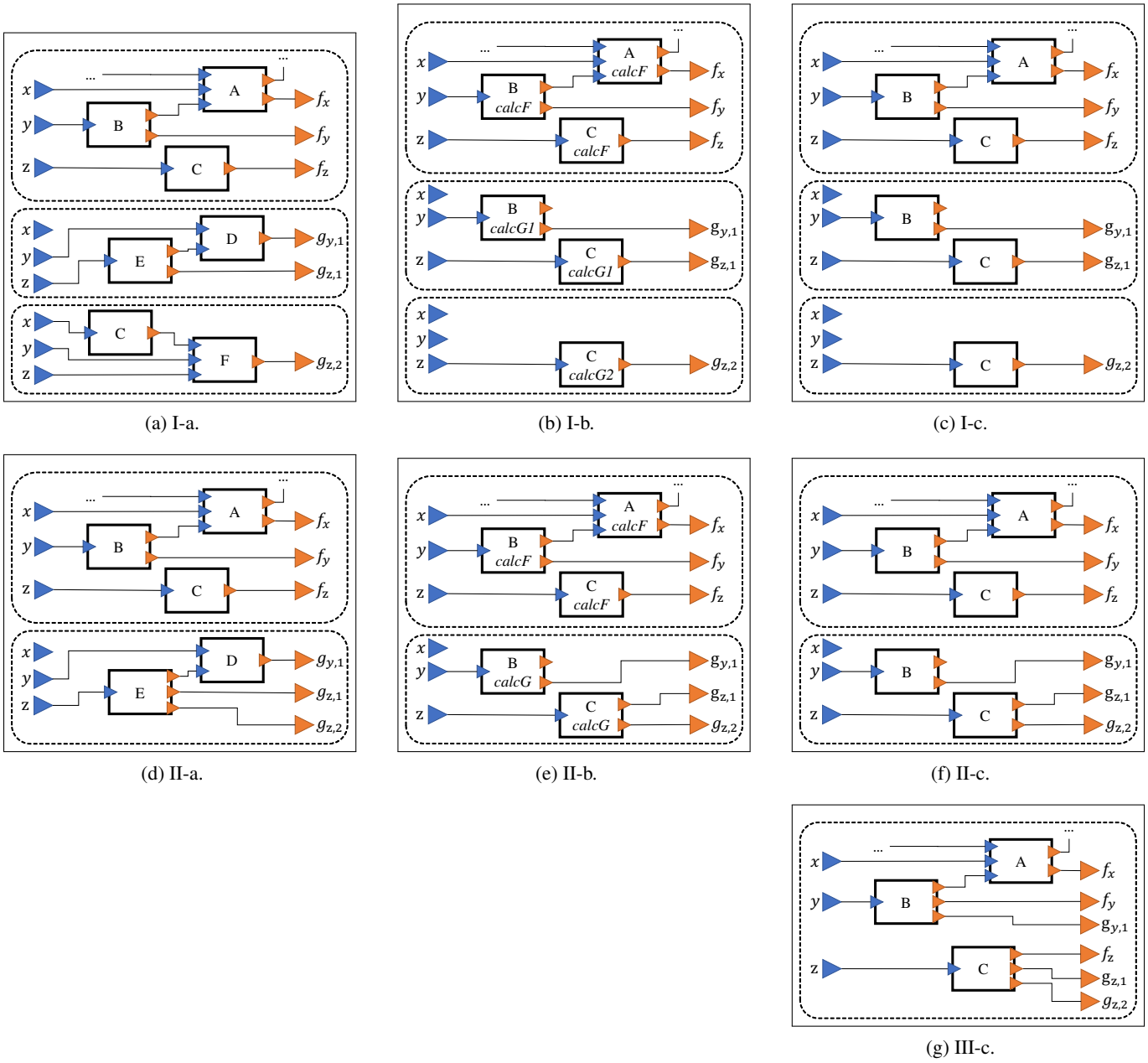
Approach 1 (Figure 12a) is the most computationally efficient and flexible. It maintains independent model lists for the drift term and for each diffusion source, with separate models in each list. This means that only the models required for a given calculation are executed, avoiding redundant work. It also allows drift and diffusion functions for the same state to be placed in entirely different models, giving maximum freedom in simulation design. The cost is usability: users must configure and maintain many different models and connectivity graphs, and states are no longer tied to a single model. This makes setup more complex and error-prone.

Approach 2 (Figure 12b) reduces this burden by reusing the same model instances across drift and diffusion lists. Each model defines multiple functions, such as one for the drift term and separate ones for each noise source, that are called depending on context. This keeps states tied to a single model while still allowing independent drift and diffusion evaluations, and it preserves much of the efficiency of Approach 1. The trade-off is a small reduction in flexibility, since all drift and diffusion functions for a state must now share the same connectivity.

Approach 3 (Figure 12c) simplifies things further by having each model implement only a single function, reused across both drift and diffusion lists. When invoked, the function must compute all drift and diffusion contributions together, even if only one of them is required. This increases usability, since model implementation is simpler, but at the cost of wasted computation. If a drift calculation is expensive but only a diffusion update is needed, unnecessary work will still be performed. This makes Approach 3 less efficient than Approaches 1 and 2, but simulations are easier to set up and maintain.

Approaches 4, 5, and 6 (Figures 12d-12f) follow the same logic as Approaches 1, 2, and 3, but group all diffusion terms into a single model list. This reduces the number of lists users must manage, improving usability by simplifying simulation setup. The downside is that the entire diffusion list is executed whenever any diffusion contribution is required, which introduces some redundant computation. Among these, Approach 4 is the most efficient but least user-friendly, Approach 5 balances efficiency with usability by allowing shared models with separate functions, and Approach 6 maximizes simplicity by requiring only one function per model, at the expense of some additional computation.

Finally, Approach 7 (Figure 12g) represents the baseline strategy originally introduced in Figure 11. Here, drift and diffusion values are computed together in a single list, with one function per model. This is the simplest option conceptually, requiring minimal changes to existing frameworks and offering the easiest setup for users. However, it is also the least efficient, since expensive drift calculations are repeated whenever diffusion values are needed, and vice versa. For simulations that include costly models such as high-fidelity gravity fields or multibody dynamics, this inefficiency can



(a) I-a. (b) I-b. (c) I-c.

(d) II-a. (e) II-b. (f) II-c.

(g) III-c.

Figure 12: Model-based simulation implementations for stochastic dynamics. Dashed rectangles denote model lists, which may run independently.

quickly become prohibitive.

In practice, the optimal choice depends on the goals of the simulation framework and the expected level of expertise of its users. Approaches 1 and 2 offer maximum efficiency and flexibility but impose significant complexity on the user. Approaches 3 and 6 provide a good compromise, retaining much of the efficiency while simplifying the workflow. Approach 7, while easiest to reason about, can be very inefficient. For Basilisk, Approach 6 was selected as the most balanced solution: it keeps states tied to a single model, requires only one additional model list beyond the drift, and avoids unnecessary drift evaluations when only diffusion values are needed, providing substantial performance gains without sacrificing usability.

The architectural choices described above are independent of the stochastic calculus interpretation adopted by the numerical integrator. In particular, the framework supports both Itô and Stratonovich formulations of stochastic differential equations, with the distinction enforced at the integration stage rather than in the model wiring or state definitions. Drift and diffusion contributions are exposed explicitly by models, allowing integrators to evaluate these terms at the appropriate states and times required by the chosen interpretation. In Basilisk, the current implementation focuses on Itô SDEs and includes support for the Euler–Maruyama method as well as two second-order weak stochastic Runge–Kutta integrators presented by Tang and Xiao for weak integration of Itô SDEs [26].

6. SIMULATION RESULTS

This section presents an example astrodynamics problem with stochastic dynamics. The chosen scenario is an uncontrolled atmospheric re-entry, representative of a piece of space debris or a satellite at the end of its operational life. The primary source of uncertainty is atmospheric density, modeled as a stochastic process.

A simplified dynamical model is used, since the goal is not to provide a precise de-orbit prediction but to illustrate qualitatively how stochastic modeling affects re-entry behavior. The body is assumed to be subject only to point-mass gravity and atmospheric drag. Gravity is modeled as

$$\boldsymbol{F}_{\text{grav}} = -m\frac{GM_\oplus}{||\boldsymbol{r}||^3}\boldsymbol{r} \tag{33}$$

where $\boldsymbol{r}$ is the position vector, m the spacecraft mass, and $GM_\oplus = 3.986 \cdot 10^5$ km^3/s^2 Earth's standard gravitational parameter. Drag is modeled with a simple cannonball formulation:

$$\boldsymbol{F}_{\text{drag}} = -\frac{1}{2}m\frac{\rho}{\beta}\boldsymbol{v}_{\text{rel}}||\boldsymbol{v}_{\text{rel}}|| \tag{34}$$

where $\beta = 30$ kg/m^2 is the ballistic coefficient and $\boldsymbol{v}_{\text{rel}}$ is the velocity relative to the atmosphere. For simplicity, winds are neglected and the atmosphere is assumed stationary in the inertial frame, so $\boldsymbol{v}_{\text{rel}} \approx \boldsymbol{v}$.

Atmospheric density ρ is decomposed into a deterministic "nominal" $\bar{\rho}$ and a stochastic correction κ:

$$\rho = \bar{\rho}(1+\kappa) \tag{35}$$

$$\mathrm{d}\kappa = -\frac{1}{\tau_\kappa}\kappa \mathrm{d}t + \sqrt{\frac{2}{\tau_\kappa}}\sigma_{\kappa,\text{st}}\mathrm{d}W \tag{36}$$

$$\kappa(t_0) \sim \mathcal{N}(0, \sigma^2_{\kappa,\text{st}}) \tag{37}$$

Here, κ evolves as an Ornstein-Uhlenbeck process with correlation time τ_κ and stationary standard deviation $\sigma_{\kappa,\text{st}}$. Short-term variations in density reported in Reference 1 range from 30% to 100%, motivating $\sigma_{\kappa,\text{st}} = 0.15$ as a representative choice. Half-life values of 1.8, 18, and 180 minutes have been reported in Reference 42. Because the OU autocorrelation is $\rho(\Delta t) = e^{-|\Delta t|/\tau}$, the half-life $t_{1/2}$ and τ are related by $t_{1/2} = \tau \ln 2$, yielding $\tau_\kappa = 74.9, 748.5, 7485$ s. The limiting case $\tau_\kappa \to \infty$ is also considered, which corresponds to a constant but random κ.

The nominal density profile is given by an exponential atmosphere:

$$\bar{\rho}(h) = \bar{\rho}_0 e^{-(h-h_0)/H} \tag{38}$$

with parameters $\bar{\rho}_0 = 3.396 \cdot 10^{-6}$ kg/m^3, $h_0 = 90$ km, and $H = 5.382$ km from Reference 43. Altitude is $h = ||\boldsymbol{r}|| - R_\oplus$ with $R_\oplus = 6371$ km.

Initial conditions correspond to a circular equatorial orbit with an altitude of 150 km:

$$\boldsymbol{r}_0 = [6521, 0, 0] \text{ km} \tag{39}$$

$$\boldsymbol{v}_0 = [0, 7.818, 0] \text{ km/s} \tag{40}$$

The complete set of first-order SDEs is

$$\mathrm{d}\begin{bmatrix}\boldsymbol{r}\\ \boldsymbol{v}\\ \kappa\end{bmatrix} = \begin{bmatrix}\boldsymbol{v}\\ -\frac{GM_\oplus}{||\boldsymbol{r}||^3}\boldsymbol{r} - \frac{1}{2}\frac{\rho(\boldsymbol{r},\kappa)}{\beta}\boldsymbol{v}||\boldsymbol{v}||\\ -\frac{1}{\tau_\kappa}\kappa\end{bmatrix}\mathrm{d}t + \begin{bmatrix}0\\ 0\\ \sqrt{\frac{2}{\tau_\kappa}}\sigma_{\kappa,\text{st}}\end{bmatrix}\mathrm{d}W \tag{41}$$

with

$$\rho(\boldsymbol{r},\kappa) = (1+\kappa)\bar{\rho}(\boldsymbol{r}) \tag{42}$$

$$\bar{\rho}(\boldsymbol{r}) = \bar{\rho}_0 e^{-(||\boldsymbol{r}|| - R_\oplus - h_0)/H} \tag{43}$$

$$\boldsymbol{r}(t_0) = \boldsymbol{r}_0 \tag{44}$$

$$\boldsymbol{v}(t_0) = \boldsymbol{v}_0 \tag{45}$$

$$\kappa(t_0) \sim \mathcal{N}(0, \sigma^2_{\kappa,\text{st}}) \tag{46}$$

Figure 13 illustrates how this problem can be implemented in a model-based toolkit such as Basilisk, using Approach 6 described in Table 1. Each block corresponds to a self-contained component of the simulation, and the connections between them define the flow of information. This modular setup allows models to be replaced, removed, or reused with minimal changes to the surrounding architecture. A more detailed description of each block follows.

The "Atm Model" provides the deterministic background density. It receives the spacecraft's position vector $\boldsymbol{r}$ and computes the nominal density $\bar{\rho}$ at the corresponding altitude. Note that this model can easily be replaced by a more sophisticated density model (e.g., NRLMSISE-00), without altering the rest of the simulation.

The "OU Model" applies the stochastic correction κ to the nominal density. It takes $\bar{\rho}$ as input and returns the corrected value $\rho = (1+\kappa)\bar{\rho}$. Additionally, it determines the evolution of κ according to the Ornstein-Uhlenbeck process by defining both its drift and diffusion terms. The model is generic: the same implementation could be used to apply an OU-based stochastic correction to any scalar quantity in the simulation (for example, to represent thrust noise or sensor bias). On the other hand, if this model were omitted, the simulation would revert to a purely deterministic density.

The "Gravity Model" computes the gravitational acceleration. In this example, it uses a point-mass formulation, but the modular architecture allows this model to be swapped for higher-fidelity alternatives, such as spherical harmonics or polyhedral gravity models, while leaving other parts of the simulation unchanged.

The "Drag Model" computes the atmospheric drag force. It receives both the density ρ from the OU model and the velocity $\boldsymbol{v}$ from forward kinematics, and returns the drag force. As with the gravity model, alternative formulations (such as a flat-plate model) can be substituted if higher fidelity is required.

The "Fwd Kin" (forward kinematics) model translates the generalized coordinate vector $\boldsymbol{q}$ into derived kinematic quantities, in this case position $\boldsymbol{r}$ and velocity $\boldsymbol{v}$. In a more complex spacecraft, $\boldsymbol{q}$ could include rotational states or joint angles, and the forward kinematics model would output the corresponding positions, velocities, or orientations needed by other models. The "Fwd Dyn" (forward dynamics) model collects the outputs of the force models (gravity and drag in this case) and combines them into the time derivative of the

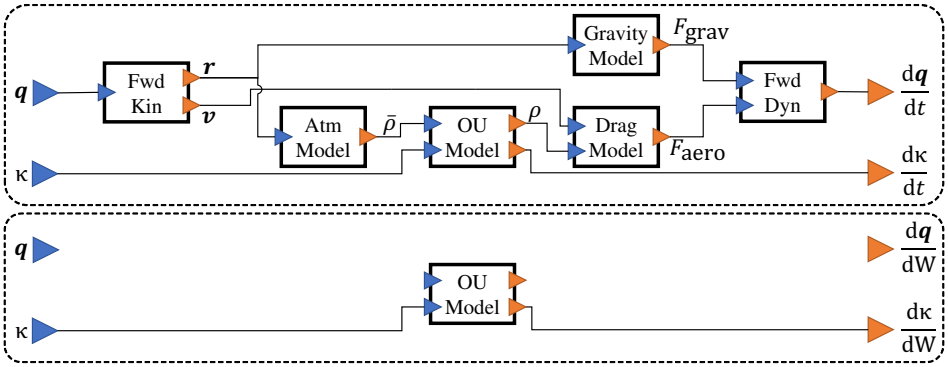


Figure 13: Model-based simulation setup of a stochastic de-orbit scenario using Approach 6 (see Table 1). Each block represents a self-contained model that can be replaced, removed, or reused.

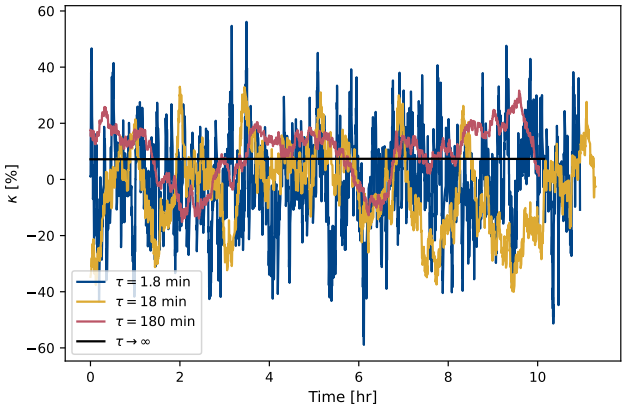


Figure 14: Sample realizations of the density correction factor κ for different τ_κ. Shorter τ_κ produces more rapid fluctuations; $\tau_\kappa \to \infty$ yields a constant bias.

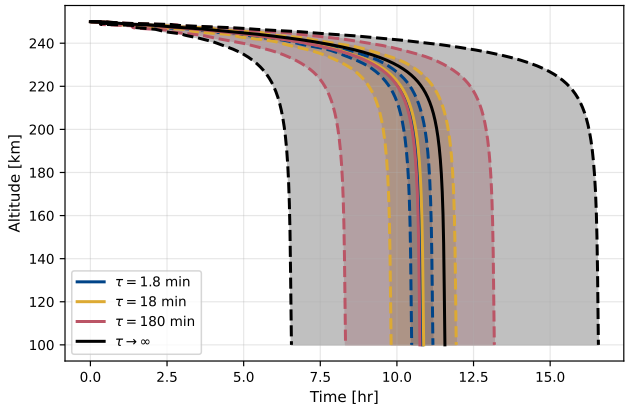


Figure 15: Monte Carlo results for 1000 realizations of the de-orbit problem under different τ_κ. Solid lines show mean altitude evolution; shaded regions denote 3σ bounds.

generalized coordinates $d\boldsymbol{q}/dt$. These models are described at length in Reference 20.

Together, these blocks illustrate the plug-and-play advantages of the model-based approach. Each model is responsible for a single, well-defined task, and the connections between them are standardized. A user can replace the atmospheric model, change the drag formulation, or add new perturbation sources without reconfiguring the rest of the simulation. Moreover, since only the OU model defines diffusion terms, stochastic integration is computationally efficient: expensive evaluations of the gravity or drag models are bypassed when only diffusion is required.

Figure 14 shows realizations of κ for different correlation times. Short τ_κ produces rapidly varying processes, while long τ_κ results in smoother, slowly varying corrections. In the limit $\tau_\kappa \to \infty$, κ is constant but randomly drawn at initialization.

Monte Carlo simulations with 1000 realizations per τ_κ reveal how τ_κ shapes re-entry predictions. Figure 15 shows the mean and 3σ bounds for altitude vs. time. Table 2 reports statistics of the time to reach 100 km altitude. Shorter τ_κ produces smaller spreads due to rapid fluctuations that cancel each other, while using a constant κ ($\tau_\kappa \to \infty$) overestimates the variability.

τ_κ	E[$t(h = 100$ km)] [hr]	Std[$t(h = 100$ km)] [hr]
74.9 s	10.83	0.12
748.5 s	10.87	0.35
7485 s	10.75	0.81
∞	11.57	1.67

Table 2: Mean and standard deviation of de-orbit time to 100 km altitude for different τ_κ.

These results highlight the importance of modeling atmospheric density as a stochastic process rather than as a constant random parameter. The assumed correlation time strongly influences the predicted dispersion in de-orbit times. Neglecting temporal variability by assuming $\tau_\kappa \to \infty$ produces a misleading characterization of trajectory uncertainty.

Scripts to run this example scenario are available in the public repository for Basilisk.

7. Conclusion

This paper presents a framework for incorporating stochastic uncertainties in astrodynamics through stochastic differential equations. The discussion begins with the mathematical foundations of SDEs and proceeds to relevant stochastic processes such as the Ornstein-Uhlenbeck and higher-order

Gauss-Markov models. Several sources of uncertainty in spacecraft dynamics are examined, including atmospheric variability, solar flux, thruster performance, prescribed motion errors, and unmodeled perturbations. Across these domains, stochastic processes provide a principled means of representing uncertain parameters and assessing their impact on orbital dynamics.

To enable their use in practice, the paper proposes an extension of model-based simulation paradigms to support stochastic dynamics. Alternative implementation strategies are analyzed in terms of computational efficiency, usability, and flexibility, and an approach suited for the Basilisk framework is described in detail. A de-orbit case study illustrates how stochastic atmospheric density can be integrated into simulation, and how different correlation times in the density process lead to markedly different predictions of orbital lifetime. The results demonstrate that treating uncertain inputs as random constants mischaracterizes orbital evolution, while SDE-based models yield more accurate and physically-meaningful outcomes.

The findings of this work argue for the systematic adoption of stochastic processes in modeling astrodynamical uncertainties. Their inclusion enables higher-fidelity simulations, better quantification of risk, and improved robustness in mission design and analysis.

ACKNOWLEDGEMENTS

The research was partly supported by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

Generative AI (ChatGPT 5) was used in the preparation of this manuscript for editing and grammar enhancement purposes.

REFERENCES

[1] L. Qian and S. C. Solomon, "Thermospheric density: An overview of temporal and spatial variations," *Space Science Reviews*, vol. 168, no. 1, pp. 147–173, June 2012. [Online]. Available: https://doi.org/10.1007/s11214-011-9810-z

[2] E. Doornbos, "Thermospheric density and wind determination from satellite dynamics," PhD thesis, Delft University of Technology, 2008. [Online]. Available: https://resolver.tudelft.nl/uuid:33002be1-1498-4bec-a440-4c90ec149aea

[3] C. R. McInnes, *Solar Sailing*, 1st ed., ser. Springer Praxis Books. Springer London, 1999, jointly published with Praxis Publishing, UK. 115 b/w illustrations. [Online]. Available: https://doi.org/10.1007/978-1-4471-3992-8

[4] G. Kopp and J. L. Lean, "A new, lower value of total solar irradiance: Evidence and climate significance," *Geophysical Research Letters*, vol. 38, no. 1, 2011. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010GL045777

[5] S. Xu, Z. Zhang, Z. Zhang, W. Yang, H. Tang, and W. Y. L. Ling, "Time-frequency-domain method for thrust noise characteristics of electric thrusters," *Acta Astronautica*, vol. 188, pp. 308–325, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S009457652100391X

[6] J. Ziemer, C. Marrese-Reading, C. Dunn, A. Romero-Wolf, C. Cutler, S. Javidnia, T. Li, I. Li, G. Franklin, P. Barela, O. Hsu, P. Maghami, J. O'Donnell, J. Slutsky, J. I. Thorpe, N. Demmons, and V. Hruby, "Colloid microthruster flight performance results from space technology 7 disturbance reduction system," in *Proceedings of the International Electric Propulsion Conference (IEPC 2017)*. Atlanta, GA, USA: NASA, Oct. 2017, document ID: 20170010216. Report Number: GSFC-E-DAA-TN47585. [Online]. Available: https://ntrs.nasa.gov/citations/20170010216

[7] N. P. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2149–2154 vol.3, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:206941306

[8] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:5230692

[9] NVIDIA, "Isaac Sim." [Online]. Available: https://github.com/isaac-sim/IsaacSim

[10] J. Garcia-Bonilla, C. Leake, A. Elmquist, T. D. Hasseler, V. Steyert, A. Gaut, and A. Jain, "Dshell-darts: A reusability-focused multi-mission aerospace and robotics simulation toolkit," in *2025 IEEE Aerospace Conference*, 2025, pp. 1–13.

[11] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. A. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An open source multi-physics dynamics engine," in *International Conference on High Performance Computing in Science and Engineering*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:218062602

[12] H. Elmqvist, S. E. Mattsson, and M. Otter, "Modelica – a language for physical system modeling, visualization and interaction," in *Proceedings of the 1999 IEEE International Symposium on Computer-Aided Control System Design (CACSD '99)*, Maui, Hawaii, USA, Aug. 1999, pp. 630–639, [Online]. Available: https://dx.doi.org/10.1109/CACSD.1999.808720; Modelica homepage: https://modelica.org/.

[13] M. Bouissou, H. Elmqvist, M. Otter, and A. Benveniste, "Efficient Monte Carlo simulation of stochastic hybrid systems," in *The 10th International Modelica Conference 2014*. Lund, Sweden: Hubertus Tummescheit and Karl-Erik Årzén, Mar. 2014. [Online]. Available: https://inria.hal.science/hal-01182410

[14] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," https://drake.mit.edu, 2019, accessed: 2025-10-02.

[15] S. P. Hughes, R. H. Qureshi, S. D. Cooley, and J. J. Parker, *Verification and Validation of the General Mission Analysis Tool (GMAT)*, 2014. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2014-4151

[16] A. Wall, *Systems Tool Kit (STK)*. CRC Press, 1 2024, pp. 11–32. [Online]. Available:

https://www.taylorfrancis.com/chapters/edit/10.1201/9781003321811-4/systems-tool-kit-stk-alexis-wall

[17] G. Dell, M. Hametz, P. Noonan, L. Newman, D. Folta, and J. Bristow, *EOS AM-1 and EO-1 support using FreeFlyer and AutoCon*, 1998. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1998-4196

[18] E. Stoneking, "42: An open-source simulation tool for study and design of spacecraft attitude control systems," Lecture presentation at Georgia Institute of Technology, NASA Goddard Space Flight Center, Atlanta, GA, United States, Presentation GSFC-E-DAA-TN52069, feb 2018, document ID: 20180000954. Work of the U.S. Government, public use permitted. [Online]. Available: https://ntrs.nasa.gov/citations/20180000954

[19] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A flexible, scalable and modular astrodynamics simulation framework," *Journal of Aerospace Information Systems*, vol. 17, no. 9, pp. 496–507, 2020. [Online]. Available: https://doi.org/10.2514/1.I010762

[20] J. Garcia-Bonilla and H. Schaub, "A message-passing simulation framework for generally articulated spacecraft dynamics," in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Boston, MA, USA, 2025.

[21] K. A. Myers and B. D. Tapley, "Dynamical model compensation for near-earth satellite orbit determination," *AIAA Journal*, vol. 13, no. 3, pp. 343–349, 1975. [Online]. Available: https://doi.org/10.2514/3.49702

[22] J. M. Leonard, F. G. Nievinski, and G. H. Born, "Gravity error compensation using second-order gauss-markov processes," *Journal of Spacecraft and Rockets*, vol. 50, no. 1, pp. 217–229, 2013. [Online]. Available: https://doi.org/10.2514/1.A32262

[23] S. Särkkä and A. Solin, *Parameter Estimation in SDE Models*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019, p. 234–250.

[24] P. E. Kloeden and R. Pearson, "The numerical solution of stochastic differential equations," *The ANZIAM Journal*, vol. 20, no. 1, pp. 8–12, 1977.

[25] A. Rößler, "Second order runge–kutta methods for itô stochastic differential equations," *SIAM Journal on Numerical Analysis*, vol. 47, no. 3, pp. 1713–1738, 2009.

[26] X. Tang and A. Xiao, "Efficient weak second-order stochastic runge–kutta methods for itô stochastic differential equations," *BIT Numerical Mathematics*, vol. 57, no. 1, pp. 241–260, 2017. [Online]. Available: https://doi.org/10.1007/s10543-016-0618-9

[27] J. P. N. Bishwal, *Parameter Estimation in Stochastic Differential Equations*, 1st ed., ser. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2008, vol. 1923. [Online]. Available: https://doi.org/10.1007/978-3-540-74448-1

[28] J. N. Nielsen, H. Madsen, and P. C. Young, "Parameter estimation in stochastic differential equations: An overview," *Annual Reviews in Control*, vol. 24, pp. 83–94, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578800900178

[29] N. C. for Environmental Information, "Data and products," Dec 2005. [Online]. Available: https://www.ngdc.noaa.gov/ngdcinfo/onlineaccess.html

[30] G. Kopp, "Sorce level 3 total solar irradiance 6-hourly means, version 020," https://doi.org/10.5067/TSLKVMDR7LPX, Greenbelt, MD, USA, 2025, accessed: 2025-09-18.

[31] S. M. Choi and C. Bach, "Experimental investigation of pwm throttling in a 50-newton-class htp monopropellant thruster: Analysis of pressure surges and oscillations," *Aerospace*, vol. 12, no. 5, 2025. [Online]. Available: https://www.mdpi.com/2226-4310/12/5/418

[32] J. Mansell, D. A. Spencer, B. Plante, A. Diaz, M. Fernandez, J. Bellardo, B. Betts, and B. Nye, *Orbit and Attitude Performance of the LightSail 2 Solar Sail Spacecraft*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2020-2177

[33] J. Garcia-Bonilla, L. Carzana, and J. Heiligers, "Uncertainty quantification for solar sails in the near-earth environment," in *Proceedings of the 6th International Symposium on Space Sailing (ISSS 2023)*, New York, USA, June 2023. [Online]. Available: https://www.researchgate.net/publication/381251800_Uncertainty_Quantification_for_Solar_Sails_in_the_Near-Earth_Environment

[34] M. Soppet and C. Frueh, *Stochastic Modeling of Solar Radiation Pressure on High Area-Mass Ratio Objects*, 2016. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2016-5201

[35] M. Hässig, K. Altwegg, H. Balsiger, U. Calmonte, A. Jäckel, B. Schläppi, T. Sémon, P. Wurz, J. J. Berthelier, J. De Keyser, B. Fiethe, S. A. Fuselier, U. Mall, H. Rème, and M. Rubin, "Spacecraft outgassing, a largely underestimated phenomenon," in *2011 2nd International Conference on Space Technology*, 2011, pp. 1–4.

[36] I. Fodde, J. Feng, and M. Vasile, "Uncertainty maps for motion around binary asteroids," *Celestial Mechanics and Dynamical Astronomy*, vol. 134, no. 5, p. 41, 2022. [Online]. Available: https://doi.org/10.1007/s10569-022-10096-2

[37] M. Wittal, K. Gucwa, and D. Batcheldor, "Orbital drag near small bodies due to lofted fines from surface activity," 09 2022.

[38] M. Reyhanoglu, *Modelling and Control of Space Vehicles with Fuel Slosh Dynamics*, 02 2011.

[39] L. Pettazzi, H. Krüger, S. Theil, and D. Izzo, "Electrostatic force for swarm navigation and reconfiguration," *Acta Futura*, vol. 4, pp. 80–86, 01 2008.

[40] A. Lassakeur and C. Underwood, "Determination and mitigation of the residual magnetic dipole moment of cubesats for improved attitude stability," 10 2021.

[41] MathWorks, "Simulink: Simulation and model-based design," https://www.mathworks.com/products/simulink.html, 2025, accessed: 2025-09-18.

[42] A. T. Hiatt, "Deriving atmospheric density estimates using satellite precision orbit ephemerides," Master's thesis, University of Kansas, Lawrence, KS, 2009.

[43] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, 4th ed., ser. Space Technology Library. Microcosm Pres, 2013.

BIOGRAPHY

Juan Garcia-Bonilla is a Robotics Technologist at the Jet Propulsion Laboratory. He holds a B.S. in aerospace engineering from the University Carlos III of Madrid and a M.S. in aerospace engineering from the Delft University of Technology. He is pursuing a Ph.D. at the University of Colorado Boulder under the guidance of Dr. Hanspeter Shaub, where he focuses on developing advanced astrodynamics simulation tools.

Hanspeter Schaub is a distinguished professor and chair of the University of Colorado aerospace engineering sciences department. He holds the Schaden leadership chair. He has over 30 years of research experience, of which 4 years are at Sandia National Laboratories. His research interests are in astrodynamics, relative motion dynamics, charged spacecraft motion as well as spacecraft autonomy. This has led to about 228 journal and 371 conference publications, as well as a 4th edition textbook on analytical mechanics of space systems. Dr. Schaub has been the ADCS lead in the CICERO mission, the ADCS algorithm lead on a Mars mission and supporting ADCS for a new asteroid mission. In 2023 he won the Hazel Barnes Prize, the top award granted to faculty at the University of Colorado. He has been awarded the H. Joseph Smead Faculty Fellowship, the Provost's Faculty Achievement Award, the faculty assembly award for excellence in teaching, as well as the Outstanding Faculty Advisor Award. He is a fellow of AIAA and AAS, and has won the AIAA/ASEE Atwood Educator award, AIAA Mechanics and Control of Flight award, as well as the Collegiate Educator of the Year for the AIAA Rocky Mountain section. In 2025 he became a member of the National Academy of Engineering.