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

INTELLIGENT OPTIMIZATION SIMULATION FRAMEWORK FOR TEST AND EVALUATION OF AUTONOMOUS SYSTEMS

S. Snarski, PhD[1], A. Menozzi, PhD[2], B. Persons[3], D. Lazar[1], N. Khan[4]

[1]MTSI, VA
[2]Novus LLC, NC
[3]Persons Industries, AL
[4]Eight Queens, CA

ABSTRACT

This paper describes ongoing research and development of an efficient optimization/search–based modeling and simulation framework for rapidly identifying low-performance scenarios in advanced autonomous systems. Ensuring predictable, safe behavior across complex, integrated systems remains a core operational test-and-evaluation challenge. Our goal is to balance rigorous validation with timely deployment. We are developing TEAAS (Test & Evaluation of Advanced Autonomous Systems), a scalable, faster-than-real-time framework designed to uncover critical failure scenarios efficiently. Key features include GPU-accelerated parallel simulation and learning, computational intelligence–based search of optimal parameters, uncertainty quantification for reproducibility, and real-time physics-accurate sensor models. We conducted simulation experiments to evaluate and demonstrate the framework performance for two black-box ground-vehicle autonomous systems. Key results were that adequate uncertainty quantification can be achieved with as few as 10 repeated runs per simulation scenario, sensor realism has a significant effect on failure rate, distinct differences between the two autonomies failure modes were identified, and our efficient optimization/search methods identify critical performance regions in a small fraction of the number of simulations required by a naïve Monte Carlo search.

Citation: S. Snarski, A. Menozzi, B. Persons, D. Lazar, N. Khan, "Intelligent Optimization Simulation Framework for Test and Evaluation of Autonomous Systems," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 11-13, 2026.

1. INTRODUCTION

Over the last decade there has been explosive growth in the autonomous systems market, driven by the convergence of sensor advances, compute scalability, artificial intelligence (AI) breakthroughs, economic and political pressure, and enabling infrastructure all maturing at the same time. Autonomous systems interact in the physical world, where

every action changes future observations and states. This creates feedback-driven dynamics where small errors can compound over time. These systems must also operate with noisy sensors in open, unpredictable environments, where objects are partially observable, filled with many unforeseen edge cases. Their algorithms cannot rely on fixed rules alone; they must generalize beyond what was previously experienced. The combination of increasing system, mission, and environmental complexity can also yield non-deterministic, unpredictable, and emergent behaviors. Autonomous systems are also safety critical. Because failures can result in physical harm or major financial loss, autonomous systems face reliability requirements far more demanding than those of most AI or software applications.

The challenge within the autonomous systems community lies in ensuring that the integrated system behaves predictably and safely over time. Exhaustive testing or simulation of all possible scenarios is infeasible, making it difficult to identify failure causes, which are typically multi-factor and interdependent. Operational test and evaluation (OT&E) for autonomous systems is at a crossroads due to the scientific rigor required for adequate testing. Most of the methods commonly used in industry for commercial deployments of autonomous systems (e.g., self-driving cars, drone delivery) fall short of the statistical rigor required. Efforts in academia typically lack operational realism/relevance needed for OT&E. The OT&E community widely recognizes the need for new, high-fidelity, cost-effective simulation and testing approaches that can handle complex state spaces and deliver reliable safety and performance assessments within operationally relevant timeframes.

A review of recent literature shows that *scenario-based testing* methodologies have achieved substantial sophistication, *simulation infrastructure* has advanced significantly, and *uncertainty quantification* is now recognized as essential. Specific findings include:

- **Scenario-based testing** has achieved consensus, but disagreement remains about how scenarios should be generated, selected, and validated. Three competing approaches exist: (a) *data-driven approaches* [1-3] develop scenarios from historical data and statistics, prioritizing realism, but are constrained by data availability; (b) *search-based optimization* [4-6] systematically explores scenario space through algorithms, discovering edge cases but potentially missing realistic distributions; (c) *formal methods* [7-9] provide mathematical rigor and specification precision but face scalability challenges.
- Though simulation capability has advanced rapidly, the **simulation-to-reality gap** is still an issue [10], with the main problem areas being (a) *dynamics fidelity*: system model complexity significantly affects control evaluation [11], (b) *simulator non-determinism*: even deterministic simulators exhibit variation in test outcomes across runs [12], raising questions about reproducibility, (c) *sensor simulation fidelity*: camera, radar, and LiDAR simulation varies widely in realism [13].
- **Uncertainty quantification** (UQ) is critical but computationally expensive. The recognition that autonomous systems must "know what they don't know" (be aware of their own uncertainty and consider it appropriately when making decisions) has driven substantial research on uncertainty quantification [14-16]. Three UQ method families are now well-characterized: (a) *bayesian methods* (e.g., MC-dropout) provide theoretical elegance but are computationally expensive; (b) *ensemble methods* (e.g., deep ensembles) demonstrate promising empirical performance [15], but have historically

been seen as heuristic, lacking formal theoretical guarantees and explainability [17]; (c) *deterministic methods* (e.g., temperature scaling) sacrifice accuracy for efficiency. However, significant gaps remain in UQ methods: real-time deployment constraints, integration with safety-critical decision-making systems, and the distinction between data and model uncertainty in operational contexts.

Insufficient testing creates unacceptable deployment risks and excessive testing delays operational technology deployment indefinitely. Our objective is to achieve an optimal balance by rigorously validating autonomous systems while enabling their deployment at a pace appropriate to stakeholder benefit and risk tolerance. To this end, we are researching and developing a scalable, faster-than-real-time, operationally realistic testing framework named TEAAS (T&E of Advanced Autonomous Systems) to efficiently and effectively identify critical failure scenarios in operations involving complex autonomous systems. Our current research and development efforts are focusing on the following areas:

- Highly parallelized GPU-accelerated simulation and learning: extend NVIDIA Isaac Sim/Lab pipeline [18] from system training to environment generation/adaption for black-box systems.
- Computational intelligence search-based optimization (CIO): biologically inspired optimization algorithms to solve complex real-world problems in very large state spaces.
- Uncertainty quantification and reproducibility: ensemble averaging techniques to manage aleatoric (random) and epistemic (model) uncertainty and provide consistent simulation statistics.
- Simulation realism: GPU-accelerated physics-accurate sensor models.

Initial TEAAS proof of concept and scalability demonstrations are described in Persons [19]. In this paper, we describe efforts to extend the framework to explore critical complexity issues essential for proper evaluation of autonomous systems, including uncertainty quantification, simulation realism, and efficient search schemes. Section 2 outlines the design of the TEAAS framework including architecture, components and engineering considerations. Section 3 discusses simulation-based experimental design including the mission use case, performance metrics, and autonomy under test (AUT). Section 4 presents the results of simulation experiments to examine the effects of proper simulation realism when assessing AUT performance and the efficacy of TEAAS framework for efficiently isolating critical scenarios. Finally, Section 5 discusses our ongoing work into open research questions and OT&E customer challenges. Conclusions are provided in Section 6.

2. FRAMEWORK DESIGN

2.1. Overview

The essential operational concept of TEAAS is efficiently exploring and identifying scenarios in which the AUT is prone to failure, without knowledge or insight into its decision-making process. From the perspective of TEAAS, what matters is its performance relative to the task. The task itself and the corresponding performance requirements are typically defined by a stakeholder whose objective is to verify and validate the AUT. Since TEAAS is being developed to assist this stakeholder, specifically by identifying potential trouble spots and failure modes, the framework must include:

- **Requirements.** TEAAS must implement task and performance requirements *as defined by the stakeholder*. This includes

specifications of scenarios that are of interest and on how to measure and assess the AUT's performance (Sections 3.1, 3.2).

- **AUT interfaces.** TEAAS must have appropriate interfaces for AUT hardware configuration (e.g., platform, sensors, actuators) and support Data Distribution Service (DDS) messaging for black-box autonomy software (Section 3.3).
- **Operation Realism.** Simulation fidelity must be appropriate to ensure that uncertainty, disturbances, and physical interactions influence the system in a realistic way to expose cascading failures, degraded performance trends, and long-horizon risks (Section 2.5).
- **Efficient T&E.** Since the focus of TEAAS is to discover unknown trouble spots and failure modes, it must *efficiently* explore the possible scenarios, which it achieves via CIO algorithms (Section 2.3) and by running GPU-based highly parallel simulations (Section 2.2).
- **Complete assessment.** Based on the requirements, TEAAS must provide a final assessment of the AUT that includes the rationale and methods used, and most importantly, *addresses and reports the* uncertainty in the process and results (Section 2.4).

The TEAAS simulation architecture is shown in Figure 1. At its core, TEAAS is about efficiently solving a difficult *optimization/search problem*. It is searching for a set of simulated test scenarios that leads the AUT to fail to complete its assigned task. From an optimization perspective, TEAAS finds a set $S_c = \{x | \hat{F}_N(x) \geq c\}$, where x is a scenario configuration, $\hat{F}_N(x)$ is the measured failure rate of the AUT after N simulated trials in scenario x, and c is a given threshold. In other words, it answers questions such as: for which scenarios does the AUT have a probability of failure of at least c? Not only does TEAAS answer these questions, but it also provides a complete set of analysis products, including measures of confidence in its answers.

This is a difficult problem because the search space is high-dimensional (x has many components), and the objective function to maximize (the measured failure rate $\hat{F}_N(x)$) may contain multiple maxima, may be discontinuous and/or non-differentiable, is in fact stochastic, and includes simulation modules and black-box functions. The optimization framework that TEAAS uses to tackle this problem is described in the rest of this section.

2.2. NVIDIA Isaac Ecosystem

TEAAS uses the open-source NVIDIA Isaac Lab framework [18] as a pipeline to integrate and run autonomous systems and optimization/search components. Built on NVIDIA Isaac Sim and powered by Omniverse, the pipeline delivers high-fidelity, GPU-accelerated physics through the PhysX engine, along with sensor-realistic environments enabled by advanced rendering via ray tracing, together forming the Isaac ecosystem for autonomous systems experimentation.

Traditionally, Isaac Lab is used to train learning agents in parallel across thousands of simulated environments, leveraging GPUs to accelerate large-scale learning for tasks like navigation, manipulation, and multi-agent coordination (Figure 2). TEAAS leverages this same infrastructure but reverses the conventional paradigm: instead of training an autonomous system to handle a fixed environment configuration, TEAAS holds the AUT constant and systematically manipulates the environment. This environmental manipulation involves controlling static and dynamic simulation variables (e.g., terrain, weather, atmosphere, material properties, and threats) that influence system behavior and decision processes. (These are the dimensions of x in the TEAAS optimization/search problem discussed in Section 2.1).

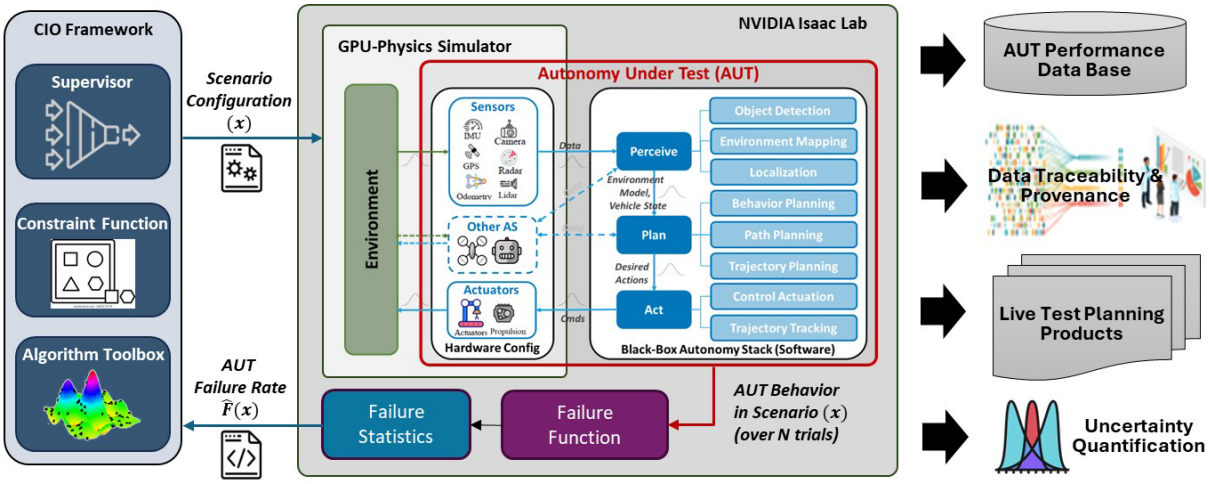

Figure 1: TEAAS intelligent optimization simulation framework architecture.

By varying these environmental parameters, TEAAS generates structured test scenarios that explore both individual and combined effects of conditions on AUT performance. This enables efficient discovery of low-performance and failure cases across a vast combinatorial space while maintaining traceability to specific environmental factors. The result is a scalable, physics-accurate, and computationally efficient approach that strengthens T&E processes by identifying critical edge cases and performance limitations prior to physical testing.

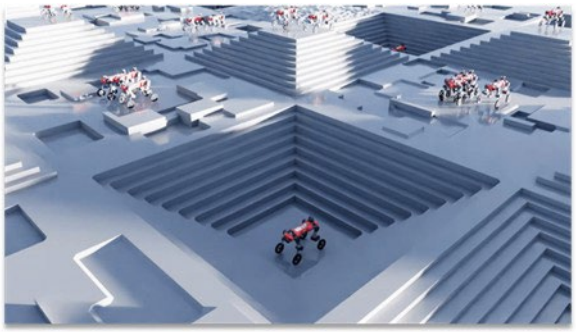

Figure 2: Representative NVIDIA Isaac ecosystem simulation environment showcasing multiple robotic agents navigating complex terrains, illustrating capability for scalable, high-fidelity autonomous systems experimentation and learning.

2.3. Computational Intelligence Optimization (CIO)

Traditional or model-based optimization approaches frequently struggle to offer efficient and effective solutions because real-world situations are typically complex, nonlinear, high-dimensional, and unpredictable. CIO algorithms blend artificial intelligence and optimization techniques inspired by biological and natural processes to efficiently explore large solution spaces and locate nearly optimum solutions, e.g., Zolghadr-Asli [20]. Examples include evolutionary algorithms, swarm intelligence, neural networks and fuzzy systems. These algorithms operate on populations of candidate solutions that share information. Population diversity drives exploration of the problem space, while performance-based selection drives exploitation of promising areas.

Regardless of the CIO algorithm chosen, the algorithm iteratively refines its search using feedback from an objective function that measures AUT performance. In TEAAS, the objective is to maximize the measured failure rate $\hat{F}_N(x)$. What constitutes an individual failure is defined by the failure function, which is developed from mission requirements and system-environment interactions. Its formulation shapes the optimization landscape (see Section 2.4 for further discussion).

In TEAAS, constraints are used to prevent infeasible scenarios. Event-driven, asynchronous population updates remove synchronization bottlenecks, improving scalability and runtime. The system also supports distributed deployment and is dockerized to facilitate scalable, portable, and consistent execution across multiple nodes. Combined with GPU-accelerated simulation, physics models, and learning tools, TEAAS is being developed to efficiently explore vast combinatorial test-scenario spaces.

Since no single algorithm is best for all problems, TEAAS is being developed to support multiple optimization algorithms and use their complimentary capabilities. We are currently evaluating self-adaptive differential evolution (jDE) and particle-swarm optimization (PSO) algorithms, which demonstrate different computational and performance advantages. For instance, jDE is highly effective at finding single-failure regions (unimodal). PSO, on the other hand, can find multiple failure regions (multimodal) but is less efficient.

2.4. Uncertainty Quantification

Random variables are part of any real-life T&E experiment, thus it is only after repeated trials that a reliable picture of results can be assembled. This is also the case in the TEAAS framework. Even though TEAAS is in full control of the simulated environment and can replicate the same scenario as needed (down to generating the same random numbers), it has no control over the AUT and no insight into its decision-making. Since the AUT's decision-making process could very well include some random components, presenting it with the exact same scenario (including the same exact sensor data) could still result in different performance outcomes. TEAAS can also deliver to the AUT's decision-making algorithm sensor data with added random noise, representative of the actual sensors. So even an AUT whose decision-making process has no random components may produce different performance outcomes due to sensor noise (inertial navigation is a prime example). The bottom line is that repeated experiments are necessary to properly assess an AUT, and *a T&E framework that does not provide that capability cannot produce reliable assessments*.

The ability to repeat the same experiment many times under precisely the same scenario is a *critical asset* of TEAAS because it gives it the ability to isolate and analyze the performance variability that is *only due to the AUT*. On the other hand, in a real-life T&E situation, one would have to contend with the reality that the environment itself has random components that are not under the control of the tester. In that case, much effort would be expended in trying to achieve a "controlled experiment" to support the claim that variability of performance results is only due to the AUT. In TEAAS, the environment is completely under control. Therefore, the TEAAS framework can provide a complete assessment of the AUT's behavior by running it through each scenario N times and computing a relevant set of statistics. This overcomes the deficiency of other search-based testing strategies that miss realistic distributions. The value of N and the type of statistics depend on the requirements defined by the stakeholder, which may include specific confidence intervals.

Because of its focus on identifying potential trouble spots and failure modes, TEAAS uses measured *failure rate* as its fundamental measure of performance, which is the objective measure delivered to the CIO module. Since the measured failure rate is an estimate of the *probability of failure*, computed with N trials, it inherently addresses the uncertainty in the AUT's behavior. For example, TEAAS can produce reports such as: "Scenario A resulted in a measured failure rate of $\hat{F} = 80\%$, where we have $\alpha = 95\%$ confidence (i.e.,

there is a 0.95 probability) that $\hat{F}$ is within ε = 5% of the true failure rate." The number N of repeated trials for each scenario depends on the required values of α and ε, and the true failure rate. Since the true failure rate is not known, the value for N can be estimated iteratively through the relation [21],

$$N_{i+1} = \left(\frac{z_\alpha}{\varepsilon}\right)^2 \hat{F}_{N_i}\left(1 - \hat{F}_{N_i}\right), \qquad (1)$$

where z_α is the z-score for confidence value α, and $\hat{F}_{N_i}$ is the measured failure rate using N_i trials. After an initial assessment with, for example, N_1 = 10, the measured $\hat{F}_{10}$ is used in Equation 1 to estimate a new number of trials, say $N_2 = 15$, leading to a required number of 5 additional runs. This process can be repeated until no additional runs are needed. As we describe in Section 4.1, the cost of these additional runs is small for the parallelized TEAAS framework. TEAAS's capability of reporting a *complete assessment* of the AUT is enabled by (a) having total control of scenario generation and (b) being able to run the AUT through multiple scenarios in parallel.

In the TEAAS framework, the details of what it means for the AUT to fail or succeed are purposely kept separate from the CIO module, which only receives failure-rate measurements. This allows the CIO module to receive a consistent measure of performance that is continuous in the range [0, 1], regardless of scenario, AUT, stakeholder requirements, or anything else. Therefore, CIO algorithms can remain general and have no parameters or added complexities related to the specifics of any given problem.

For a given scenario, the measured failure rate is simply

$$\hat{F}_N = \frac{1}{N}\sum_{k=1}^{N} f_k, \qquad (2)$$

where f_k is 1 if the AUT failed in the k^{th} trial and 0 if it succeeded. All the complexity is therefore relegated to the *failure function f*, which is responsible for defining what constitutes failure or success (see Section 3.2). This is where the specifics of the problem and the stakeholder requirements come into play. For example, failure may be a result of failing any one of several requirements: $f = F_a \vee F_b \vee F_c$, where F_a is failure to reach a specified location in the allotted time, F_b is failure to conserve a specified amount of fuel, and F_c is failure to comply with traffic laws. This failure function must be designed to closely comply with the stakeholder requirements. (Involving the stakeholder in this design is even better.)

By this overall process, TEAAS (a) implements task and performance requirements as defined by the stakeholder (b), efficiently explores scenarios to discover the AUT's trouble spots and failure modes, and (c) provides a complete final assessment that addresses and reports the uncertainty in the process and results.

2.5. Simulation Realism

Closing the simulation-to-reality gap is critical when evaluating sequential decision-making systems under uncertainty because decisions are not isolated but rather build on each other over time. These systems operate in a continuous perception–decision–action loop. Unmodeled artifacts in the observed environment and state-estimation errors (e.g., misperceiving an obstacle, misjudging terrain) do not just cause a single mistake. That error becomes part of the system's internal state and propagates forward, influencing future decisions. This compounding effect can push the system into increasingly unsafe or suboptimal trajectories.

Real-world failures often start with sensing issues such as those resulting from weather (fog, rain, dust), lighting (glare, high dynamic range), object surfaces (reflective, transparent, low contrast), occlusions (lens contaminants, physical damage), platform motion (vibration, heat) and sensor hardware

limitations (noise, resolution, crosstalk). Sensor realism is mandatory for testing perception software in advanced autonomy applications. Without sensor realism, simulations cannot capture subtle but critical effects that may result in cascading failures, degraded performance trends and long-horizon risks.

Currently, TEAAS is focusing on LiDAR sensor fidelity with a specific interest in perception failure modes resulting from the interaction of the LiDAR with object surfaces (see Figure 3). For a given LiDAR beam incidence angle θ and surface distance R, the measured signal to noise ratio (SNR) and resulting detectability of the object depends upon the surface bidirectional reflection distribution function (BDRF) $\rho(\theta)$ and atmospheric transmittance between the LiDAR and the object $\gamma(R)$. For highly reflective (specular) surfaces, object partial observability or self reflections can cause significant perception errors. Since sensor model fidelity typically results in increased runtimes, TEAAS is leveraging the NVIDIA PhysX Engine, which provides high fidelity, fast running sensor models for commonly used sensors, such as LiDAR, visible cameras, radar and ultrasound, through the integration of ray tracing extension (RTX) rendering from Isaac Sim [22]. The RTX LiDAR includes reflection and scattering physics based on first order BDRF models for a variety of surface types and noise models for range inaccuracies and beam dispersion (via JSON configuration files).

Before proceeding with integrating the RTX LiDAR model into the TEAAS framework studies, we undertook a model validation study comparing laboratory measurements using a Velodyne VLP-16 LiDAR against the outputs of a simple mesh collision (ray casting) LiDAR model, the NVIDIA RTX LiDAR model, and the ANSYS's AVxcelerate "physics-accurate" real-time LiDAR sensor model [23]. The ANSYS model provides advanced full ("echo") waveform physics, calibrated BRDF models for a wide range of materials, and data driven capabilities allowing them to handle important real-world physical effects (e.g., atmospheric interference, multipath scattering, pulse distortion) and edge cases (e.g., sun glint, fog/rain, sensor lens occlusions).

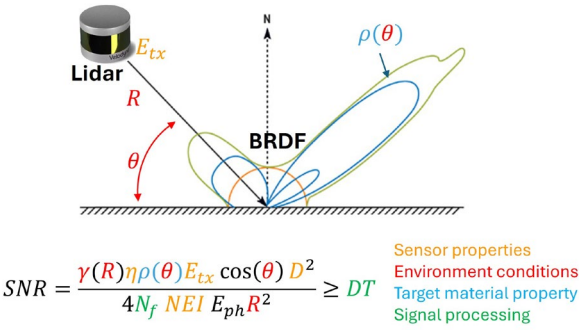

Figure 3: Interaction of LiDAR with object surface affects measured signal-to-noise ratio and object detectability leading to potentially significant perception errors.

The study involved LiDAR measurements and simulations at two distances (2 m and 10 m) from a braced thin flat plate (4 ft × 8 ft × ¾ in plywood sheet) and at three angles relative to the plate normal (0, 45, 85 deg) for two different plate surface materials: finished plywood, which is diffusive (Lambertian), and chrome tape, which is reflective (specular). The primary goals of the study were to (a) validate that the RTX model (already integrated with the Isaac Sim/Lab ecosystem) was suitable for our initial development efforts and to (b) explore the capabilities of the much more capable ANSYS LiDAR sensor model for subsequent integration.

Our key findings were that the RTX model captured the salient reflection properties observed in the lab measurements for the two BRDF surfaces. Most notably, for the reflective chrome tape and plate angle of 45 deg, only the front ~1/3 of plate was visible while at a plate angle of 85 deg the entire plate "disappeared." The partial and total loss of obstacle observability are clearly critical physical

effects that need to be properly assessed to maintain operational realism. The collision model captured the plate in its entirety under all conditions (a potentially catastrophic sensor modeling error). The ANSYS model captured all first order effects shown by the RTX model and showed higher fidelity second-order effects (e.g., pulse broadening, multipath effects).

3. SIMULATION EXPERIMENT DESIGN

3.1. Mission Use Case

For this paper, TEAAS framework research and development focuses on a use case derived from a representative real-world operational environment, which involves an autonomous ground vehicle (e.g., autonomous tank) with a precision-strike mission in a contested urban environment (Figure 4). The vehicle's mission is to navigate from its injection point to a known target location through a known urban environment (a-priori map) that includes multiple potential stressors such as a GPS-denied (GPS-D) zone induced by electronic jamming, a visually degraded environment (VDE) resulting from atmospheric and/or environmental conditions that impair LiDAR sensor range performance, and unknown obstacle. Test variables included the GPS-D zone location within the unoccupied map regions, VDE severity (as reflected by max LiDAR range), and three obstacle properties (location within the unoccupied map regions, orientation, and material properties). Three materials were used from the NVIDIA vMaterials 2.0 library: Polished Aluminum (specular, mirror-like), Scratched Chromium (semi-glossy) and OBS Wood (diffusive); herein referred to as polished aluminum, scratched aluminum, and wood.

In our simulated environment, we place the GPS-D zone and the obstacle at arbitrary locations in any of the four shorter parallel hallways (see Figure 4) and we encode the corresponding location variables as a number

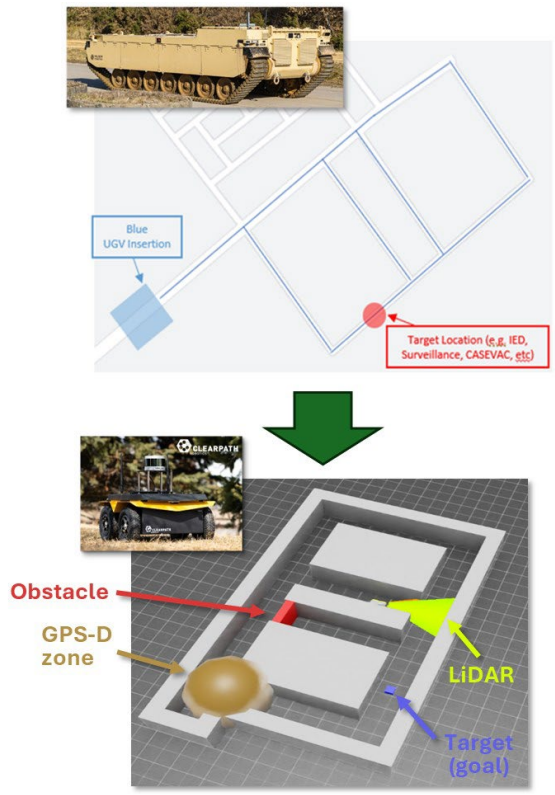

Figure 4: Mission use case representing real-world operational environment involving an autonomous ground vehicle precision strike mission within a contested urban environment.

whose integer part corresponds to the hallway label, and the fractional part corresponds to the location as a fraction within that hallway. Therefore, a location value of 0.77 means a point along the centerline of hallway '0' (closest hallway to AUT start location) and located at 77% of the length of the hallway. The remaining dimensions are represented directly by continuous variables or categorical identifiers; VDE severity (specified as a LiDAR maximum detection range) from 1-10 m, obstacle orientation from 0-180 degrees, and obstacle material as a categorical integer value of 1, 2, or 3.

So that we could maintain tight coupling with real-world T&E, a Clearpath Jackal robotic ground vehicle [24] was used as the autonomous vehicle to support physical experimentation in lab and field environments. Correspondingly, scaled physical analogs were

used for urban environment features and interferences. The direct comparison between simulated and real-world scenarios provided the ability to both validate predictions and evaluate effects of model fidelity [19].

3.2. Measuring AUT Performance

As discussed in Section 2.4, TEAAS uses measured *failure rate* $\hat{F}_N$, Equation 2, as its fundamental measure of performance to identify potential AUT trouble spots and failure modes. In this experiment, we define a failure function that returns a value of 1 (true) if either of two failure modes are true: (a) failure to reach a specified location G_k within a time limit, or (b) colliding with an obstacle in the environment C_k. This may be expressed as

$$F_k = [\neg G_k \vee C_k], \quad (3)$$

where G_k is true if the vehicle gets to within a distance $d = 0.3$m from the target within a time limit τ, and C_k is true if the vehicle collides with an obstacle. The value of τ is set as

$$\tau = 1.5 \frac{(\text{optimal path length})}{(\text{control velocity})}. \quad (4)$$

3.3. Autonomy Under Test

Hardware

As described in Section 3.1, a Clearpath Jackal robotic ground vehicle [24] was used as the autonomous vehicle to maintain tight coupling with real-world T&E. The Jackal standard sensor suite is being used for these TEAAS evaluation studies which includes an Xsens MTi-3 IMU for motion tracking, Clearpath's proprietary Quadrature wheel motor encoders for odometry, and a u-blox ZED-F9P GNSS module for positioning, all integrated into the autonomy stack. A Velodyne VLP-16 LiDAR was added to provide 360-degree environmental scanning for distance measurement and mapping.

TEAAS leverages the IsaacSim implementation of the Jackal, using the built-in USD model provided by Clearpath Robotics in collaboration with NVIDIA, along with model (.mdl) files that define optical material properties such as diffuseness and specularity for realistic visual rendering. The robot is configured in IsaacLab via customization of several physical parameters including mass, friction, and damping. Differential-drive kinematics are implemented using the standard equations $v = \frac{r}{2}(\omega_r + \omega_l),\ \omega = \frac{r}{L}(\omega_r - \omega_l)$, where v is linear velocity, ω is angular velocity, r is wheel radius, L is wheelbase, and ω_R, ω_L are right and left wheel angular velocities, respectively. This configuration allows Isaac Sim to simulate realistic motion, sensor feedback, and control.

Software

In practice, TEAAS' knowledge of the AUT is limited to its interfaces (sensors and actuators) with no insight into how it makes its decisions (i.e., black-box software). To support TEAAS framework development and evaluation, two different autonomy software modules were independently developed (AUT 1 and AUT 2) and integrated into TEAAS using the same API, such that the internals of each module are not exposed to the rest of the TEAAS pipeline.

3.4. Simulation Experiments

Simulation experiments involving four test cases were conducted to examine the effects of sensor model fidelity and autonomy level on predicted AUT mission performance and to evaluate the efficacy of the TEAAS framework for efficiently isolating critical scenarios (Table 1).

4. RESULTS

4.1. Uncertainty Quantification

As discussed in Section 2.4, repeated simulation runs are necessary to quantify uncertainty. To this end, the TEAAS framework repeats each scenario *N* times to compute

Table 1. Simulation experiments to examine effects of using proper simulation realism when assessing AUT performance and efficacy of TEAAS framework for efficiently isolating critical scenarios.

Experiment	Autonomy	Lidar	Effect
1.1	AUT 1	Collision	Baseline
1.2	AUT 1	RTX	Sensor Physics
2.1	AUT 2	Collision	Autonomy Level
2.2	AUT 2	RTX	Sensor Physics + Autonomy Level

failure rate (probability of failure) as the fundamental measure of performance which inherently addresses the uncertainty in the AUT's behavior.

Figure 5 shows four example scenarios from simulation experiment 1.1 (see Table 1). For each scenario, we show 100 trajectories (N = 100 repeated runs), where each trajectory is colored by the failure function f_k for run k: blue = mission success (f_k = 0), red = mission failure (f_k = 1). The mission success metric is described in Section 3.2. For each scenario, the measured failure rate $\hat{F}_{100}$ (or mission failure probability) computed from the 100 repeated runs is shown along with the measured standard deviation.

The robot mission is to get within a specified distance (0.3 m) from the goal, shown as the green shaded circle. The GPS denied zone is shown as a brown shaded circle, the obstacle is shown as a rectangle within the horizontal hallways, and the LiDAR max range is indicated by the yellow shaded circle (shown only at the AUT start location). In Figure 5(a), the original planned (optimal) route is uncontested (GPS-D zone and obstacle in fourth horizontal hallway) and the AUT reaches the goal for every run ($\hat{F}_{100}$ = 0). In Figure 5(b), the obstacle is at the beginning of the first hallway, and the GPS-D zone is at the beginning of the second hallway. The AUT seems to reliably detect the obstacle and replan accordingly. However, the GPS-denied zone is problematic, as the AUT seems to be replanning in a suboptimal way after entering it. Of the 100 runs, only 50 reached the goal in time resulting in a measured failure rate of $\hat{F}_{100}$ = 0.50. Figures 5(c) and 5(d) confirm the sensitivity of the AUT's performance to encountering a GPS-D zone, with measured failure rates of $\hat{F}_{100}$ = 0.76 and $\hat{F}_{100}$ = 1.0, respectively.

As a T&E pipeline, the objective of TEAAS is to discover these scenarios and behaviors and to present them in a clear and concise manner to stakeholders, not to speculate as to *why* these behaviors are occurring. Armed with TEAAS-provided information, the stakeholders (including the AUT developers) can investigate causes and take corrective actions. What is clear from this experiment is that simulating a scenario just once is not enough to provide useful information.

To properly address the uncertainty in the AUT's behavior, the number of repeated runs N must be large enough to meet the requirements defined by the stakeholder, such as a α = 95% confidence (z_α = 1.96) that the measured AUT failure rate $\hat{F}_N$ is within ε = 5% of the true failure rate. It is only through such an approach that we can reliably identify repeatable differences in performance across different tested systems. For the data presented in Figure 5, N = 100 yielded a 95% confidence that the estimated value is within ε = 10% of the true value (Equation 1). Since the focus of TEAAS is to identify potential trouble spots and failure modes for which the failure rate is large, say $\hat{F}_N$ > 0.9, a smaller number of repeated runs of N = 10 can be used to provide satisfactory performance (ε < 19% for $\hat{F}_N$ > 0.9, ε < 6% for $\hat{F}_N$ > 0.99). For this reason, we use N = 10 for the results presented throughout the remainder of this paper to minimize the required computational effort.

4.2. Simulation Realism

In this section we present simulation results for the four simulation experiments using a Monte Carlo search with 50,000 runs (5000 scenarios × 10 repeated runs per scenario),

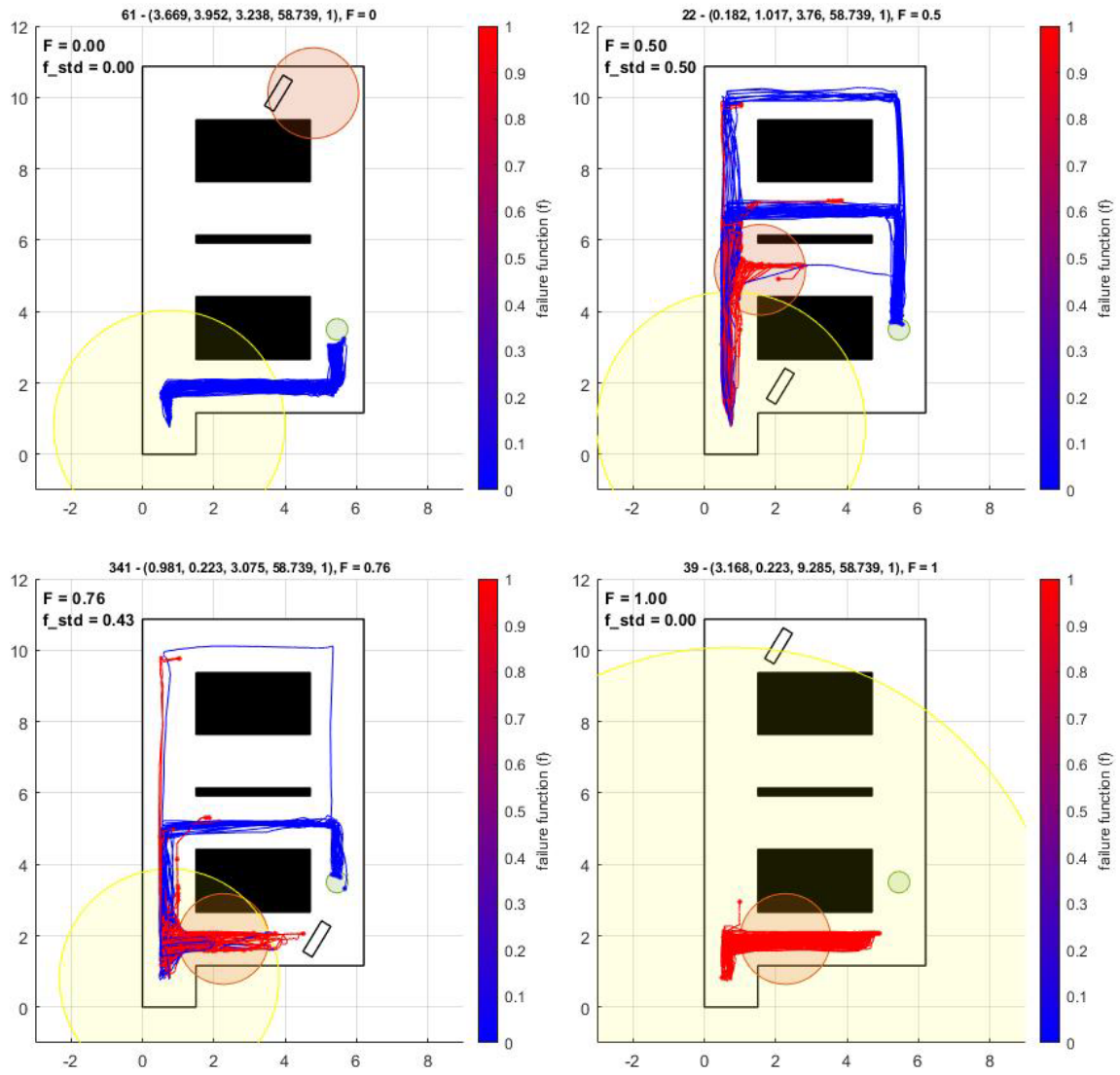

Figure 5: Effect of system uncertainty on mission failure rate for four different scenarios (combinations of state space variables) for Experiment 1.1; (a) F = 0.0, (b) F = 0.5, (c) F = 0.76, (d) F = 1.0. State space variables include GPS denied zone location, obstacle location, and LiDAR max range (obstacle angle and material are fixed since collision LiDAR is insensitive to both).

which provides a thorough exploration of the five-dimensional scenario state space. We use these results to illustrate the impact of sensor realism and autonomy level and to provide a baseline to which the CIO search results (Section 4.3) can be compared.

Changes in sensor realism (e.g., simulating the response of LiDAR to different materials versus a simple ray-tracing model for collisions) affect how the AUT senses the environment and therefore how it behaves, which may affect its failure rate. Similarly, differences in autonomy algorithms may result in different behaviors with consequent changes in failure rates. To detect and explain these differences, we must determine a) which scenario state-space variables are significant in affecting failure rates, and b) if there are failures that do not seem to be related to any of the state-space variables being considered. In this paper, we use standard statistical tests to address both tasks.

For each of the four data sets, we conduct statistical techniques (based on PCA and F-tests) to get a picture of which state-space variables are informative and which are redundant. This helps in selecting the most relevant states for visualization and further analysis. We also use residual diagnostics and lack-of-fit tests to quantify how much variance in failure rate is explainable vs unexplained. This can inform consideration of additional state variables or simply support the conclusion that there are some unexplained/unobservable behaviors in the given AUT.

In Figure 6, we present performance surface results for the four simulation experiments using the three variables that were identified as important across the collective data set. The failure rate is shown in the color bar. From these plots, we can make the following observations:

- **Effects of increased sensor realism.** In the case of the collision LiDAR, the statistical analysis reports that only two state variables significantly affect failure rates (GPS-D and obstacle locations). However, in the case of the RTX LiDAR an additional variable (obstacle material) is reported as being significant. The RTX LiDAR response to the different materials greatly affects the failure rate results. For example, the obstacle made of polished aluminum (essentially a mirror) is a significant issue for both AUT 1 and AUT 2 since it is not detectable. The LiDAR cannot detect this surface and both AUTs navigate right up to it and get stuck. The primary effect of material appears when the obstacle is in the first hallway, which is the optimal pre-planned pathway. Although a lesser effect, obstacle angle also impacts failure rate for both AUTs for the scratched aluminum surface over a limited range of angles near +/- 45 degrees (not observable in Figure 6 as plotted). This is due to the combined effects of the scratched aluminum BRDF (Section 2.5) and the observable obstacle area. The LiDAR max range is not a significant factor in changing the failure rate. (A different definition of what constitutes a failure could change this result, e.g., fuel consumption due to going down long hallways only to have to turn around and go back.)
- **Effects of autonomy level.** Even with the same sensors, AUT 1 and AUT 2 exhibit differences in failure rates and modes. This is evident from the analyses and visualizations in at least two ways. First, AUT 2 seems to have more problems negotiating GPS-D zones. Second, AUT 2 has some failure modes that are not statistically correlated to any of the five scenario state variables that we considered. These uncorrelated modes appear in Figure 6 as scattered points of medium to high failure rates.

The results described here provide an example of essential information that may be used by stakeholders as either a) inputs for T&E process test planning, b) an opportunity for revisiting the design and/or parameter-tuning of either system, or (c) a reason to reject poor performing systems. For high-dimensional parameter spaces, exhaustive testing or simulation of all possible scenarios is infeasible. What is needed is an approach to quickly extract the characteristic structure of the performance spaces in as few simulations as possible to provide performance assessments in operationally relevant time frames. In the next section, we present results of the TEAAS framework demonstrating efficiency and accuracy of our CIO approaches.

4.3. T&E Efficiency

Our CIO approach allows us to efficiently find critical regions of the solution space which can either be used directly for evaluation or provide feedback for future more focused exploration. Here, we focus on simulation experiment 2.2, which involves the RTX

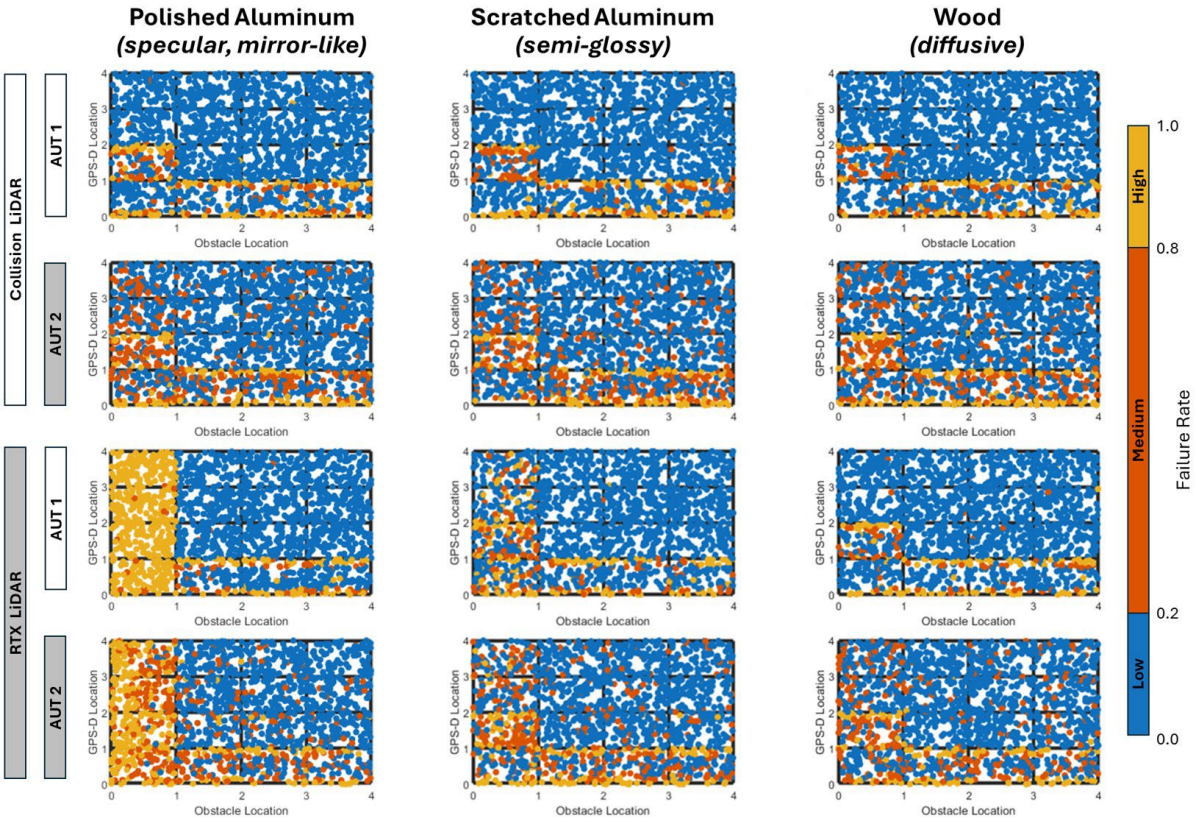

Figure 6: Scatter plots of the Monte Carlo data sets for all four simulation experiments using the variables identified through the importance-ranking results of the F-tests (GPS-D location, obstacle location, obstacle material). These results illustrate the impact of sensor realism and autonomy level and provide a baseline to compare the CIO search results (Section 4.3).

LiDAR and AUT 2. We chose this data set for analysis because the performance surface has the highest complexity of the four cases tested involving 3 variables with high importance and scattered failure modes that are not statistically correlated to any of the five scenario state variables that we considered.

To rapidly find scenario parameters that result in high failure rates for the AUT, we employ the jDE algorithm with a population size of 10. We tune the hyperparameters for the jDE search to limit exploration and rapidly converge on a critical point. Note that for our performance space there are several regions filled with points of high failure rates (i.e., $\hat{F}_{10} = 1$) so rather than searching for a global maximum, we are focused on rapidly finding a critical area and sampling its surroundings enough to verify it is not a standalone outlier, but a region that can confidently be exploited. The result of this search is shown in Figure 7.

In Figure 7, the jDE algorithm (upper row of plots) finds and converges on a set of high-failure-rate solutions within 800 total simulations (8 jDE iterations × 10 population members / iteration × 10 repeated runs / member). For a very small number of simulations (less than 2% of our Monte Carlo run in Section 4.2), this search found a critical region identifying the RTX LiDAR's poor performance with a polished aluminum obstacle in the first hallway. Repeated jDE searches can yield additional high-failure-rate solutions by adding constraints to avoid previous identified regions.

To demonstrate an alternate method for efficiently and searching the critical regions of

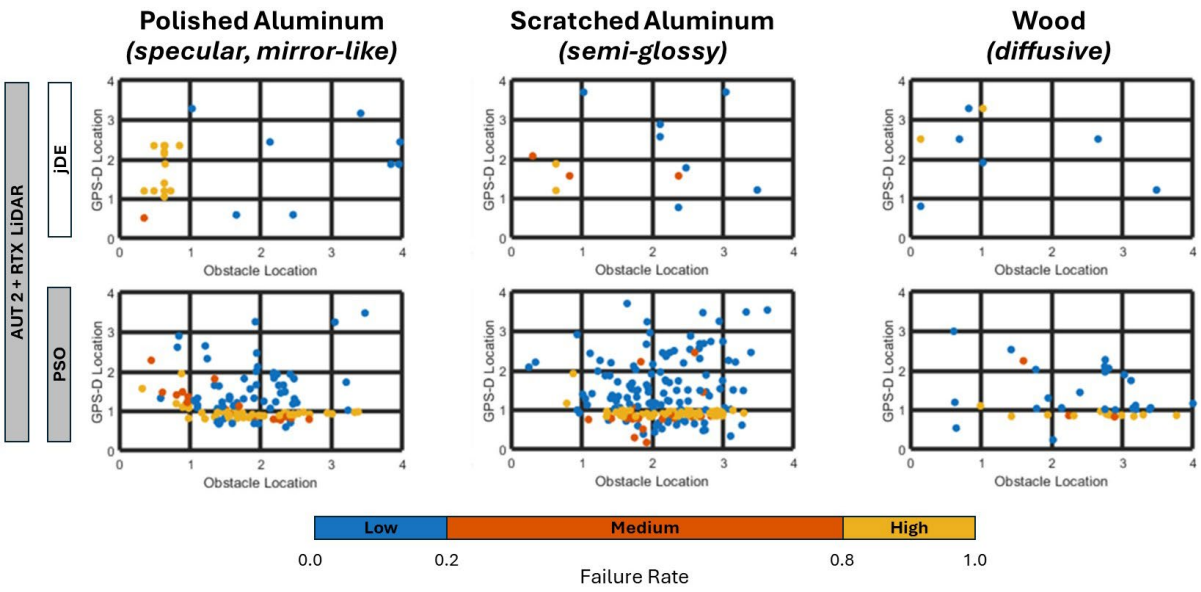

Figure 7: Scatter plots of the TEAAS CIO data sets for AUT 2 with RTX LiDAR. Upper row of plots is a modified jDE algorithm which converges with a minimum number of simulations on a critical failure region (8 iterations = 800 simulations). Lower row of figures is a modified PSO algorithm which converges after densely mapping two distinct critical regions (10 iterations = 4000 simulations). Monte Carlo search of the same use case with 50,000 simulations are shown in the lower row of plots in Figure 6. As state space dimensions increase, the CIO search exhibits increasing efficiency over Monte Carlo [19].

the performance space, we employ a PSO algorithm. While this algorithm would typically converge as quickly as the jDE, we implement several features to encourage additional exploration without fully sacrificing the efficiency of the algorithm.

Our PSO implementation results in a multimodal approach that uses multiple swarms with independent cognitive and social interactions so the swarm can focus on more than a single area. We also allow the algorithm to continue exploring for a set number of runs after the initial solution is found. To avoid the particles from settling and continuously sampling the same point, whenever a particle location is evaluated to have a failure rate of 1, that location in the parameter space will apply a small repulsive force to all particles in all swarms for all following iterations. The balance between this repulsive force and the social swarm behavior drives exploration to points that border the critical areas and find regions of similar performance. For this search, we use 2 swarms with 20 particles in each, for 40 parameter samples per iteration.

In Figure 7, the PSO algorithm (lower row of figures) identifies two distinct critical regions: (1) a dense mapping of a failure boundary associated with the AUT's inability to navigate a GPS-D zone at the end of the first hallway (GPS-D location ~ 0.9), and (2) a sparse identification of the specular obstacle region identified by jDE. These results were produced after 4000 simulations (10 iterations × 40 particles × 10 repeated runs), 8% of the Monte Carlo run. Note that there is a tradeoff between how long we let the algorithm search beyond conventional convergence and how completely we want to investigate the space.

Next steps, depending on the problem needs, could be to repeat additional searches to identify other failure regions or leverage the initial results of these fast search algorithms to motivate more focused and constrained explorations/experiments in understanding the performance of the AUT.

It's important to consider, however, that

while we evaluated the Monte Carlo surface in Section 4.2 for demonstrative and understanding purposes we were running with only a 5-dimensional space which is relatively small. Many problems can have orders of magnitude higher dimensionality, making brute force searches and dense explorations of interesting regions infeasible. Hence, the full nature of the output space would not be known and could not be used to focus/tweak exploration. As such, additional techniques are required to address underexplored regions of the CIO search data sets through surrogate model fitting and rule-mining techniques (see Section 5.1).

The scalability of the TEAAS CIO framework up to thousands of dimensions was explored in Persons [19]. Specifically, it was found that a jDE search exhibited increasing efficiency as the dimensionality grows, leveraging its capacity for targeted exploration and exploitation in high-dimensional state spaces. In contrast, the Monte Carlo search showed a declining efficiency trend with higher dimensionality, as its reliance on uniform random sampling becomes increasingly ineffective in adequately covering the exponentially expanding state space. As an additional demonstrative example of the efficiency of these techniques, we've performed research recently on a reinforcement learning (RL) T&E of AI-enabled fire control in complex ballistic missile defense scenarios. This problem required a 41-dimensional parameter space and using our particle swarm optimization algorithm with 20 particles we've seen convergence to optimal solutions within 7 iterations (7000 simulations), or < 1% of the number of simulations that Monte Carlo would have required.

We have found that choosing the correct search strategy for a given problem requires care and expertise. Ongoing research would include implementing a multiagent supervisor and finding sets of hyperparameters that most effectively search the space according to a stated requirement/goal. We could then use this more hands-off end-to-end simulation setup consistently across any autonomy/sensor combination for similar problem sets.

5. OTHER ONGOING & FUTURE WORK

This section discusses our other ongoing work into open research questions (Section 5.1) and framework development to address our OT&E customer's challenges (Section 5.2).

5.1. Research:

Underexplored Regions: Because CIO samples adaptively, with computational efficiency as an objective, the data will be clustered near high-failure regions and sparse elsewhere. This is a feature for optimization, but a problem for coverage assessment. Blending coverage of the state space with efficiency in finding high-failure regions is a research topic within CIO development. Outside of CIO, after the data has been produced, there are various options for additional stages to improve the TEAAS pipeline. These include:

- Fit a "surrogate" model to the data. This model should reproduce the original data as well as interpolate/extrapolate in places that data did not cover.
- The surrogate model should properly handle disconnected regions in the state space (e.g., the hallway-encoding scheme) and categorical variables (e.g., materials). This could be achieved by an ensemble of models, perhaps in a tree-based structure (e.g., random forest).
- The output of the surrogate model, its predictions, should be paired with a measure of uncertainty based on sampling density, and "unknown" or unexplored regions should be clearly shown.
- Using the surrogate model, employ techniques for automatic discovery of rules

(rule mining) that are human-readable. By providing predictions over sparsely sampled regions, the surrogate model may also be used to perform the same kind of analyses and visualizations that are performed on dense data sets (see Sec. 4.2).

High-D visualization: Although the results in this paper were constrained to state spaces with 5 dimensions, state spaces associated with real operational problems can have dimensions that are orders of magnitude larger. The scalability of the TEAAS CIO framework up to thousands of dimensions was explored in Persons [19]. What wasn't addressed is how such high-dimensional datasets can be visualized to preserve meaningful structure, relationships, and interpretability when projecting data from many dimensions into the limited dimensions humans can visually perceive.

To visualize high-dimensional data sets, the first step is to apply dimensionality-reduction techniques to isolate the top two or three significant variables that can explain the variance in failure-rate. There are several dimensionality-reduction techniques that are available (i.e., statistical analysis of variance, PCA, UMAP, t-SNE) and the best one depends on the particulars of the problem (e.g., the amount of data, the presence of categorical variables). Once the significant variables are isolated there are several canonical plots that may be used (e.g., scatter-plot matrices, 3D surfaces, parallel-coordinate plots, etc.). In Sec. 4.2 we used some of these techniques to analyze and visualize the dense data from four different experiments.

Dynamic Environments: The current TEAAS approach considers static environmental conditions that contribute to long-horizon risks. Although this is valuable for identifying scenarios that a test engineer (or the system designer) needs to evaluate, autonomous systems typically operate in dynamic environments in which the sequential decision-making landscape is continuously changing. To address such dynamic conditions, what is required is a strategy to find the *optimal sequence of configurations for the environmental parameters* that maximizes the likelihood of failure over time. One approach that has been explored by Nguyen [28] is to train an adversarial agent (e.g., reinforcement learning agent) to learn the optimal actions to configure the environment by observing the environmental parameters and the AUT internal states.

5.2. Development

Porting to military-specific tools: The current TEAAS framework is built on the GPU-based NVIDIA Isaac Sim/Lab environment that exploits massive parallelism to efficiently execute large numbers of concurrent environments on consumer-grade hardware. Although such tools are being widely embraced by the commercial Autonomous Vehicle (AV) Industry (e.g., Tesla, Waymo), these tools have not yet made their way into the Defense OT&E communities.

To address these customer challenges, we are porting TEAAS to military-specific tools/conops that MTSI is uniquely positioned to address, specifically, the Advanced Framework for Simulation, Integration, and Modeling (AFSIM) [29]. AFSIM is a serial, CPU-based simulation framework. To run AFSIM with TEAAS, we either need to rebuild AFSIM as a parallel / GPU tensor simulation (currently a non-starter) or forgo Isaac Sim/Lab, keep the AFSIM serial worldview, and parallelize around it (GPU-enabled models, distributed computing, multi-processing/multi-threading). The former is elegant and the ultimate (FAST) way forward, but the DoD community is not there yet.

In our research in other domains, we have already integrated a reinforcement learning (RL) T&E approach to run in line with AFSIM. This approach has been applied to

problems in cognitive electronic warfare and in large-scale highly complex ballistic missile defense scenarios. While this AFSIM integration work was not related to ground vehicle autonomy, the overall approach was similar to that described in this paper. We were attempting to determine simulation conditions that caused the AI algorithm under test to fail. This simulation was more complex and depended on a larger number of state space variables, making the algorithmic search much more challenging. We had a 41-dimensional parameter space that described the kinematics of a raid of threat trajectories that would have been entirely impossible to map out exhaustively with a Monte Carlo approach, especially since (as mentioned above) AFSIM is a serial simulation.

For any search to be feasible, any large-scale T&E using complex AFSIM scenarios will require distributed computing capabilities and a search approach that can fully utilize those resources. Our simulations were run on a High-Performance Cluster with 768 processing cores, each able to run a separate instance of AFSIM. The RL T&E framework was also developed to self-manage the handoff of jobs to these computational resources to take full advantage of them within the needs of each given algorithm. These simulations demonstrated high run-to-run variance (as quantified by an external tool used to train the internal AI models) and necessitated the use of 50 repeated runs. Even with 50 simulations per sample to reduce variance and serialized AFSIM runtimes of about 6 minutes per simulation, our T&E framework was able to converge on high-quality solutions in under 12 hours.

Fast accurate sensor models: In our R&D efforts thus far we have leveraged rigorous, physics-accurate, full-waveform ANSYS AVxcelerate sensor models and reduced order, GPU-accelerated, ray tracing NVIDIA RTX sensor models. Although both models provide realistic physical effects required for perception realism (though at different levels), each presents framework integration and/or parallelization challenges. To get around this bottle neck, we are investigating using narrow digital twins which are task-specific, physics-informed deep neural networks [30] trained on physics-accurate sensor data. Narrow digital twins provide a real-time representation of a physical asset, system, or process that focuses on a single, highly specific set of data and capabilities (optimized instantaneous look-up tables).

6. CONCLUSIONS

This paper presents work from an ongoing research and development effort to evaluate and demonstrate the advantages of using an intelligent optimization/search modeling and simulation framework to rapidly identify low performance scenarios for advanced autonomous systems.

The challenge of autonomous systems lies in ensuring that the entire integrated system behaves predictably and safely over time. The OT&E community widely recognizes the need for new, high-fidelity, cost-effective simulation and testing approaches that can handle complex state spaces and deliver reliable safety and performance assessments within operationally relevant timeframes. Insufficient testing creates unacceptable deployment risks and excessive testing delays operational technology deployment indefinitely. Our objective is to achieve an optimal balance by rigorously validating autonomous systems while enabling their deployment at a pace appropriate to stakeholder benefit and risk tolerance.

Key components of our approach include: (1) highly parallelized GPU-accelerated simulation and computational intelligence search-based optimization, (3) statistically significant number of repeated runs to provide consistent and reproducible performance results and to quantify uncertainty, (4) GPU-accelerated physics-accurate sensor

models to provide required levels of simulation realism. Initial TEAAS proof of concept and scalability demonstrations are described in AUVSI (2025). In this paper, we describe our efforts to address some critical complexity issues essential for proper evaluation of autonomous systems as summarized below.

- Experiments were conducted to validate repeated run requirements for reproducible failure rate simulation statistics. Although 100 repeated runs yield a 95% confidence that the estimated value is within $\varepsilon = 10\%$ of the true value across the range of measured failure rates, 10 repeated runs provide adequate performance for high failure rates of interest here.
- Experiments were conducted with a Monte Carlo search to examine the impact of sensor realism and autonomy level on predicted AUT mission performance and to provide a baseline to which the CIO search results can be compared.
- Data analytics techniques were applied to the Monte Carlo search data sets (PCA, F-Tests, residual analyses) to rank importance of state space variables and to identify uncorrelated unexplainable failure modes. Such techniques can also be applied to sparsely sampled CIO search data sets through surrogate model fitting and rule-mining techniques.
- Two key phenomenological observations were made from the data sets: (1) the RTX LiDAR response to the different materials greatly affects the failure rate due to inadequate observability of specular and glossy obstacles which interferes with AUT route planning, and (2) the two black box autonomies exhibited important differences in failure modes regarding their ability to navigate GPS-D zones and uncorrelated failure modes (essential information for OT&E stakeholders).
- Using the TEAAS CIO we presented techniques for rapidly finding parameter sets that reliably defeat the AUT (jDE) and efficiently searching the interesting regions of the performance envelope more densely than a naïve Monte Carlo approach could do within realistic time constraints (PSO).
- For less than 2% of our Monte Carlo simulation runs, the jDE search found the critical region in the state space where specular obstacles significantly impacted the AUT failure rates. The PSO search identified two distinct critical regions: (1) a dense mapping of a failure boundary associated with the AUT's inability to navigate a GPS-D zone, and (2) a sparse identification of the specular obstacle region identified by jDE.
- Our next step is to implement a multiagent CIO supervisor to identify an appropriate set of algorithms and hyperparameters that most effectively search the state space according to a stated requirement/goal for use across a broad range of autonomy/sensor configurations.

The TEAAS framework tools presented here are also being ported to the Advanced Framework for Simulation, Integration, and Modeling (AFSIM) to support military-specific conops. Currently this effort is focused on using distributed computing and multi-processing/multi-threading approaches since AFSIM is a serial, CPU-based simulation framework.

7. REFERENCES

[1] Z. Wei, H. Huang, G. Zhang, R. Zhou, X. Luo, S. Li, and H. Zhou, "Interactive critical scenario generation for autonomous vehicles testing based on in-depth crash data using reinforcement learning," IEEE Transactions on Intelligent Vehicles, 2024.

[2] L. Yang, S. Liu, S. Feng, H.Wang, X. Zhao, G. Qu, and S. Fang, "Generation of critical pedestrian scenarios for autonomous vehicle testing," Accident Analysis & Prevention, vol. 214, p. 107962, 2025.

[3] D. Karunakaran, J. S. Berrio Perez, and S. Worrall, "Generating edge cases for testing autonomous vehicles using real-world data," Sensors, vol. 24, no. 1, p. 108, 2023.

[4] H. Yang, Y. Zhou, and T. Chen, "Simadfuzz: Simulation-feedback fuzz testing for autonomous driving systems," arXiv preprint arXiv:2412.13802, 2024.

[5] V. Crespo-Rodriguez, Neelofar, and A. Aleti, "Pafot: A position-based approach for finding optimal tests of autonomous vehicles," in Proceedings of the 5th ACM/IEEE International Conference on Automation of Software Test (AST 2024), 2024, pp. 159–170.

[6] Y. Ji, Z. Xu, C. Zhao, K. Chen, and Y. Du, "Accelerated testing and evaluation for black-box autonomous driving systems via adaptive markov chain monte carlo," IEEE Transactions on Intelligent Transportation Systems, 2025.

[7] K. Viswanadha, F. Indaheng, J. Wong, E. Kim, E. Kalvan, Y. Pant, D. J. Fremont, and S. A. Seshia, "Addressing the ieee av test challenge with scenic and verifai," in 2021 IEEE International Conference on Artificial Intelligence Testing (AITest). IEEE, 2021, pp. 136–142.

[8] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2020, pp. 1–8.

[9] X. Zhang, W. Zhao, Y. Sun, J. Sun, Y. Shen, X. Dong, and Z. Yang, "Testing automated driving systems by breaking many laws efficiently," in Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis, 2023, pp. 942–953.

[10] A. Stocco, B. Pulfer, and P. Tonella, "Mind the gap! a study on the transferability of virtual versus physical-world testing of autonomous driving systems," IEEE Transactions on Software Engineering, vol. 49, no. 4, pp. 1928–1940, 2022.

[11] S. Sagmeister, P. Kounatidis, S. Goblirsch, and M. Lienkamp, "Analyzing the impact of simulation fidelity on the evaluation of autonomous driving motion control," in 2024 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2024, pp. 230–237.

[12] M. H. Amini and S. Nejati, "Bridging the gap between real-world and synthetic images for testing autonomous driving systems," in Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, 2024, pp. 732–744.

[13] F. M¨utsch, H. Gremmelmaier, N. Becker, D. Bogdoll, M. Zofka, and J. Z¨ollner, "From model-based to data-driven simulation: Challenges and trends in autonomous driving. arxiv 2023," arXiv preprint arXiv:2305.13960, 2023.

[14] K. Wang, C. Shen, X. Li, and J. Lu, "Uncertainty quantification for safe and reliable autonomous vehicles: A review of methods and applications," IEEE Transactions on Intelligent Transportation Systems, 2025.

[15] R. Grewal, P. Tonella, and A. Stocco, "Predicting safety misbehaviours in autonomous driving systems using uncertainty quantification," in 2024 IEEE Conference on Software Testing, Verification and Validation (ICST). IEEE, 2024, pp. 70–81.

[16] W. Zhou, Z. Cao, N. Deng, K. Jiang, and D. Yang, "Identify, estimate and bound the uncertainty of reinforcement learning for autonomous driving," IEEE

Transactions on Intelligent Transportation Systems, vol. 24, no. 8, pp. 7932–7942, 2023.
[17] G., Loaiza-Ganem, V. Villecroze, & Y. Wang, "Deep Ensembles Secretly Perform Empirical Bayes," arXiv preprint arXiv: 2501.17917, 2025.
[18] NVIDIA Isaac Lab, https://developer.nvidia.com/isaac/lab
[19] B. Persons, S. Snarski, N. Khan, "Test and Evaluation AI-enabled Autonomous Systems," *AUVSI Xponential 2025, Technical Research & Platform Development Track*; paper published in conference bound proceedings, Houston, TX, 22 May 2025.
[20] B, Zolghadr-Asli, "Computational intelligence-based optimization algorithms: From theory to practice," CRC Press, 2023.
[21] D. Bertsekas, J. Tsitsiklis, "Introduction to Probability," 2nd ed., Athena Scientific, Belmont, MA, 2002.
[22] NVIDIA Isaac Sim RTX Sensors, https://docs.isaacsim.omniverse.nvidia.com/5.1.0/sensors/isaacsim_sensors_rtx.html
[23] ANSYS AVxcelerate Sensors, https://www.ansys.com/products/av-simulation/ansys-avxcelerate-sensors
[24] Clearpath Robotics Jackal, https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/
[25] N. Kaiser, "Autonomous Navigation using Jackal UGV & Velodyne LIDAR," https://github.com/njkaiser/nu_jackal_autonav
[26] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in 2017 IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS), 2017, pp. 31–36.
[27] H. Taheri, S. R. Hosseini, and M. A. Nekoui, "Deep reinforcement learning with enhanced ppo for safe mobile robot navigation," under review by Int. J. of Intelligent Machines and Robotics, arXiv:2405.16266, 2024.
[28] T-H Nguyen, et al., "Generating Critical Scenarios for Testing Automated Driving Systems," arXiv:2412.02574v1 [cs.RO], 3Dec2024
[29] Advanced Framework for Simulation, Integration, and Modeling (AFSIM), https://dsiac.dtic.mil/models/afsim/
[30] J. O'Farrill, N. Highsmith, "Electro-optical Signature Generation Using Physic-Aided Deep Neural Networks," Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), 2025, https://amostech.com/TechnicalPapers/2025/Poster/O'Farrill.pdf

8. CONTACT INFORMATION

Dr. Steve Snarski
Technical Fellow
Modern Technology Solutions, Inc.
360C Quality Circle, Suite 310
Huntsville, AL 35806
steve.snarski@mtsi-va.com
Linked In: https://www.linkedin.com/in/stephen-snarski-ph-d-3a785613/

9. ACKNOWLEDGMENT

This work was funded through internal research and development funds of Modern Technology Solutions, Inc.

10. ACRONYMS

AFSIM	Advanced Framework for Simulation, Integration, and Modeling
AI	Artificial Intelligence
AUT	Autonomy Under Test
BRDF	Bidirectional Reflective Distribution Function
CIO	Computational Intelligence Optimization

FOV	Field Of View
GPS	Global Positioning System
GPS-D	GPS Denied
GPU	Graphics Processing Unit
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
jDE	Self-adaptive Differential Evolution
LiDAR	Light Detection And Ranging
MTSI	Modern Technology Solutions, Inc.
OT&E	Operational T&E
PSO	Particle Swarm Optimization
RL	Reinforcement Learning
T&E	Test and Evaluation
TEAAS	T&E of Advanced Autonomous Systems
UQ	Uncertainty Quantification
VDE	Visually Degraded Environment