The Journal of Open Source Software

AST Toolbox: An Adaptive Stress Testing Framework for Validation of Autonomous Systems

Mark Koren[1], Xiaobai Ma[1], Anthony Corso[1], Robert J. Moss[1], Peter Du[2], Katherine Driggs Campbell[2], and Mykel J. Kochenderfer[1]

1 Stanford University **2** University of Illinois

DOI: 10.21105/joss.03312

Software

- Review
- Repository
- Archive

Editor: Stefan Pfenninger

Reviewers:

- @giodegas
- @abhiramm7

Submitted: 22 April 2021
Published: 29 May 2021

Summary

The AST Toolbox is a python package that uses reinforcement learning to find failures in autonomous systems while treating the system and the simulator as black-boxes. Adaptive stress testing (AST) was recently developed to identify the most likely failure of a system in simulation (Lee et al., 2020). AST frames the validation problem as a Markov decision process (MDP) (Kochenderfer, 2015), where the AST agent controls the simulation through environment actions to find the most likely failure in the system under test (Koren et al., 2020). Understanding the most likely failure allows engineers to address issues in their system prior to deployment. To facilitate the use of AST for validation, this paper presents a new software package called the AST Toolbox.

The AST Toolbox is a software package for integrating AST with any simulator, making validation of autonomous agents easier. There are three major components to the toolbox: the solvers, the simulator interface, and the reward function. The solvers are different algorithms for finding the most likely failure of the system under test. The AST simulator interface provides a systematic way of wrapping a simulator to be used with the AST environment. The reward function uses the standard AST reward structure (Koren et al., 2020) together with heuristics to guide the search process and incorporate domain expertise.

The AST method is shown in Figure 1a, and the corresponding AST Toolbox architecture is shown in Figure 1b. The three core concepts of the AST method (simulator, solver, and reward function) have abstract classes associated with them. These base classes provide interfaces so they can interact with the AST module, represented by the `ASTEnv` class. `ASTEnv` is a gym environment (Brockman et al., 2016) that interacts with a wrapped simulator `ASTSimulator` and a reward function `ASTReward`. In conjunction with `ASTSpaces`, which are gym spaces, the AST problem is encoded as a standard gym reinforcement learning problem. Many open-source reinforcement learning algorithms are written to work with gym environments, and our solvers are implemented using the `garage` framework (The garage contributors, 2019). The solver derives from the `garage` class `RLAlgorithm`, and uses both a `Policy`, such as a Gaussian LSTM, and an optimization method, such as TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017). Using the `garage` framework, new solvers can be quickly implemented.

Statement of Need

Prior to deployment, it is important to validate that autonomous systems behave as expected to help ensure safety. It is generally difficult to provide comprehensive testing of a system in the real world because of the large space of possible edge cases. In addition, real-world testing

might present a safety risk. We therefore have to rely upon simulation using models of the environment to provide an adequate level of test coverage.

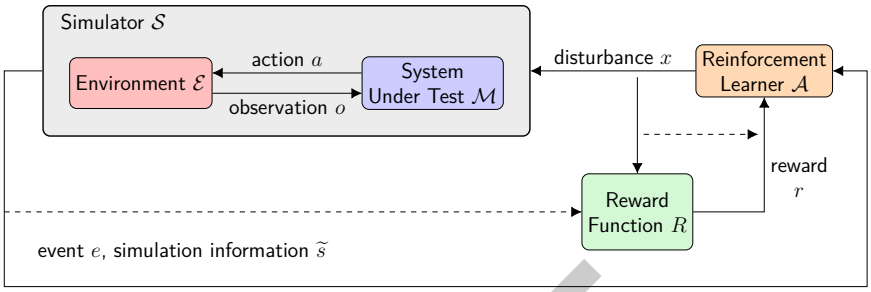
However, even in simulation, validating the safety of autonomous systems is often challenging due to high-dimensional and continuous state-spaces. Recent research has explored the use of adaptive stress testing (AST) for finding the most likely system failures in simulation using reinforcement learning. Finding the most likely examples of a system's failures may highlight flaws that we wish to fix. Adaptive stress testing has the flexibility of treating the simulator as a black box, meaning that access to the simulation state is not needed, allowing AST to be used to validate policies from a wide range of domains. To facilitate the general use of adaptive stress testing for validation, this paper presents a new software package called the AST Toolbox.

Research and Industrial Usage

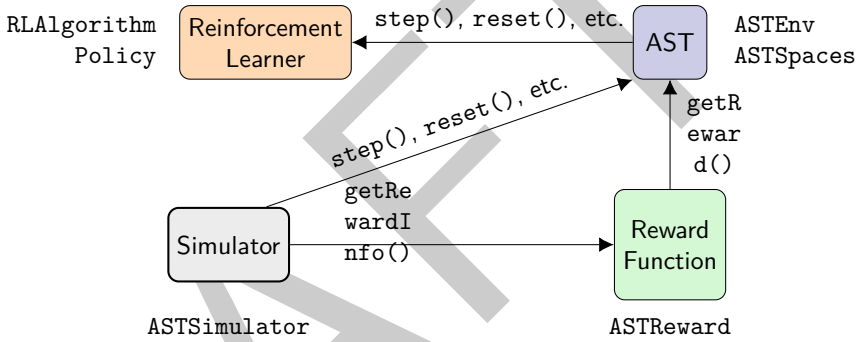
The authors have published multiple papers on AST at venues including the Intelligent Vehicle Symposium (IV), the Intelligent Transportation Systems Conference (ITSC), and the Digital Avionics Systems Conference (DASC). Research vectors include adding new solver algorithms (Koren & Kochenderfer, 2020) , improving failure diversity (Corso et al., 2019), adding interpretability (Corso & Kochenderfer, 2020), and improving scalability (Koren & Kochenderfer, 2019). Applications have included autonomous vehicles (Koren et al., 2018), aircraft collision avoidance software (Lee et al., 2015), aircraft flight management systems (Moss et al., 2020), and image-based neural network controllers (Julian et al., 2020). We have also worked with a range of industrial and government partners, including Nvidia, NASA Ames, Uber ATG, Samsung, and the FAA.

The Journal of Open Source Software

Figures



(a) The AST method. The simulator is treated as a black box. The solver optimizes a reward based on transition likelihood and whether an event has occurred.



(b) The AST Toolbox architecture. `ASTEnv` combines the simulator and reward function in a gym environment. The solver is implemented using the *garage* package.

Figure 1: The AST method and the AST Toolbox architecture.

Acknowledgments

References

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI gym. *arXiv Preprint arXiv:1606.01540*.

Corso, A., Du, P., Driggs-Campbell, K., & Kochenderfer, M. J. (2019). Adaptive stress testing with reward augmentation for autonomous vehicle validation. *IEEE Intelligent Transportation Systems Conference (ITSC)*, 163–168. https://doi.org/10.1109/ITSC.2019.8917242

Corso, A., & Kochenderfer, M. J. (2020). Interpretable safety validation for autonomous vehicles. *arXiv Preprint arXiv:2004.06805*. https://doi.org/10.1109/ITSC45102.2020.9294490

Julian, K. D., Lee, R., & Kochenderfer, M. J. (2020). Validation of image-based neural network controllers through adaptive stress testing. *arXiv Preprint arXiv:2003.02381*. https://doi.org/10.1109/itsc45102.2020.9294549

Kochenderfer, M. J. (2015). *Decision making under uncertainty* (pp. 113–132). MIT Press. https://doi.org/10.7551/mitpress/10187.001.0001

Koren, M., Alsaif, S., Lee, R., & Kochenderfer, M. J. (2018). Adaptive stress testing for autonomous vehicles. *IEEE Intelligent Vehicles Symposium*. https://doi.org/10.1109/

IVS.2018.8500400

Koren, M., Corso, A., & Kochenderfer, M. J. (2020). The adaptive stress testing formulation. *arXiv Preprint arXiv:2004.04293*.

Koren, M., & Kochenderfer, M. J. (2019). Efficient autonomy validation in simulation with adaptive stress testing. *IEEE Intelligent Transportation Systems Conference (ITSC)*, 4178–4183. https://doi.org/10.1109/ITSC.2019.8917403

Koren, M., & Kochenderfer, M. J. (2020). Adaptive stress testing without domain heuristics using go-explore. *arXiv Preprint arXiv:2004.04292*. https://doi.org/10.1109/ITSC45102.2020.9294729

Lee, R., Kochenderfer, M. J., Mengshoel, O. J., Brat, G. P., & Owen, M. P. (2015). Adaptive stress testing of airborne collision avoidance systems. *Digital Avionics Systems Conference (DASC)*. https://doi.org/10.1109/DASC.2015.7311613

Lee, R., Mengshoel, O. J., Saksena, A., Gardner, R. W., Genin, D., Silbermann, J., Owen, M., & Kochenderfer, M. J. (2020). Adaptive stress testing: Finding likely failure events with reinforcement learning. *Journal of Artificial Intelligence Research*, *69*, 1165–1201. https://doi.org/10.1613/jair.1.12190

Moss, R. J., Lee, R., Visser, N., Hochwarth, J., Lopez, J. G., & Kochenderfer, M. J. (2020). Adaptive stress testing of trajectory predictions in flight management systems. *Digital Avionics Systems Conference (DASC)*. https://doi.org/10.1109/DASC50938.2020.9256730

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. *International Conference on Machine Learning (ICML)*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, *abs/1707.06347*. http://arxiv.org/abs/1707.06347

The garage contributors. (2019). Garage: A toolkit for reproducible reinforcement learning research. In *GitHub repository*. https://github.com/rlworkgroup/garage; GitHub.