2026 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
MODELING, SIMULATION, PROTOTYPING, AND VALIDATION (MSPV) TECHNICAL SESSION
AUG. 11-13, 2026 - NOVI, MICHIGAN

PHYSICS-CONSTRAINED PROCEDURAL GENERATION OF SYNTHETIC ENVIRONMENTS FOR THE ROBUST VALIDATION OF AUTONOMOUS MANEUVER AND LOGISTICS

Bulent Soykan [1], Ghaith Rabadi [2], Grace Bochenek[1], and Victor J. Paul[3]

[1]University of Toledo, Toledo, OH
[2]University of Central Florida, Orlando, FL
[3]US Army DEVCOM GVSC, Warren, MI

ABSTRACT

The validation of Autonomous Ground Vehicles (AGVs) and intelligent logistics planners is frequently compromised by the "Sim-to-Real" gap, where simulation environments fail to replicate the physical friction of operational deployment. Ideally, valid test cases must enforce strict mobility constraints and impose realistic sustainment penalties; however, many current generation tools rely on idealized terrain interactions and infinite-resource assumptions. We present a real-time procedural framework designed to generate high-friction validation environments that stress-test the robustness of the System Under Test (SUT). The architecture integrates gradient-based terrain analysis with a stochastic contested logistics model. It ingests synthetic heightmaps to precompute mobility corridors, ensuring that every generated evaluation episode adheres to vehicle-specific traversability limits. Simultaneously, a logistics kernel enforces fuel consumption scaled by terrain gradients and models supply chain interdiction as a parameterized Bernoulli process. We validate this framework through a "Digital Twin" methodology, demonstrating that terrain-aware generation eliminates invalid initialization states (0% mobility violations) while the logistics model induces operationally relevant failure modes in the SUT. This unclassified, open-architecture approach supports DoD Verification, Validation, and Accreditation (VV&A) requirements by providing deterministic, reproducible edge cases for autonomous system evaluation.

Citation: Bulent Soykan, Ghaith Rabadi, Grace Bochenek, and Victor J. Paul "Physics-Constrained Procedural Generation of Synthetic Environments for the Robust Validation of Autonomous Maneuver and Logistics," In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 11-13, 2026.

1. INTRODUCTION

The deployment of autonomous systems in contested environments requires a validation strategy that goes beyond "happy path" testing [1, 2]. While autonomous navigation algorithms often perform well in idealized simulations, their performance degrades rapidly when subjected to the physical friction of real-world topography and the stochastic na-

DISTRIBUTION A. Approved for public release; distribution is unlimited. OPSEC #:10808

ture of sustainment operations [3]. To bridge this Sim-to-Real gap, validation environments must do more than visualize a scenario; they must enforce physical and logistical constraints that rigorously bound the Operational Design Domain (ODD) of the System Under Test (SUT) [4].

Current simulation-based testing often suffers from a lack of environmental fidelity regarding ”logistics friction” and ”terrain adherence [5].” In many legacy testbeds, the environment allows the SUT to traverse steep gradients without energy penalty and treats resupply as a guaranteed, instantaneous event. Consequently, autonomous planners trained or validated in these environments develop brittle policies, optimizing for shortest-path solutions that are physically impassable or logistically unsustainable in the field.

Also, standard validation workflows often rely on static, hand-scripted scenarios. These limited datasets fail to expose the SUT to the ”long tail” of edge cases required to certify robust autonomy. There is a critical need for procedural generation frameworks that can automatically produce high-entropy, physics-constrained test vectors that force the SUT to account for slope-dependent fuel burn and probabilistic supply chain failure [6].

The objective of this research is to develop an automated pipeline that procedurally generates rigorous evaluation episodes where terrain strictly dictates mobility and threat levels dictate sustainment. The proposed framework shifts the focus from ”scenario visualization” to ”constraint generation.”

The system achieves this through two core mechanisms:

- *Terrain-Aware Constraint Generation:* Utilizing gradient-based heightmap analysis to strictly enforce mobility limits. The system pre-computes slope and elevation data to classify terrain into `NO_GO`, `KEY_TERRAIN`, and `MOBILITY_CORRIDOR` zones, ensuring 100% validity in SUT initialization and waypoint generation.

- *Contested Logistics Stress-Testing:* Implementing a logic kernel that tracks per-agent fuel and energy states. Consumption is scaled by terrain friction angles, and resupply events are subject to probabilistic interdiction. This forces the SUT’s high-level planner to solve for survivability and energy conservation, not just geometric path length.

To facilitate broad dissemination within the autonomy verification community and enable transparent Verification, Validation, and Accreditation (VV&A), the proposed framework is architected as an unclassified system.

The research presented here relies exclusively on synthetic terrain data generated via fractal Perlin noise algorithms. The use of synthetic heightmaps allows for the rigorous validation of slope analysis and placement algorithms without the use of classified Geospatial Intelligence (GEOINT) or Digital Terrain Elevation Data (DTED). By validating the logic kernel against synthetic data with known mathematical properties, we demonstrate that the test harness will perform correctly when evaluating the SUT against classified real-world data in a secure environment.

The remainder of this paper is organized as follows: Section 2 reviews related work in procedural content generation and autonomous simulation; Section 3 details the system architecture and data contracts; Sections 4 and 5 describe the algorithmic foundations of the Terrain Intelligence and Contested Logistics modules; Section 6 presents the VV&A results derived from our digital twin experiments; and Section 7 concludes with a summary of contributions and future research directions.

2. RELATED WORK

The development of the proposed framework builds upon three distinct bodies of research: Procedural Content Generation (PCG) for autonomous system validation, geomorphometric terrain analysis for mobility estimation, and discrete event simulation for logistics constraints. While substantial work exists in each domain, the integration of strict terrain-mobility coupling with stochastic logistics friction remains a distinct challenge for the verification of robust autonomous planners.

2.1. Procedural Content Generation for Validation

PCG is increasingly adopted in the autonomous vehicle industry to generate massive datasets for training and testing machine learning models. While Smelik et al. demonstrated that declarative modeling could generate consistent scenarios for human-in-the-loop applications [7], the validation of autonomous systems requires a shift from visual plausibility to physical validity.

In autonomy validation, the primary goal of PCG is to identify the "boundary cases" of the SUT's capabilities. Unlike commercial game engines that utilize simplified navigation meshes (NavMesh) for non-player character pathfinding, the proposed framework utilizes gradient-based analysis derived from raw heightmap data. This allows for the generation of edge-case topographies that specifically stress-test the SUT's traction control and path-planning subsystems, ensuring the agent is validated against the physical limits of the platform rather than the simplifications of the rendering engine.

2.2. Terrain Analysis and Mobility Prediction

The extraction of mobility costs from Digital Elevation Models (DEMs) is a foundational element of automated route planning. Horn's algorithm for hill shading and slope calculation [8] remains a standard for determining surface gradients. The framework implements a central finite-difference approximation similar to Horn's method to compute slope angles θ in real-time, serving as the "Ground Truth" oracle for the simulation.

Simulation-based validation often suffers from the "Sim-to-Real" gap, where simulated soil interaction does not match physical reality. Integration of soil types and slope gradients is necessary in order to accurately model off-road mobility [9]. We extend this by coupling the slope gradient θ directly to the SUT's energy cost function. This ensures that the simulation environment enforces a physics-based penalty for aggressive maneuvering, aligning with DoD VV&A guidelines for physics-based fidelity [10].

2.3. Logistics and Energy-Constrained Planning

Standard validation pipelines often focus on obstacle avoidance and geometric path planning, frequently abstracting away sustainment constraints. Traditional discrete event simulations model logistics at a strategic level [11], but rarely interface directly with the tactical mobility planner of an individual autonomous agent.

The validation of long-duration autonomy requires "energy-aware" planning logic. Recent research emphasizes that sustainment in contested environments is stochastic. The proposed framework implements a probabilistic interdiction model to inject uncertainty into the SUT's state estimation. By treating resupply events as Bernoulli trials subject to threat-based failure, the framework forces the SUT to solve for survivability and energy conservation under uncertainty, bridging the gap between kinematic simulation and mission-level endurance testing.

2.4. Synthetic Environments and Domain Randomization

To ensure the unclassified distribution and open verification of these validation algorithms, the framework utilizes synthetic terrain generation based on gradient noise functions [12]. In the context of autonomous learning, this approach supports "Domain Randomization," preventing the SUT from overfitting to specific geospatial tiles (e.g., DTED/SRTM). By validating the logic kernel against synthetic data with known mathematical properties, we provide a deterministic test harness that allows Test Engineers to isolate algorithmic failures in the SUT from artifacts in the map data [13, 14].

3. SYSTEM ARCHITECTURE

The proposed framework is architected as a modular, service-oriented test harness designed to decouple the generation of environmental constraints ("Ground Truth") from the decision-making logic of the SUT. This separation of concerns supports a Modular Open Systems Approach (MOSA), ensuring that the validation environment remains agnostic to the specific autonomous stack or visualization engine being employed [15]. The architecture consists of four distinct stages communicating via lightweight messaging protocols.

3.1. The Validation Pipeline

The pipeline functions as a deterministic state machine that transforms abstract stochastic variables into concrete, physics-constrained simulation states.

Stochastic Event Injector (ZMQ)

The entry point of the validation loop is the Event Injector. Utilizing the ZeroMQ (ZMQ) Publish-Subscribe pattern, this module acts as a "chaos monkey" for the simulation, broadcasting probabilistic state perturbations. Rather than following a fixed script, the injector emits `ENVIRONMENT_UPDATE` and `THREAT_VECTOR` messages based on configurable entropy parameters. This forces the SUT to operate in a non-deterministic environment, testing its ability to adapt to dynamic exogenous factors (e.g., sudden weather degradation or pop-up threats) rather than memorizing a static sequence of events.

Environment Logic Kernel

The core of the testbed is the Environment Logic Kernel. This component enforces the physical and logical consistency of the simulation. It processes incoming event streams through three validation sub-modules:

- *Constraint Manager:* Maps abstract threat levels to concrete friction coefficients (e.g., enabling stochastic fuel leaks or delay penalties).
- *Terrain Analysis Engine:* Queries the static heightmap to strictly validate SUT initialization points against slope and elevation limits, ensuring no test case begins in an invalid state.
- *Logistics State Machine:* Maintains the "Ground Truth" for every entity, calculating physics-based fuel consumption and adjudicating the success/failure of sustainment actions requested by the SUT.

The output is a rigorously validated `ScenarioManifest`, representing the indisputable state of the world at time t.

State Transmission Layer (UDP)

To support real-time Hardware-in-the-Loop (HWIL) or Software-in-the-Loop (SIL) testing, the framework utilizes a low-latency

UDP Bridge. The `ScenarioManifest` is serialized into a compact JSON payload and transmitted on port 3000. UDP is selected to minimize transport latency, ensuring that the SUT receives state updates at a frequency matching its internal control loop (typically 10-60 Hz), mimicking a real-time tactical data link.

Visualization and Sensor Emulation (Unreal Engine)

The visualization layer serves as the interface between the Logic Kernel and the SUT's perception systems. Implemented as a C++ plugin, it utilizes two primary subsystems:

- *Receiver Subsystem:* Listens for state updates on a non-blocking thread, ensuring frame-independent data ingestion.
- *Actor Management Subsystem:* Interpolates entity positions and updates environmental states (e.g., sun angle, fog density). This layer provides the visual or sensor-based "stimuli" that the SUT perceives, completing the simulation loop.

3.2. Interface Control and Data Contracts

To ensure the integrity of the validation process, the framework enforces strict data contracts using Pydantic schemas. These schemas function as the Interface Control Document (ICD) between the test generation framework and the SUT.

Type Safety and Validation

The data exchange is governed by the `ScenarioManifest` schema. Critical subschemas include:

- `EntityState`: Defines the kinematic vector (position, rotation) and the internal logistic state (`fuel`, `ammo`) of the SUT. The `fuel` parameter (float, 0.0–100.0) serves as a critical boundary condition for energy-aware planning algorithms.
- `OperationalStatus`: An enumeration (`NOMINAL`, `CRITICAL`, `RESUPPLYING`) that provides semantic grounding for the SUT's self-diagnostic logic.
- `EnvironmentState`: Captures global variables such as time-of-day and visibility, which directly impact the SUT's sensor performance models.

Input Sanitization for VV&A

By enforcing these contracts within the Logic Kernel, the system guarantees that the simulation engine never instantiates an invalid test case. If the procedural engine attempts to generate an entity with negative fuel or out-of-bounds coordinates, the schema validator intercepts the error before transmission. This guarantees that any failure observed during testing can be attributed to the SUT's logic rather than a defect in the simulation environment, a critical requirement for formal system accreditation.

4. TERRAIN INTELLIGENCE AND CONSTRAINT GENERATION

The Terrain Intelligence Module serves as the spatial logic kernel of the framework, transforming raw heightmap data into semantic mobility cost manifolds. In the context of autonomous system validation, this module is responsible for defining the geometric boundaries of the ODD. By analyzing the physical properties of the terrain, the system replaces arbitrary waypoint generation with a deterministic, constraints-based approach that guarantees that every generated test case is kinematically feasible for the platform.

4.1. Gradient-Based ODD Definition

The input to the system is a grayscale heightmap image where pixel intensity $I(x,y) \in [0,255]$ maps linearly to world elevation $E(x,y)$. To validate the SUT's traction control and path-planning subsystems, the validation framework must derive the surface slope angle θ at every point, establishing the "Ground Truth" for mobility costs.

The framework employs a central finite-difference approximation to compute the gradient vector ∇E of the elevation field. For a grid cell at coordinates (x,y), the partial derivatives are approximated as:

$$\frac{\partial E}{\partial x} \approx \frac{E(x+1,y) - E(x-1,y)}{2\Delta s} \quad (1)$$

$$\frac{\partial E}{\partial y} \approx \frac{E(x,y+1) - E(x,y-1)}{2\Delta s} \quad (2)$$

where Δs represents the grid spacing. The slope angle $\theta(x,y)$ is derived from the gradient magnitude:

$$\theta(x,y) = \arctan\left(\sqrt{\left(\frac{\partial E}{\partial x}\right)^2 + \left(\frac{\partial E}{\partial y}\right)^2}\right) \times \frac{180}{\pi} \quad (3)$$

This computation is vectorized to process high-resolution synthetic environments in approximately 200 milliseconds. The resulting slope field serves as the oracle for determining whether a specific coordinate falls within the SUT's traversability limits.

4.2. Semantic Mobility Classification

Once the slope field is computed, the system segments the environment into discrete mobility classes. This logic automates the identification of "edge cases" (boundary conditions) for the autonomous planner. The system defines three mutually exclusive zones:

- *NO_GO (Hard Constraint):* Defined as terrain where $\theta(x,y) > 30^\circ$. This threshold represents the critical traction limit for the specific vehicle class under test (e.g., standard tracked gradeability). These zones are strictly prohibited for SUT initialization, preventing the generation of "False Positive" failures where the SUT fails simply because it was spawned in an impossible state.
- *KEY_TERRAIN (Vantage Points):* Defined as terrain that is both traversable ($\theta \leq 30^\circ$) and situated in the top 20th percentile of elevation ($E(x,y) \geq P_{80}$). These zones are used to generate navigation goals that stress-test the SUT's ability to maximize sensor coverage or reach high-cost objectives.
- *MOBILITY_CORRIDOR (Admissible State Space):* Defined as all remaining traversable terrain. These areas represent the valid configuration space ($\mathscr{C}_{free}$) for standard maneuver.

These thresholds are exposed as configurable parameters in the experiment configuration, allowing Test Engineers to tighten or relax the ODD constraints based on the specific platform being validated.

4.3. Valid-by-Construction Initialization

Standard Monte Carlo testing often utilizes uniform random sampling across map bounds, resulting in invalid test vectors when applied to complex topography. This "guess-and-check" method creates noise in the validation results.

The proposed framework replaces this with *rejection-free sampling*. During initialization, the analysis engine constructs Boolean masks for each mobility class. The valid initialization set S_{init} for the SUT is defined as:

$$S_{init} = \{(x,y) \mid \theta(x,y) \leq \theta_{max}\} \quad (4)$$

To spawn the SUT or generate navigation waypoints, the system samples uniformly from S_{init} rather than the global map. This guarantees that every generated evaluation episode begins in a valid state (0% mobility violations), ensuring that any subsequent immobilization observed during the simulation is attributable to the SUT's decision-making logic rather than an artifact of the test generation process.

5. LOGISTICS CONSTRAINT MODELING AND STRESS-TESTING

To validate the robustness of autonomous sustainment logic, the framework integrates a discrete-event logistics kernel directly into the real-time simulation loop. This module serves as a rigorous "adversarial environment," enforcing two critical physical constraints: terrain-coupled energy expenditure and stochastic supply chain failure. These constraints force the SUT to optimize for survivability and endurance, exposing brittleness in planners that solve only for geometric shortest paths.

5.1. Terrain-Coupled Energy Expenditure

Standard validation environments often model energy consumption as a linear function of distance ($E \propto d$). This simplification fails to capture the "Sim-to-Real" gap regarding the mechanical effort required to traverse complex topography. The proposed framework implements a physics-based consumption model within the `LogisticsManager`, where the energy burn rate is dynamically scaled by the terrain gradient derived from the Terrain Intelligence Module.

The instantaneous energy consumption C_e (in units per tick) is calculated as:

$$C_e = R_{base} \times v \times \left(1 + \frac{\theta}{30}\right) \quad (5)$$

Where:

- R_{base} is the nominal consumption rate for the platform on flat ground.
- v is the scalar velocity of the SUT.
- θ is the slope of the terrain in degrees at the SUT's current position.
- The denominator 30 represents the reference slope angle (30°) at which mechanical effort doubles compared to flat terrain.

This formula couples the mobility cost directly to the environment. An autonomous agent traversing a 30° incline will deplete its energy reserves at twice the rate of an agent moving on flat ground. Consequently, this forces the SUT's path planner to treat the terrain gradient as a high-cost variable in its optimization function (J). Failure to account for this non-linear cost results in "energy starvation" before reaching the objective, a critical failure mode for long-duration autonomy.

5.2. Stochastic Service Availability

In contested environments, the availability of resupply (refueling/recharging) is not guaranteed. The framework models this uncertainty using a Bernoulli trial approach, triggered whenever the SUT requests a logistics service (e.g., automated refueling rendezvous).

When the validation scenario dictates a `HIGH_THREAT` condition, every resupply request is subject to a probabilistic outcome governed by a tunable interdiction parameter. The outcome space Ω for a service request R is defined as:

$$P(R) = \begin{cases} 0.60 & \text{Success (Imm. replenishment)} \\ 0.20 & \text{Denial (Total resource loss)} \\ 0.20 & \text{Latency (Arrival} > t + \delta) \end{cases} \tag{6}$$

This stochastic model tests the SUT's "Risk-Aware Planning." A robust autonomous system must anticipate potential service failures by maintaining safety buffers (e.g., requesting resupply at 30% capacity rather than 5%). A "Service Denial" result forces the SUT to execute contingency logic, while "Service Latency" tests the agent's ability to maintain station-keeping or enter a low-power state while awaiting support.

5.3. Dynamic Constraint Injection

The transition between permissive and non-permissive validation states is managed by the `TriggerManager`. This component acts as a deterministic state machine that maps abstract environmental priors to concrete simulation constraints.

The manager ingests stochastic events (e.g., `THREAT_LEVEL_INCREASE`) and evaluates them against rule sets to emit state changes. For example, an injected event indicating "Degraded Infrastructure" emits a `LOGISTICS_FRICTION` command. This command reconfigures the `LogisticsManager` to engage the stochastic interdiction probabilities described above. This architecture allows the Test Engineer to dynamically scale the difficulty of the evaluation episode, verifying that the SUT can detect and adapt to changing environmental friction in real-time.

6. VERIFICATION, VALIDATION, AND ACCREDITATION (VV&A)

Consistent with DoD Instruction 5000.61 [10], the proposed framework employs a rigorous, layered VV&A strategy. In the context of autonomous system validation, the primary objective of this phase is to establish confidence in the *testbed*. We must demonstrate that the procedural engine generates valid ODDs and that the physical constraints imposed on the SUT are deterministic and reproducible.

6.1. The "Digital Twin" Testbed Methodology

A primary challenge in autonomy verification is distinguishing between failures caused by SUT logic errors and those caused by defects in the simulation environment (e.g., collision mesh artifacts or invalid waypoints). To isolate these domains, the framework utilizes a "Digital Twin" methodology.

We developed a standalone Python-based validation kernel (`viz_mock_kernel`) that executes the exact logic modules used in the production pipeline but renders to a lightweight 2D surface. This headless environment allows Test Engineers to validate:

- *Kinematic Integration:* Verifying that velocity vectors and boundary constraints behave deterministically before integrating complex vehicle dynamics.
- *Sensor Masking:* Visualizing strict Line-of-Sight (LOS) filtering logic against ground truth data to ensure the SUT receives accurate partial observability.
- *State Serialization:* Confirming that the `ScenarioManifest` JSON payload is structurally valid prior to transmission over the UDP bridge.

By validating the evaluation episode in this headless environment first, we ensure that the test vectors are structurally perfect, reducing the "debugging the debugger" phenomenon during HWIL/SIL integration.

6.2. Verification (Code Correctness)

Verification establishes that the simulation logic is built correctly according to specification. The codebase includes an automated test suite comprising 31 deterministic unit tests covering critical boundary conditions:

- *Terrain Logic Verification (10 tests):* Validates that the analysis engine correctly identifies edge cases, such as identifying a single pixel of 31° slope as `NO_GO` within a field of traversable terrain.
- *Logistics State Verification (21 tests):* Validates state transitions within the `LogisticsManager`. Specifically, tests confirm that energy consumption functions are strictly non-negative, ensuring the SUT cannot "regenerate" fuel by driving backwards or downhill, a common exploit in poorly constrained simulations.

6.3. Validation (Operational Utility)

Validation establishes that the generated environments are useful for stress-testing autonomous agents. We conducted two controlled experiments to quantify the utility of the constraint generation modules using a baseline heuristic agent.

Experiment A: ODD Definition Viability

To quantify the "Sim-to-Real" gap in test case generation, we compared the validity of SUT initialization points generated via Uniform Random Sampling versus the proposed Terrain-Aware method. The experiment attempted to spawn $n = 2000$ agents (1000 per method) on a high-relief synthetic heightmap. A "Validity Failure" was defined as any agent spawned on a slope exceeding the 30° `NO_GO` threshold.

Table 1: Experiment A: Initialization Validity Results

Generation Method	Sample Size	Invalid States	Failure Rate
Random Uniform	1,000	718	71.8%
Terrain-Aware	1,000	0	**0.0%**

The results (Table 1) demonstrate that unconstrained placement fails over 70% of the time in complex terrain. This implies that 7 out of 10 evaluation episodes generated by legacy random methods would yield "False Positive" failures (agent immobilized immediately). The proposed framework's rejection-free sampling achieved a 100% validity rate, verifying its ability to generate "Valid-by-Construction" test cases.

Experiment B: Logistics Stress-Testing

To validate the framework's ability to impose robust constraints, we simulated 100 autonomous supply runs under two environmental conditions: Control (Permissive, 0% interdiction) and Contested (High Friction, 40% interdiction). We measured Mission Completion Rate and Mean Time To Failure (MTTF).

Table 2: Experiment B: Environmental Stress Results

Condition	Completion	Energy Starvations	MTTF (ticks)
Control	100%	0	N/A
Contested	82%	18	104.4

Under contested conditions (Table 2), the environment successfully induced operational failure modes. The 18% attrition rate due to energy starvation confirms that the stochastic interdiction model forces the SUT to contend with non-trivial sustainment pressure. This proves the testbed is capable of distinguishing between robust planners (which would reserve safety buffers) and brittle planners (which fail when resupply is delayed).

6.4. Sensitivity Analysis and Parameter Space

To prove the stability of the test generation process, we performed parameter sweeps across three key variables to ensure the system behaves monotonically.

- *Slope Threshold (15°–45°):* As the `NO_GO` threshold increased, the available traversable configuration space ($\mathscr{C}_{free}$) increased non-linearly, accurately reflecting the specific geomorphology of the validation terrain.
- *Elevation Percentile (50^{th}–95^{th}):* Sweeping the definition of `KEY_TERRAIN` demonstrated a predictable scarcity curve. At the 95th percentile, only 1.4% of the map was available for goal generation, validating the system's ability to generate "needle in a haystack" navigation problems.
- *Interdiction Rate (0%–60%):* SUT mission completion rates degraded monotonically from 100% to 76% as interdiction increased. Notably, the steepest degradation occurred between 0% and 10% (dropping to 92%), highlighting that even minimal friction levels are sufficient to expose brittleness in non-robust planners.

This analysis confirms that the proposed framework provides a stable, tunable validation environment suitable for certifying autonomous ground systems.

7. RESULTS AND DISCUSSION

The proposed procedural framework was evaluated against two primary validation criteria. First, we measured its ability to eliminate invalid SUT initialization states on high-relief terrain, ensuring strict ODD compliance. Second, we assessed its capacity to induce measurable, verifiable friction through stochastic logistics constraints. All validation experiments utilized the synthetic heightmap described in Section 4 to ensure deterministic reproducibility.

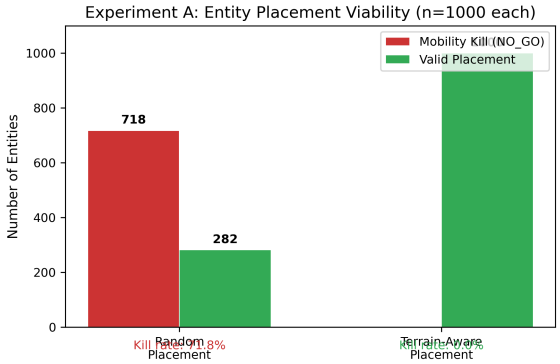

Figure 1: Comparison of ODD initialization viability between random sampling and terrain-aware constraint generation ($n = 2000$).

7.1. ODD Initialization Viability

In Experiment A, we generated $n = 1000$ SUT initialization points using a uniform random distribution. We then generated an additional $n = 1000$ points using the proposed terrain-aware constraint engine. A "Validity Failure" was defined as any agent spawned on a slope exceeding the 30° `NO_GO` threshold, representing an immediate kinematic failure outside the platform's ODD.

As shown in Figure 1, uniform random placement resulted in a 71.8% validity failure rate (718 immobilized agents). This high failure rate is characteristic of complex, fractal terrain where traversable configuration space ($\mathscr{C}_{free}$) represents a minority of the surface area. In contrast, the terrain-aware constraint engine achieved a 0% failure rate. It successfully identified and utilized only the 27.6% of the map classified as traversable.

The spatial distribution of these test cases is visualized in Figure 2. The random generation method (left) distributes

test vectors indiscriminately across impassable gradients (red zones). This renders the evaluation episodes analytically useless, as the SUT fails before executing any planning logic. The terrain-aware method (right) restricts initialization exclusively to valid MOBILITY_CORRIDOR (green) and KEY_TERRAIN (gold) zones. This approach guarantees that every generated test case provides a physically viable starting state for the SUT.

Sensitivity analysis of the slope threshold confirms this result is robust. Even when the NO_GO threshold was relaxed to 45°—an extreme limit for ground vehicles—random placement still yielded a 25.8% failure rate. This establishes that procedural terrain analysis is a mandatory prerequisite for valid test case generation on any non-trivial topography.

7.2. Logistics Stress-Testing and SUT Endurance

Experiment B evaluated the framework's ability to stress-test the SUT's energy-aware planning logic. We simulated 100 autonomous supply routes under Permissive conditions (0% interdiction) and 100 under Contested conditions (40% interdiction probability). We utilized a baseline heuristic planner as the SUT.

Under Permissive conditions, the SUT achieved a 100% mission completion rate. In the Contested environment, the completion rate dropped to 82%, with 18 autonomous agents suffering catastrophic energy starvation. As illustrated in Figure 3, the mean fuel remaining upon arrival dropped from 55.4% in the Permissive group to 53.2% in the Contested group.

The degradation in SUT performance followed a monotonic curve relative to the environmental friction parameter. Table 3 presents the sensitivity analysis of the service interdiction rate.

Table 3: Sensitivity Analysis: Interdiction Rate vs. SUT Mission Outcomes

Interdiction Rate	Completion Rate	SUT Failures	Mean Fuel on Arrival
0% (Permissive)	100%	0	55.4%
10%	92%	8	55.7%
20%	90%	10	55.1%
30%	84%	16	53.6%
40% (Default)	80%	20	50.7%
50%	79%	21	50.7%
60%	76%	24	45.4%

The data reveals the fragility of basic path-planning algorithms under stochastic constraints. The introduction of merely a 10% interdiction rate caused an immediate 8% drop in mission completion. This highlights the vulnerability of autonomous agents that optimize solely for geometric distance without reserving energy buffers. SUTs that survived high-friction environments (60% interdiction) arrived with significantly depleted energy reserves (45.4%). This indicates they endured multiple "Service Latency" penalties before successfully replenishing.

Figure 4 breaks down the service request outcomes. In the Contested condition, 20% of service requests resulted in absolute denial, and 18% resulted in forced delays. This confirms the environment accurately enforces the intended probabilistic boundary conditions, forcing the SUT to execute contingency replanning.

7.3. Testbed Performance and Latency

To provide value in HWIL or SIL validation, the testbed must operate within strict frame budgets. Latency introduces artificial noise into the SUT's control loop.

Initialization Overhead: The gradient-based slope analysis and semantic classification of a 1000×1000 heightmap (10^6 data points) requires approximately 200 ms on a

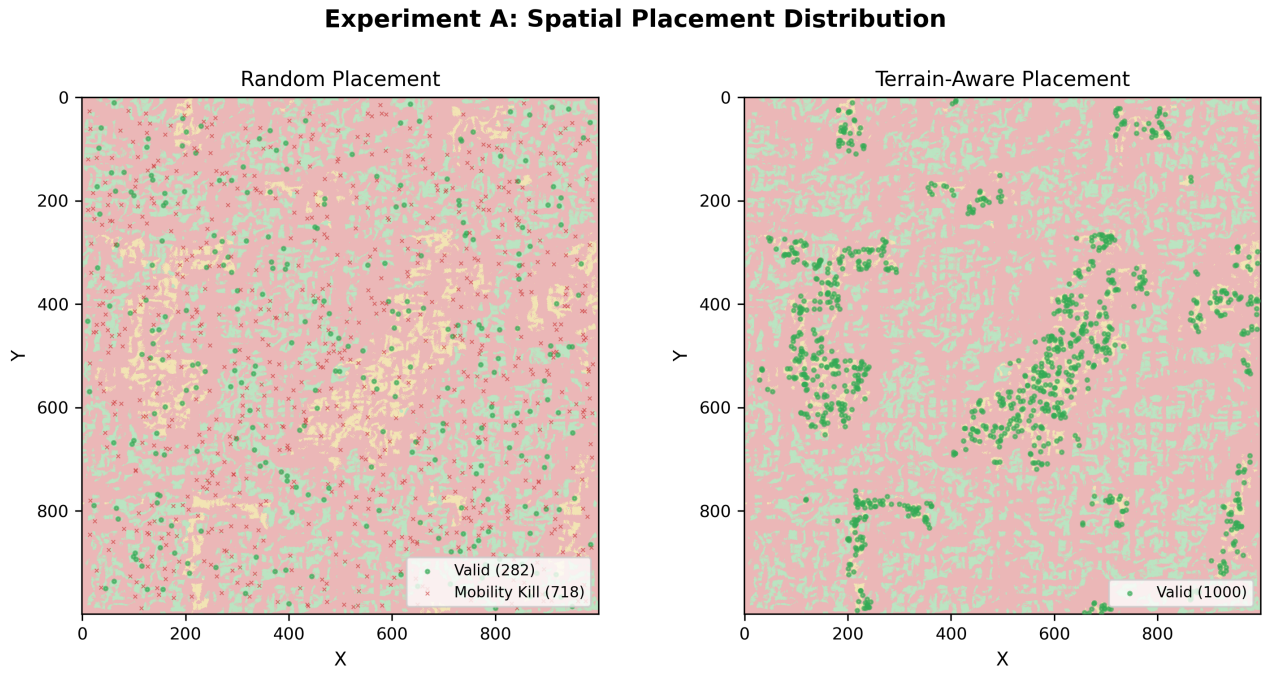

Figure 2: Spatial distribution of generated initialization points. Left: Random placement creates a high volume of invalid test cases in NO_GO zones (red). Right: Terrain-aware placement restricts the SUT strictly to valid corridors, ensuring physical feasibility.

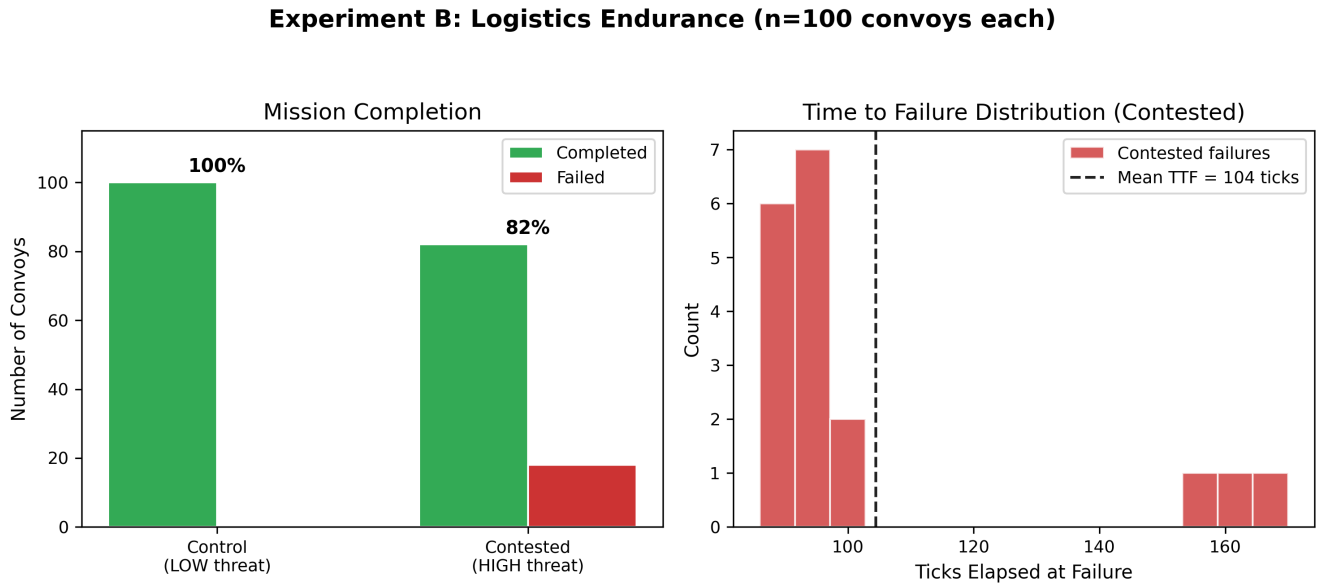

Figure 3: Logistics endurance stress-test results. The contested environment forced an 18% attrition rate due to energy starvation, with a Mean Time To Failure (MTTF) of 104 ticks.

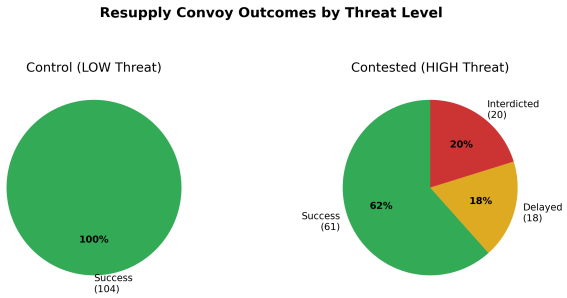

Figure 4: Outcome distribution for SUT service requests. The Contested condition successfully imposes the probabilistic boundary parameters: ∼20% absolute denial and ∼20% latency.

standard workstation CPU. This is a one-time, pre-computation cost incurred before the evaluation episode begins. Subsequent point queries for SUT elevation and bulk sampling for waypoint generation execute in microseconds ($O(1)$).

State Transmission Latency: The UDP Bridge imposes negligible transmission overhead. Serialization of a 100-entity `ScenarioManifest` into a valid JSON payload requires < 1 ms. Network transport latency across a local loopback interface is < 1 ms. This low-latency architecture easily supports update frequencies of 60 Hz or higher. It ensures that the simulation environment remains tightly synchronized with the SUT's perception and control frequencies, satisfying the strict timing requirements of autonomous systems validation.

8. CONCLUSION AND FUTURE WORK

We presented a modular, procedural generation framework that successfully bridges the gap between stochastic environmental parameters and physics-constrained evaluation episodes for autonomous ground vehicles. By integrating gradient-based terrain analysis with a stochastic logistics kernel, the architecture replaces static, hand-scripted test cases with dynamic, valid-by-construction boundary constraints.

Our validation experiments demonstrate that the framework eliminates ODD initialization errors common in uniform Monte Carlo generation, reducing physical validity failures from 71.8% to 0% on high-relief terrain. Furthermore, the logistics stress-testing module introduces operationally relevant friction, forcing the SUT to contend with non-deterministic supply chain degradation (successfully inducing an 18% failure rate in baseline heuristic planners). The architecture, built on open standards and validated through a headless "Digital Twin" methodology, provides a rigorous, unclassified testbed capable of automatically generating the edge cases required for formal DoD VV&A of autonomous systems.

Future development will focus on increasing the environmental fidelity of the testbed and closing the simulation control loop for HWIL and SIL applications.

Real-World Geospatial Ingestion: While the current iteration utilizes synthetic Perlin noise to validate constraint logic without exposing classified geospatial data, the underlying analysis engine is architected to support standard formats. Future work will integrate GDAL bindings to ingest DTED and Shuttle Radar Topography Mission (SRTM) datasets. This will enable geo-specific ODD validation, allowing Test Engineers to certify autonomous platforms against the exact topography of targeted deployment zones.

Closed-Loop State Synchronization: Currently, the pipeline operates primarily in a feed-forward manner (Constraint Engine $\rightarrow$ Sensor Emulation/SUT). We plan to fully implement the telemetry return channel (UDP port 3001) to enable bidirectional state synchronization. This will allow the high-fidelity physics engine (e.g., Unreal Engine's Chaos physics) to report dynamic mobility events—such as terrain-induced wheel slip, chassis collisions, or suspension faults—back

to the logic kernel, updating the SUT's energy and mobility state in real-time.

High-Fidelity Energy Models: We intend to expand the logistics kernel to support platform-specific powertrain profiles (e.g., hybrid-electric discharge curves vs. internal combustion efficiency, wheeled vs. tracked mechanical friction). This will further narrow the Sim-to-Real gap by forcing the SUT to optimize multi-objective routing problems based on highly specific, non-linear energy consumption dynamics.

References

[1] Anthony Corso, Robert Moss, Mark Koren, Ritchie Lee, and Mykel Kochenderfer. A survey of algorithms for black-box safety validation of cyber-physical systems. *Journal of Artificial Intelligence Research*, 72:377–428, 2021.

[2] Baiming Chen, Xiang Chen, Qiong Wu, and Liang Li. Adversarial evaluation of autonomous vehicles in lane-change scenarios. *IEEE transactions on intelligent transportation systems*, 23(8):10333–10342, 2021.

[3] Alexander Kott and William M McEneaney. *Adversarial reasoning: computational approaches to reading the opponent's mind*. Chapman and Hall/CRC, 2006.

[4] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *European conference on artificial life*, pages 704–720. Springer, 1995.

[5] Karl Iagnemma and Steven Dubowsky. *Mobile robots in rough terrain: Estimation, motion planning, and control with application to planetary rovers*, volume 12. Springer Science & Business Media, 2004.

[6] Mark Koren, Saud Alsaif, Ritchie Lee, and Mykel J Kochenderfer. Adaptive stress testing for autonomous vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7. IEEE, 2018.

[7] Ruben M Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. A survey on procedural modelling for virtual worlds. In *Computer graphics forum*, volume 33, pages 31–50. Wiley Online Library, 2014.

[8] Berthold KP Horn. Hill shading and the reflectance map. *Proceedings of the IEEE*, 69(1):14–47, 2005.

[9] Mieczyslaw Gregory Bekker. Introduction to terrain-vehicle systems. *(No Title)*, 1969.

[10] U.S. Department of Defense. Dod instruction 5000.61: Dod modeling and simulation verification, validation, and accreditation.

[11] Jerry Banks, John S. Carson, II, Barry L. Nelson, and David M. Nicol. *Discrete-Event System Simulation*. Pearson, 5 edition.

[12] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.

[13] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[14] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12627–12637, 2019.

[15] Mike P Papazoglou and Willem-Jan Van Den Heuvel. Service oriented architectures: approaches, technologies and research issues. *The VLDB journal*, 16(3):389–415, 2007.

CONTACT INFORMATION

Bulent Soykan
Research Professor
Department of Mechanical, Industrial & Manufacturing Engineering
University of Toledo
2801 W BANCROFT ST
Toledo, OH 43606
Bulent.Soykan@utoledo.edu
786-695-3301

9. ACKNOWLEDGMENT

The authors wish to acknowledge the technical and financial support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-24-2-0001 U.S. Army DEVCOM Ground Vehicle Systems Center (GVSC) Warren, MI.