A STRESS TESTING FRAMEWORK FOR AUTONOMOUS SYSTEM VERIFICATION AND VALIDATION (V&V)

Gregory Falco *

Johns Hopkins University
Institute for Assured Autonomy
Baltimore, MD

Leilani H. Gilpin

Massachusetts Institute of Technology
CSAIL
Cambridge, MA

ABSTRACT

Autonomous cyber-physical systems are prone to error and failure. Verification and validation (V&V) is necessary for their safe, secure and resilient operations. Methods to detect faults in aerospace engineering (fault trees) and later adapted for security (attack trees) could capture a wide array of critical risks and we argue how stress testing could be a pragmatic approach to evaluating the assurance of autonomous cyber-physical systems.

***Index Terms*—** Stress testing, Autonomous Systems, Formal Methods, Cyber-physical systems, Robust AI, XAI, Assured autonomy, Verification and Validation, V&V

1. INTRODUCTION

Cyber-physical autonomous systems are prone to failures and are not currently tested properly. Verification and validation (V&V) testing must fully capture both physical safety and digital security risks, which are compounded by the inherent complexity of autonomous systems. Current V&V testing and proving properties can harden these systems, but they are inadequate–it is impossible to "formally" test all failure modes. The key idea is that these failures are not isolated. Instead of building provable properties, our research is a complementary approach: we propose work on *AI stress testing*.

Stress testing is crucial for autonomous cyber-physical systems in *open environments*. Image recognition systems have been shown to be brittle and biased [1], and this is illuminated as a threat to humanity in the domain of self driving cars [2]. These mistakes and errors need to become test cases, similar to the types of stress testing that is done in consumer vehicles, aerospace systems, and commercial aircrafts. We discuss the merits of stress testing via a risk-based approach to build trust and security in autonomous, cyber-physical systems. While a stress test should be customized to the system of interest, we propose a consistent approach to evaluating and interpreting the results of stress tests to successfully compare V&V tests across autonomous agents. Our stress test evaluation framework is based on methods that have been in use for decades in safety science. We provide an example for how our stress testing framework could be employed for the autonomous agents that comprise NASA's future lunar habitat - the Artemis Base Camp.

2. PRIOR WORK ON V&V FOR AUTONOMY

Safety-critical systems need appropriate testing protocols. Human operators of machinery or personal vehicles are subject to driving tests, safety protocols, and certifications. Autonomous operators should be subject to the same types of testing.

But what do we seek to understand from these tests? There has been work on documenting failures, but there is an increasing need to categorize and prioritize autonomous system needs and challenges [3]. The AI incident database [4] was released as a means to avoid "repeated AI failures [by] making past failures known." We are inspired by the work of the AI incident database to distill past failures into an accessible testing framework. There have been many V&V mechanisms proposed for autonomous agents[5]. Below is a small sampling of some predominant tests for autonomous agents, each of which have notable draw-backs.

Formal methods is among the most used V&V testing techniques that has been employed for safety-critical systems [6, 7]. However, there are certain characteristics of autonomous agents that are not conducive to formal methods. For example, autonomous agents generally lack "unambiguous" requirements and specifications, they operate in semi-known environments that may change at a moment's notice, and they may hand off control to a human operator at some point in the mission thereby introducing further uncertainty into the operating equation [8]. Additionally, there is often incomplete information about what went into the training of the agent and its subsequent learned behavior. The agent may have learned "unsafe" behavior, unknown to operators [8].

There are also challenges using formal methods to eval-

*This research was supported in part by the Fulbright Commission and the National Science Foundation.

uate the security of an autonomous agent. Many have tried to remedy formal methods for autonomous applications [9, 10, 11], including work that is quite similar to our contribution: using some sort of fault tree to derive verification properties [12, 13]. But, formal methods has struggled to gain traction in security testing communities, given the ever-expanding state space and unpredictability of creative attackers. For example, formal methods will not be able to detect a potential issue associated with previously unseen vulnerabilities or exploits [14]. This is the very reason why many security researchers still employ attack trees rather than formal methods to evaluate security holes in complex systems. Ultimately, the challenge with formal methods is that they are generally reliant on specifications, static analysis, well-known outcomes and determinism to develop a strong model - whereas autonomous agents change at run-time given that they are constantly learning and making decisions in undefined environments.

Differential testing is generally engaged to make sure that different versions of software that may have been updated produce a consistent output [15]. It has been used for both cyber-physical systems and information technology systems alike. A challenge engaging this approach for autonomous agents is that it only intends to capture changes in operation between different versions - not identify net new risks.

Simulation testing is commonly employed in reinforcement learning, where the agent training process involves sequential Markov decision problems which act as essentially a series of stress tests. Algorithms that can be engaged for this simulation include a Monte Carlo tree search or deep reinforcement learning[5]. Usually, these ”tests” occur in a realistic, but closed-world simulation. The problems arise with this approach when these agents transfer to real, open world environments given their dependence on some pre-existing domain knowledge which can be poorly defined in unknown environments.

3. FAILURE TYPES AND THEIR STRESSORS

There are three failure axes for cyber-physical systems. The system can fail due to an internal fault (in Section 3.1), or an error that can be pinpointed to a part or connection inside the system. Another failure mode is due to an unexpected external factor (in Section 3.2); an attack or one-off incident from external factors, such as weather. Finally, a less considered, but equally important failure mode in the context of testing is that of ethics (in Section 3.3). Autonomous agent ethics has been robustly discussed for autonomous agents [16], but less so in the context of testing.

For each axis, we propose a series of stressors that induce the associated failures. The stressors should be individually tested for each autonomous agent. The specific tests employed for the stressors should vary depending on the type of agent being stress tested; however the tests should be evaluated in a consistent manner so that systems engineers can compare and prioritize failures.

Importantly, the questions aim to distinguish between failures that matter in the context of autonomous agent resilience and others that do not. Autonomous agents are inherently complicated and will therefore be prone to failures - but not all will be consequential. Stress tests should elucidate this distinction between failure severity. Resilience is used as the baseline requirement for distinguishing what failures matter because it indicates what failures an agent could tolerate while still achieving its mission. The questions are explicitly described further in the Stress Testing Evaluation Framework.

3.1. Internal Fault

Internal faults can be caused by stresses due to a failed component or a failed connection between parts. One type of local failure is a mechanical failure such as a sensor failure. This occurs when a mechanical component is obfuscated, misaligned, misinterpreted or malfunctions altogether. An obfuscation example is LiDAR sensors that cannot detect objects in the rain or snow [17]. Since sensor data is commonly noisy, it can be easily misinterpreted, which happens in wireless networks, vehicles, and other smart systems. And finally, sensors, like all subsystems can malfunction or crash. The main commonality between these failures are that they are *local* to the sensor subsystem.

Software bugs are another stress that can result in an internal fault, which can be local or between components. An example is the NaN error in the autonomous racecar[1], or the hallucinating behavior of deep network networks[1], which can be monitored with commonsense data and rules [18]. Other communication failures can be due to network latency, incorrect assumptions, or other external factors, which we cover in the next section.

3.2. External Forces

External forces on an agent could induce a variety of failures. One such external force is that of a cyberattack. Autonomous cyber-physical systems have a great deal of surface area that could be subject to attack. Attackers may be particularly attracted to autonomous agents given the grandeur and physical impact of their potential failure. Attackers can target anything from the training data set to the control system itself. Cyber-physical autonomous agents are finely tuned where even a slight timing attack could throw off the real-time operating systems inherent to these agents. A timing attack to an autonomous robotic arm operating in a chemical plant could cause an explosion should chemical compounds be mixed at the incorrect frequency. While not a fully autonomous agent,

[1] Autonomous racecar slams into a wall: https://www.thedrive.com/news/37342/autonomous-race-car-starts-test-lap-immediately-slams-into-wall

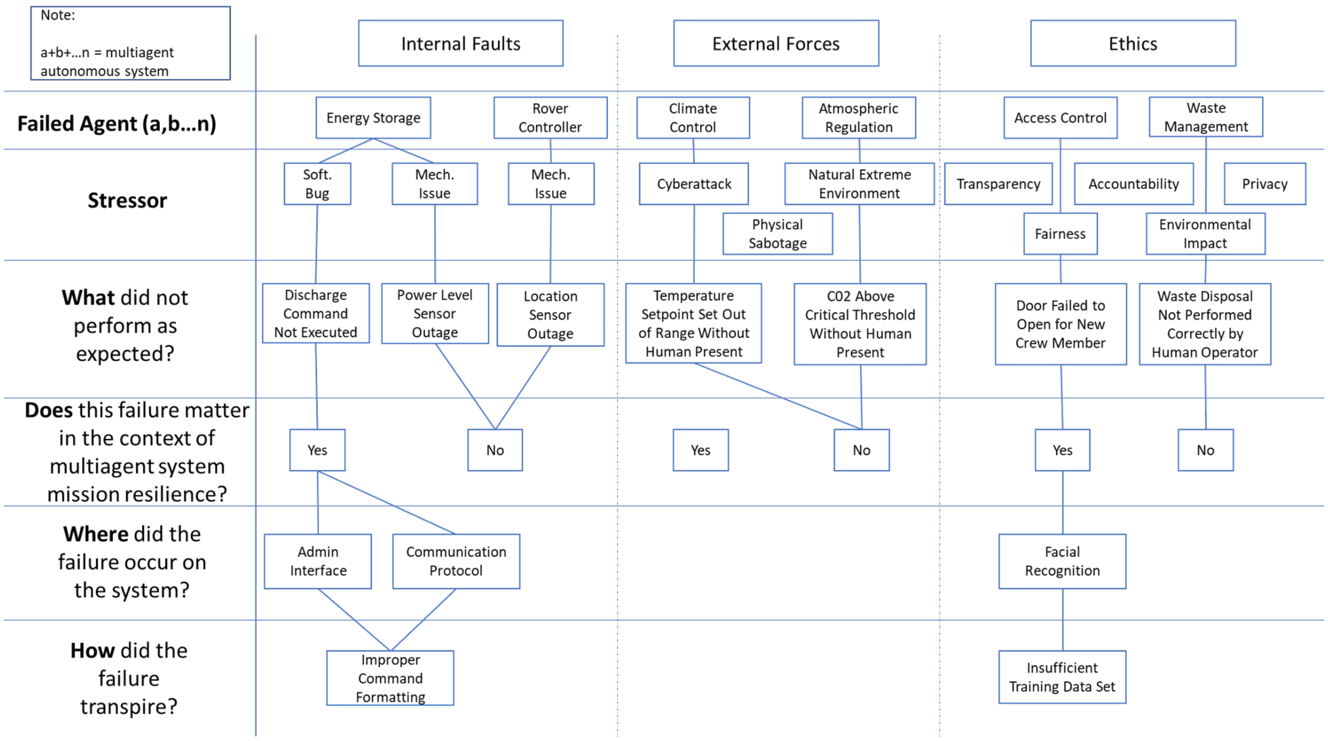

Fig. 1. Our stress test evaluation framework.

a similar cyber incident occurred at a German steel mill in 2014 causing significant damage to the plant [19].

The physical nature of cyber-physical autonomous agents also poses the risk of physical sabotage. Drones are increasingly autonomous and being employed for important tasks such as in military surveillance and reconnaissance missions. There have been incidents where semi-autonomous drones have been shot down such as the Global Hawk Spy Drone by Iran in 2019 [20].

A less considered, but equally devastating external force is natural extreme environmental conditions, such as weather. Many autonomous agents are designed to operate in extreme environments so that humans do not have to be present. An example of such an autonomous robotic agent is one used for deep sea arctic exploration [21]. Autonomous agents have challenges with far less extreme environments - such as in rain, wind or fog, which have been demonstrated to induce mission failure in autonomous vehicles [22]. Such extreme natural conditions' impact can become compounded in autonomous agents - inciting system failure.

3.3. Ethics

An ontology of ethics stressors have been previously enumerated to include: transparency, accountability, privacy, fairness [16]. Each has the capacity to cause a failure that inhibits a system's mission resilience. An additional ethics stressor that has not been as discussed is environmental impact. Specifically, this could include how a system's performance may damage its surrounding environment while achieving its mission. For example, an autonomous robot whose mission is to retrieve a series of artifacts from a delicate environment such as an archaeological excavation may succeed in retrieving the artifact at the expense of the surrounding environment that housed the artifact - thereby inhibiting its ability to return to retrieve further specimens. This presents an ethical failure of the autonomous agent.

4. STRESS TESTING EVALUATION FRAMEWORK

We propose a hierarchical tree structure that serves to aid systems engineers to evaluate each agent's stressors across an autonomous system. This hierarchy employs the framework established for fault tree analysis (originally developed for the aerospace community in the 1960s) [23], which has been used extensively in the field of safety science and then later adapted by the security community in the form of attack trees [24]. Tree structures have been used to enumerate risk for automotive reliability and safety studies[25]. Generally these tree structures do not have significant structural requirements beyond enumerating subsequent detail as one descends the tree on how a component failed or is attacked. However, by furnishing each tree "branch" level with a series of questions about the failure, the systems engineer can more easily compare and prioritize the failures for each agent. Establishing further structure for the tree hierarchy has been previously demonstrated [26].

5. EXAMPLE SCENARIO

To demonstrate how the stress testing evaluation framework could be employed, a sample is illustrated in Figure 1 concerning NASA's future autonomous lunar habitat. The scenario illustrates an autonomous agent that has been stress tested for each stressor for each autonomous agent described in Section 3. The framework would have been completed by a systems engineer after the stress test for each agent. A systems engineer could use any level of the tree hierarchy (question) as their prioritization filter; however, the failures that affect mission resilience should be addressed first.

The lunar habitat will be composed of a series of autonomous control system agents that will be required to work together with other agents and humans. In some cases agents will be acting with humans present and co-operated, while at other times the agents will be acting without the physical presence of humans. In all cases, the agents will be working towards the mission of establishing a sustained habitable environment that enables scientific exploration on the lunar surface. Agents that compose the autonomous lunar habitat may include, but is not limited to: resource (water, energy, materials, etc.) harvesting, resource (water, energy, materials, etc.) management (storage, allocation, discharge, etc.), vehicle control, vehicle maintenance, climate control, atmospheric regulation, access control, and waste management. The success of the mission will be reliant on the accomplishment of each agent's operations as well as their interactions. For example, a vehicular control system will be dependent on the resource management system given a lunar rover will require proper energy storage, allocation and distribution. The lunar habitat will exist in an inherently extreme environment with considerable failure risks from external forces. Given the complexity of the autonomous agents, there are also many internal faults that can possibly occur. The necessary agent-human and agent-environment interaction also poses the opportunity for ethical failures. Each stressor must be evaluated in the context of the operating parameters of the autonomous agent at any given time. Evaluating NASA's future lunar habitat is an especially interesting and critical case for stress testing given the lack of physical access to devices, extreme costs associated with repairs and the delicate nature of the overall mission. One autonomous agent's failure could ostensibly cause the lunar habitat to fail.

6. DISCUSSION AND FUTURE WORK

Although there has been previous work on documenting and classifying failure cases, there has been little work on what information is sought when a system fails. In this paper we have shown a proof-of-concept stress testing framework for cyber-physical autonomous agents. This is especially important for *assured autonomy* and building trust in our autonomous counterparts.

As autonomous agents take control of operation that was previously entrusted to humans, it is necessary to test these mechanisms in the same way that human operators are tested. With the increasing number of connections, parts, and complexity of these systems, the state space has evolved making it challenging to fully address using formal methods. Unlike other V&V frameworks, our approach offers a means for flagging issues without extensive data or quantitative analysis (which may be unavailable). The stress testing framework can customized to prioritize stressors and their associated failures to help ensure the autonomous agent's assurance. While some existing V&V methods are useful for static systems, it is time for the community to expand how autonomous agents are evaluated and stress testing will be a critical aspect of this. Now, it is imperative that we start testing and refining stress test evaluation frameworks such as the one proposed to help build trust in autonomous agents.

7. CONCLUSION

In this paper, we have revisited themes from classical fault diagnostics to chart a path forward for stress testing autonomous cyber-physical systems. Our stress testing framework enables end users to determine what they should be testing for (given each system is unique), while leaving it up to the systems engineers to devise sufficient tests for their systems. We do not believe that the stress testing framework proposed is comprehensive and we encourage the community to build on this to propose new questions critical to mission resilience and system assurance. Fundamentally, there is merit to strategically breaking the autonomous agent and methodically questioning and documenting what went wrong.

8. REFERENCES

[1] Anh Nguyen, Jason Yosinski, and Jeff Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, 2015, pp. 427–436, IEEE.

[2] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[3] Gregory Falco, "Autonomy's hierarchy of needs: Smart city ecosystems for autonomous space habitats," in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2021.

[4] Sean McGregor, “Preventing repeated real world ai failures by cataloging incidents: The ai incident database,” 2020.

[5] Anthony Corso, Robert J Moss, Mark Koren, Ritchie Lee, and Mykel J Kochenderfer, “A survey of algorithms for black-box safety validation,” *arXiv preprint arXiv:2005.02979*, 2020.

[6] Shaoying Liu, Victoria Stavridou, and Bruno Dutertre, “The practice of formal methods in safety-critical systems,” *Journal of Systems and Software*, vol. 28, no. 1, pp. 77–87, 1995.

[7] Yingxu Wang, Ming Hou, Konstantinos N Plataniotis, Sam Kwong, Henry Leung, Edward Tunstel, Imre J Rudas, and Ljiljana Trajkovic, “Towards a theoretical framework of autonomous systems underpinned by intelligence and systems sciences,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 52–63, 2020.

[8] Ufuk Topcu, Nadya Bliss, Nancy Cooke, Missy Cummings, Ashley Llorens, Howard Shrobe, and Lenore Zuck, “Assured autonomy: Path toward living with autonomous systems we can trust,” *arXiv preprint arXiv:2010.14443*, 2020.

[9] Kristin Yvonne Rozier, “Specification: The biggest bottleneck in formal methods and autonomy,” in *Working Conference on Verified Software: Theories, Tools, and Experiments*. Springer, 2016, pp. 8–26.

[10] Michael Winikoff, “Assurance of agent systems: what role should formal verification play?,” in *Specification and Verification of Multi-agent systems*, pp. 353–383. Springer, 2010.

[11] Hoang Tung Dinh and Tom Holvoet, “A framework for verifying autonomous robotic agents against environment assumptions,” in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2020, pp. 291–302.

[12] Marie Farrell, Matthew Bradbury, Michael Fisher, Louise A Dennis, Clare Dixon, Hu Yuan, and Carsten Maple, “Using threat analysis techniques to guide formal verification: A case study of cooperative awareness messages,” in *International Conference on Software Engineering and Formal Methods*. Springer, 2019, pp. 471–490.

[13] Michael Winikoff, “Towards deriving verification properties,” *arXiv preprint arXiv:1903.04159*, 2019.

[14] J Voas and K Schaffer, “Whatever happened to formal methods for security?,” *Computer*, vol. 49, no. 8, pp. 70, 2016.

[15] William M McKeeman, “Differential testing for software,” *Digital Technical Journal*, vol. 10, no. 1, pp. 100–107, 1998.

[16] Pradeep K Murukannaiah, Nirav Ajmeri, Catholijn M Jonker, and Munindar P Singh, “New foundations of ethical multiagent systems,” in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 1706–1710.

[17] N. Charron, S. Phillips, and S. L. Waslander, “Denoising of lidar point clouds corrupted by snowfall,” in *2018 15th Conference on Computer and Robot Vision (CRV)*, 2018, pp. 254–261.

[18] Leilani H Gilpin, Jamie C Macbeth, and Evelyn Florentine, “Monitoring scene understanders with conceptual primitive decomposition and commonsense knowledge,” *Advances in Cognitive Systems*, vol. 6, pp. 45–63, 2018.

[19] Robert M Lee, Michael J Assante, and Tim Conway, “German steel mill cyber attack,” *Industrial Control Systems*, vol. 30, pp. 62, 2014.

[20] Lily Hay Newman, “The drone iran shot down was a $220m surveillance monster,” *Wired.com*, 2019.

[21] Clayton Kunz, Chris Murphy, Hanumant Singh, Claire Pontbriand, Robert A Sohn, Sandipa Singh, Taichi Sato, Chris Roman, Ko-ichi Nakamura, Michael Jakuba, et al., “Toward extraplanetary under-ice exploration: Robotic steps in the arctic,” *Journal of Field Robotics*, vol. 26, no. 4, pp. 411–429, 2009.

[22] Shizhe Zang, Ming Ding, David Smith, Paul Tyler, Thierry Rakotoarivelo, and Mohamed Ali Kaafar, “The impact of adverse weather conditions on autonomous vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car,” *IEEE vehicular technology magazine*, vol. 14, no. 2, pp. 103–111, 2019.

[23] AF Hixenbaugh, “Fault tree for safety,” Tech. Rep., Boeing Co Seattle WA Support Systems Engineering, 1968.

[24] Bruce Schneier, “Attack trees,” *Dr. Dobb’s journal*, vol. 24, no. 12, pp. 21–29, 1999.

[25] Howard E Lambert, “Use of fault tree analysis for automotive reliability and safety analysis,” *SAE transactions*, pp. 690–696, 2004.

[26] Gregory Falco, Arun Viswanathan, Carlos Caldera, and Howard Shrobe, “A master attack methodology for an ai-based automated attack planner for smart cities,” *IEEE Access*, vol. 6, pp. 48360–48373, 2018.