Gremlin: AI-Based Adversarial Stress-Testing for Autonomous Space Systems

Marie Ethvignot[1,2], Hiro Ono[2], Richard R. Rieber[2]
[1]École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[2]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
marie.ethvignot@epfl.ch

***Abstract*—As robotic space missions push into increasingly remote and poorly characterized environments, onboard autonomy is becoming essential. Yet the verification and validation (V&V) methods used to certify autonomous systems before launch were designed for largely pre-programmed spacecraft and do not scale to systems whose behavior evolves in response to unknown conditions. This work introduces *Gremlin*, an adversarial framework for simulation-based V&V of autonomous spacecraft. Acting as an intelligent adversary within a mission simulator, Gremlin injects structured disturbances during execution and uses Monte Carlo Tree Search to direct exploration toward scenarios that degrade mission performance. Disturbances are modeled as continuous correlated functions through a Gaussian process prior, and a global Mahalanobis budget constraint ensures that identified scenarios remain statistically plausible under the assumed uncertainty model. This is a deliberate design choice: in physical systems, the most operationally relevant failure scenarios are not the most extreme ones but the plausible ones that happen to stress the system in unexpected ways. Gremlin therefore searches for scenarios at the boundary of plausible behavior rather than pushing toward statistically unlikely extremes. Evaluated on a Uranus atmospheric entry and relay communication scenario, Gremlin exposes a failure mode in which the relay window simultaneously shifts and compresses due to coupled aerodynamic and pressure-based termination effects, a scenario that independent parameter sweeps and standard sensitivity analysis are unlikely to uncover, as it requires a correlated multi-event disturbance, resulting in a 19.7% reduction in science data return under a statistically plausible atmospheric disturbance. While demonstrated on atmospheric entry, Gremlin applies to any autonomous system evaluable in simulation under uncertainty, including rover traverse planning, autonomous rendezvous, or onboard science scheduling. As autonomy becomes more capable and adaptive, the tools used to certify it must become more intelligent too.**

***Index Terms*—space robotics, autonomy, verification and validation, adversarial testing, Monte Carlo Tree Search, Gaussian processes, planetary exploration**

I. Introduction

Autonomous robotic systems are increasingly deployed in environments that cannot be fully characterized before operation, from planetary surfaces and atmospheric probes to underwater vehicles and aerial robots operating in unknown terrain. In all these settings, a shared challenge arises: how do you verify and validate a system whose behavior will evolve in response to conditions that were not fully known at design time?

This challenge is especially acute for robotic space exploration. Future missions will rely on autonomous systems to operate in distant, uncertain environments such as planetary atmospheres and icy moons where real-time human intervention is impossible [1], [2]. Rather than incrementally reducing environmental uncertainty through successive missions as has been done for Mars, future one-shot missions must deploy adaptive systems capable of modifying their behavior *in situ* [1], [3]. Traditional V&V relies on requirement-based testing and manually designed scenarios, practices that do not scale when autonomy adapts online to uncertain conditions. As systems become more adaptive, the space of possible behaviors grows rapidly and human-designed test cases can cover only a vanishingly small fraction of it. This gap is not unique to space: any autonomous system operating under significant environmental uncertainty faces the same challenge between what can be tested before deployment and what will actually be encountered.

This paper introduces Gremlin, a framework that addresses this by placing an adversarial agent inside the simulator, searching intelligently for disturbance sequences that stress autonomous mission performance.

II. The Gremlin Framework

A. Core Idea

Gremlin operates as a second player inside a mission simulation. As illustrated in Fig. 1, the onboard autonomy and Gremlin interact in a closed loop: the autonomous system acts, Gremlin reacts by selecting a disturbance, and the simulator propagates the system forward. Over a full simulation run, Gremlin constructs an entire disturbance sequence and observes its effect on mission performance. The goal is not to find the single most extreme disturbance at each step, but to find the *sequence* of disturbances that together cause the autonomy to fail in a realistic and informative way.

B. Structured, Plausible Disturbances

Two design choices distinguish Gremlin from Adaptive Stress Testing (AST) [4], which searches for failure trajectories but typically operates over discrete, uncorrelated disturbance values.

Continuous, correlated disturbances. Physical uncertainties such as atmospheric density, terrain properties, or sensor drift vary continuously and are correlated over time or space. Treating them as independent scalar perturbations can produce sequences that are adversarial on paper but physically and

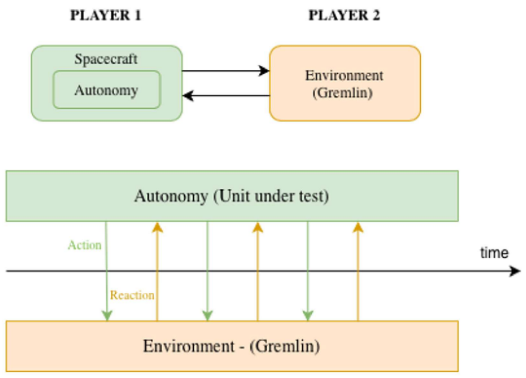

Fig. 1. Gremlin interacts with the onboard autonomy in closed loop, selecting disturbances at each decision point while the autonomy responds, allowing it to reason about multi-step consequences rather than perturbing one parameter at a time.

statistically very unlikely to happen. Gremlin models the disturbance process $g(\tau)$ as a Gaussian Process (GP),

$$g(\tau) \sim \mathcal{GP}\big(0,\, \sigma^2 k(\tau, \tau')\big)\,, \tag{1}$$

where $k(\tau, \tau')$ is a squared-exponential kernel encoding the expected smoothness of physical variation. The disturbed quantity is applied multiplicatively in log-space, $y(\tau) = m(\tau)\exp(g(\tau))$, ensuring scale-consistent relative perturbations that remain physically positive.

Global statistical plausibility. An unconstrained adversary can accumulate individually plausible disturbances into a globally implausible sequence. Gremlin prevents this by bounding the squared Mahalanobis distance of the full disturbance sequence from the GP prior (informally, the *insanity budget*),

$$\mathbf{d}_{0:T}^{\top} K^{-1} \mathbf{d}_{0:T} \le C, \tag{2}$$

where K is the GP covariance matrix and C is a tunable confidence threshold, restricting the search to statistically plausible scenarios rather than extreme corner cases the system would never realistically encounter.

C. Search via Monte Carlo Tree Search

Gremlin uses Monte Carlo Tree Search (MCTS) [5] with a search budget N_{iter} (number of tree iterations) to explore multi-step disturbance sequences, backpropagating mission performance scores to guide exploration toward adversarial regions. At each expansion step, a new disturbance value d_t is sampled via *sequential slicing*: the admissible interval given history $\mathbf{d}_{0:t-1}$ is,

$$d_t \in \left[\mu_t - \sqrt{S_t(C - \eta_{t-1})},\; \mu_t + \sqrt{S_t(C - \eta_{t-1})}\right], \tag{3}$$

where μ_t is the GP conditional mean, S_t the conditional variance, and η_{t-1} the cumulative Mahalanobis distance consumed. As the budget is used up the interval shrinks automatically, preventing drift into implausible territory.

TABLE I
RELAY DATA RECEIVED ACROSS DISTURBANCE STRATEGIES AND SEARCH BUDGETS ($n = 7$ SEEDS). Δ IS DEVIATION FROM NOMINAL (36.94 MB).

Method	Mean [Mb]	Std [Mb]	Δ [%]
$N_{iter} = 5$ *(low search budget)*			
MCTS + Indep.	28.57	4.32	-22.7
MCTS + GP	29.30	4.93	-20.7
MCTS + GP + Global (ours)	32.95	2.82	-10.8
$N_{iter} = 20$ *(high search budget)*			
MCTS + Indep.	26.40	1.21	-28.5
MCTS + GP	23.41	2.88	-36.6
MCTS + GP + Global (ours)	31.49	2.27	-14.8

III. APPLICATION: URANUS ATMOSPHERIC ENTRY

A. Scenario

The framework is evaluated on the Uranus Orbiter and Probe (UOP) mission concept, a NASA Flagship mission [7], simulated within MuSCAT [6], a mission-level simulator developed at JPL. The probe descends through the atmosphere of Uranus while transmitting science measurements in real time to an orbiter overhead. The probe has no onboard data storage: any data not transmitted during the relay window is permanently lost, making total science data received by the orbiter the primary performance metric.

The probe's onboard logic triggers key descent events automatically from real-time sensor readings: parachute deployment fires when dynamic pressure crosses 2874 Pa, heatshield separation follows 15 s later, and relay activation triggers when atmospheric pressure reaches 2 bar. The relay link operates at 25 kbps subject to geometric visibility. Gremlin perturbs the atmospheric density profile at five correlated decision points. The nominal profile comes from Voyager 2 radio occultation data [8], the only direct atmospheric measurements of Uranus ever taken, making atmospheric uncertainty a central and genuine source of mission risk.

B. Results

Table I compares three disturbance strategies at two search budgets, isolating the contribution of each design choice: MCTS + Indep. samples disturbances independently with no GP correlation and no global constraint; MCTS + GP adds inter-event correlation through the GP prior without a global bound; MCTS + GP + Global (ours) adds the Mahalanobis constraint. Across seeds, pure random sampling under the proposed disturbance model achieves a mean of 36.79 Mb, barely below nominal, confirming that the constrained disturbance space is rarely adversarial by chance and that structured search is necessary to exploit it.

Without a global bound, MCTS + GP drifts into increasingly extreme scenarios as budget grows, from -20.7% at $N_{\text{iter}} = 5$ to -36.6% at $N_{\text{iter}} = 20$, well beyond what the GP prior considers plausible. MCTS + GP + Global stabilizes as the insanity budget is consumed, finding consistently adversarial yet statistically plausible scenarios: the only method whose results are guaranteed to constitute genuine stress tests under

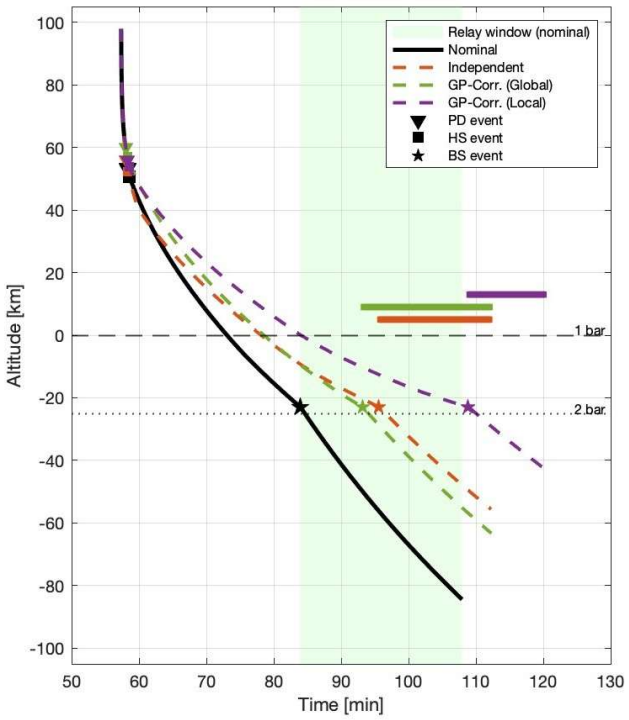

Fig. 2. Worst-case descent trajectory and relay window (horizontal bars) per disturbance strategy: MCTS + Indep. (orange), MCTS + GP (purple), and MCTS + GP + Global, our proposed method (green). MCTS + GP displaces the relay window the most, reflecting drift into globally implausible scenarios without the Mahalanobis constraint. MCTS + GP + Global stays closest to nominal while remaining adversarial within the plausibility budget.

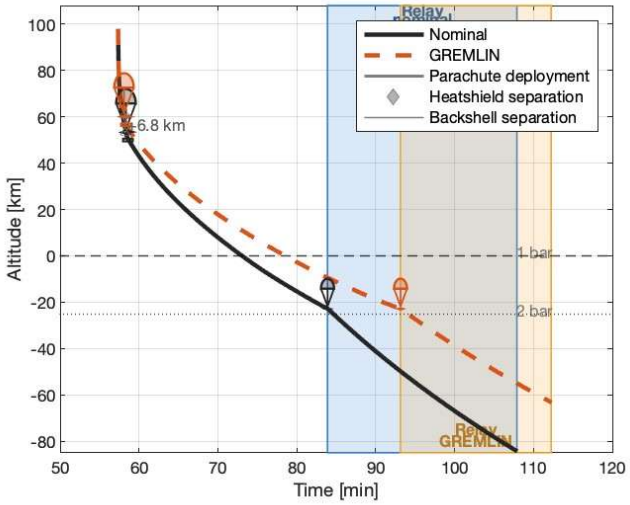

Fig. 3. Descent trajectories for the nominal (solid) and Gremlin-perturbed (dashed) scenarios (MCTS + GP + Global). Relay windows are shown as shaded regions (blue: nominal, orange: Gremlin). Key descent events are triggered at different altitudes and times under the denser atmosphere, shifting and compressing the relay window by 4.9 min and reducing science data return by 19.7%.

the assumed uncertainty model. Fig. 2 shows the worst-case descent trajectory and relay window per method, making the spatial consequence of each strategy immediately visible.

For the experimental setup described above, the most adversarial scenario found by MCTS + GP + Global reduces relay data to 29.65 Mb (-19.7%) under a statistically plausible atmospheric disturbance. The denser lower atmosphere triggers parachute deployment 6.79 km higher than nominal and heatshield separation 6.57 km higher. The relay activation trigger, governed by a pressure threshold rather than altitude, fires 9.25 min later. The result is a relay window that both shifts and compresses: relay duration drops from 24.0 min to 19.1 min, costing 4.9 min of transmission time and 7.29 Mb of science data, as shown in Fig. 3.

This failure mode is unlikely to surface in standard single-parameter sensitivity analysis, as it requires a correlated density profile that simultaneously affects multiple onboard triggers in ways that interact. Gremlin finds it by searching over structured, physically meaningful disturbance sequences rather than perturbing one parameter at a time.

IV. Discussion and Conclusion

The failure mode Gremlin identified would not be obvious from nominal mission analysis: an orbiter pre-programmed with a nominal contact schedule would be unlikely to anticipate a relay window that shifts and compresses simultaneously. It takes a correlated, multi-event disturbance to expose it, precisely the kind of scenario adversarial search is designed to find.

The framework is also tunable by design. The Mahalanobis budget C lets a user dial between finding plausible edge cases close to nominal behavior and exploring more adversarial scenarios that remain admissible under the uncertainty model, making Gremlin adaptable to different V&V objectives, risk tolerances, and sources of uncertainty beyond atmospheric density.

As motivated in the introduction, the V&V challenge Gremlin addresses is not unique to space. Any autonomous robotic system evaluable in simulation under uncertain conditions is a valid target: rovers planning traverses over uncertain terrain, manipulation systems under uncertain contact dynamics, aerial vehicles in unknown wind fields, or underwater robots in unmapped environments. The only requirements are a forward simulator and a probabilistic model of the uncertainty, making it immediately applicable to the high-fidelity black-box simulators commonly used across robotics development.

Much of the current discussion in robotics focuses on making autonomy more capable. An equally important and underaddressed question is how to know, before deployment, where that autonomy will break. Gremlin contributes to that goal by using simulation to actively discover the conditions under which autonomous behavior becomes fragile, a capability that must grow alongside the autonomy it is meant to certify.

LIMITATIONS

The current implementation has three main limitations. First, each node evaluation requires propagating the full mission simulation to completion, making large search budgets computationally expensive; surrogate modeling is a promising direction to address this. Second, the disturbance is parameterized at five fixed decision events; finer resolution would allow more expressive profiles at the cost of a deeper search tree. Third, the relay model uses a fixed data rate, a simplification that does not capture how link quality varies with elevation angle during descent; a higher-fidelity communication model is a direction for future work. Extending Gremlin to multiple simultaneous uncertainty sources and non-Gaussian uncertainty models are further open directions.

ACKNOWLEDGMENT

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The first author was supported by the Swiss Federal Institute of Technology in Lausanne (EPFL). We thank Dr. Saptarshi Bandyopadhyay for his support with the MuSCAT simulation framework, and Dr. Reza Karimi for providing the reference trajectory data and SPICE kernels for the Uranus Orbiter and Probe mission concept.

REFERENCES

[1] M. Ono *et al.*, "To boldly go where no robots have gone before – Part 1: EELS robot to spearhead a new one-shot exploration paradigm with in-situ adaptation," in *AIAA SciTech Forum*, 2024.

[2] T. Vaquero *et al.*, "EELS: Autonomous snake-like robot with task and motion planning capabilities for ice world exploration," *Science Robotics*, vol. 9, 2024.

[3] M. Ono, D. Selva, M. L. Cable, M. Ethvignot *et al.*, "Planetary Exploration 3.0: A Roadmap for Software-Defined, Radically Adaptive Space Systems," arXiv:2604.20910, 2026.

[4] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," arXiv:1902.01909, 2019.

[5] M. Świechowski, K. Godlewski, B. Sawicki, and J. Mańdziuk, "Monte Carlo Tree Search: A review of recent modifications and applications," *Artif. Intell. Rev.*, vol. 56, no. 3, pp. 2497–2562, 2023.

[6] S. Bandyopadhyay, Y. K. Nakka, L. Fesq, and S. Ardito, "Design and development of MuSCAT: Multi-Spacecraft Concept and Autonomy Tool," in *AIAA AVIATION Forum and ASCEND*, 2024.

[7] A. Simon, F. Nimmo, and R. C. Anderson, *Uranus Orbiter and Probe Mission Concept Study*. NASA, 2023.

[8] G. F. Lindal *et al.*, "The atmosphere of Uranus: Results of radio occultation measurements with Voyager 2," *J. Geophys. Res.*, vol. 92, no. A13, pp. 14987–15001, 1987.