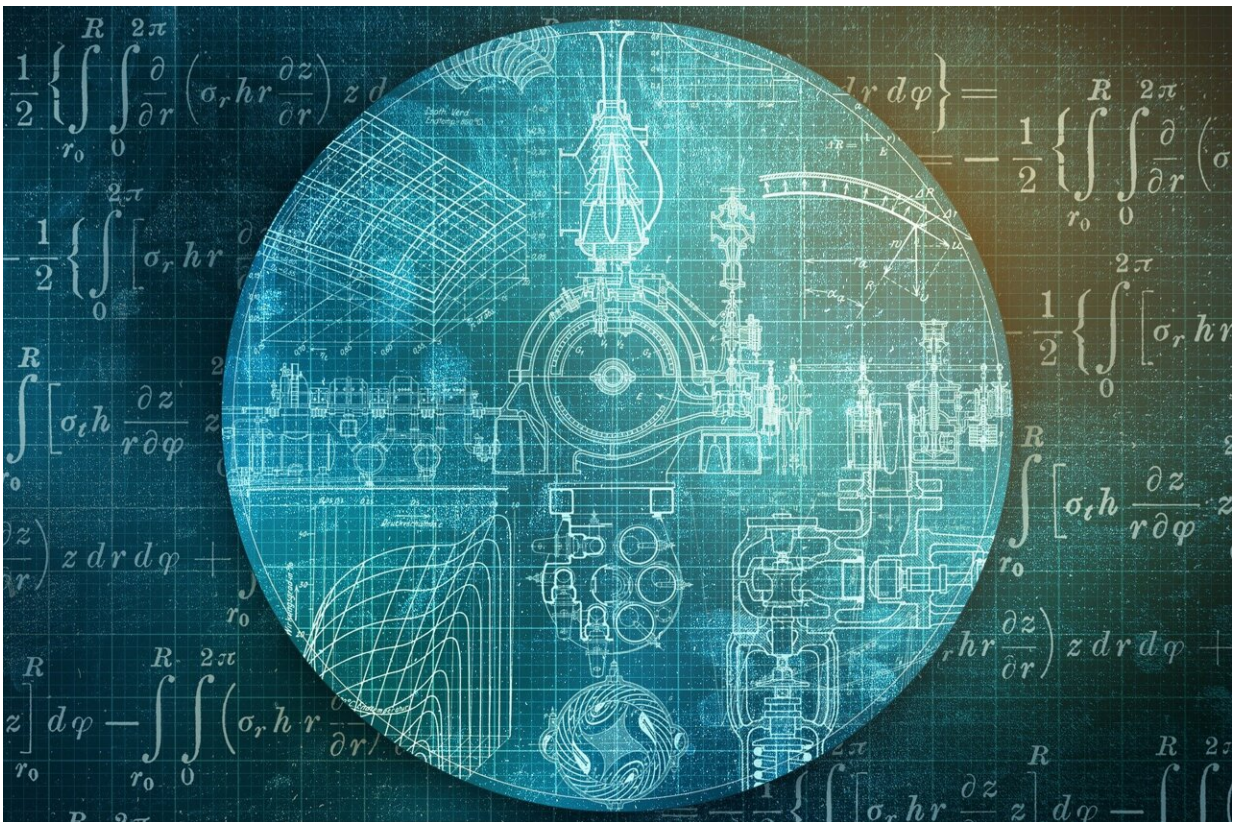


# New algorithm finds failures and fixes in autonomous systems, from drone teams to power grids

November 9 2023, by Jennifer Chu



Credit: Pixabay/CC0 Public Domain

From vehicle collision avoidance to airline scheduling systems to power supply grids, many of the services we rely on are managed by computers.

As these autonomous systems grow in complexity and ubiquity, so too could the ways in which they fail.

Now, MIT engineers have developed an approach that can be paired with any [autonomous system](#), to quickly identify a range of potential failures in that system before they are deployed in the real world. What's more, the approach can find fixes to the failures, and suggest repairs to avoid system breakdowns.

The team has shown that the approach can root out failures in a variety of simulated autonomous systems, including a small and large [power grid](#) network, an aircraft collision avoidance system, a team of rescue drones, and a robotic manipulator. In each of the systems, the new approach, in the form of an automated sampling algorithm, quickly identifies a range of likely failures as well as repairs to avoid those failures.

The new algorithm takes a different tack from other automated searches, which are designed to spot the most severe failures in a system. These approaches, the team says, could miss subtler though significant vulnerabilities that the [new algorithm](#) can catch.

"In reality, there's a whole range of messiness that could happen for these more [complex systems](#)," says Charles Dawson, a graduate student in MIT's Department of Aeronautics and Astronautics. "We want to be able to trust these systems to drive us around, or fly an aircraft, or manage a power grid. It's really important to know their limits and in what cases they're likely to fail."

Dawson and Chuchu Fan, assistant professor of aeronautics and astronautics at MIT, are presenting [their work](#) this week at the [Conference on Robotic Learning](#) in Atlanta.

## Sensitivity over adversaries

In 2021, a major system meltdown in Texas got Fan and Dawson thinking. In February of that year, winter storms rolled through the state, bringing unexpectedly frigid temperatures that set off failures across the power grid. The crisis left more than 4.5 million homes and businesses without power for multiple days. The system-wide breakdown made for the worst energy crisis in Texas' history.

"That was a pretty major failure that made me wonder whether we could have predicted it beforehand," Dawson says. "Could we use our knowledge of the physics of the electricity grid to understand where its weak points could be, and then target upgrades and software fixes to strengthen those vulnerabilities before something catastrophic happened?"

Dawson and Fan's work focuses on robotic systems and finding ways to make them more resilient in their environment. Prompted in part by the Texas power crisis, they set out to expand their scope, to spot and fix failures in other more complex, large-scale autonomous systems. To do so, they realized they would have to shift the conventional approach to finding failures.

Designers often test the safety of autonomous systems by identifying their most likely, most severe failures. They start with a computer simulation of the system that represents its underlying physics and all the variables that might affect the system's behavior. They then run the simulation with a type of algorithm that carries out "adversarial optimization"—an approach that automatically optimizes for the [worst-case scenario](#) by making small changes to the system, over and over, until it can narrow in on those changes that are associated with the most severe failures.

"By condensing all these changes into the most severe or likely failure, you lose a lot of complexity of behaviors that you could see," Dawson

notes. "Instead, we wanted to prioritize identifying a diversity of failures."

To do so, the team took a more "sensitive" approach. They developed an algorithm that automatically generates random changes within a system and assesses the sensitivity, or potential failure of the system, in response to those changes. The more sensitive a system is to a certain change, the more likely that change is associated with a possible failure.

The approach enables the team to route out a wider range of possible failures. By this method, the algorithm also allows researchers to identify fixes by backtracking through the chain of changes that led to a particular failure.

"We recognize there's really a duality to the problem," Fan says. "There are two sides to the coin. If you can predict a failure, you should be able to predict what to do to avoid that failure. Our method is now closing that loop."

## **Hidden failures**

The team tested the new approach on a variety of simulated autonomous systems, including a small and large power grid. In those cases, the researchers paired their algorithm with a simulation of generalized, regional-scale electricity networks. They showed that, while conventional approaches zeroed in on a single power line as the most vulnerable to fail, the team's algorithm found that, if combined with a failure of a second line, a complete blackout could occur.

"Our method can discover hidden correlations in the system," Dawson says. "Because we're doing a better job of exploring the space of failures, we can find all sorts of failures, which sometimes includes even more severe failures than existing methods can find."



The researchers showed similarly diverse results in other [autonomous systems](#), including a simulation of avoiding aircraft collisions, and coordinating rescue drones. To see whether their failure predictions in simulation would bear out in reality, they also demonstrated the approach on a robotic manipulator—a robotic arm that is designed to push and pick up objects.

The team first ran their algorithm on a simulation of a robot that was directed to push a bottle out of the way without knocking it over. When they ran the same scenario in the lab with the actual robot, they found that it failed in the way that the algorithm predicted—for instance, knocking it over or not quite reaching the bottle. When they applied the algorithm's suggested fix, the robot successfully pushed the bottle away.

"This shows that, in reality, this system fails when we predict it will, and succeeds when we expect it to," Dawson says.

In principle, the team's approach could find and fix failures in any autonomous system as long as it comes with an accurate simulation of its behavior. Dawson envisions one day that the approach could be made into an app that designers and engineers can download and apply to tune and tighten their own systems before testing in the real world.

"As we increase the amount that we rely on these automated decision-making systems, I think the flavor of failures is going to shift," Dawson says. "Rather than mechanical failures within a system, we're going to see more failures driven by the interaction of automated decision-making and the physical world. We're trying to account for that shift by identifying different types of failures, and addressing them now."

**More information:** Charles Dawson and Chuchu Fan, A Bayesian approach to breaking things: efficiently predicting and repairing failure modes via sampling. [openreview.net/forum?id=fNLBmtYBiC](https://openreview.net/forum?id=fNLBmtYBiC)

*This story is republished courtesy of MIT News ([web.mit.edu/newsoffice/](http://web.mit.edu/newsoffice/)), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

Citation: New algorithm finds failures and fixes in autonomous systems, from drone teams to power grids (2023, November 9) retrieved 8 May 2024 from <https://techxplore.com/news/2023-11-algorithm-failures-autonomous-drone-teams.html>

<p>This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.</p>
--