Posto: Statistical Safety Verification for Complex Autonomous Systems

A technical overview of a data-driven monitoring tool for modern autonomy.

The Verification Gap in Modern Autonomy

As autonomous systems grow in complexity, traditional verification methods are struggling to keep pace.

Surge in Complexity

The rapid advancement of autonomy is creating systems where formal models are difficult or impossible to create and maintain.

The Black Box Problem

The increasing reliance on components like Deep Neural Networks (DNNs) introduces behaviors that are not easily verifiable with existing model-based techniques.

The Scaling Challenge

Traditional verification and monitoring techniques fail to scale effectively, leaving a critical gap in safety assurance for these complex systems.

Introducing Posto: Monitor Safety with Real-World Data

Posto is a statistical monitoring tool that provides safety guarantees for complex systems without requiring a formal model.

Model-Free

Operates using only the system's Input/Output (I/O) execution model. **No formal system model is required.**

Data-Driven

Analyzes execution logs, even if they are **noisy and incomplete**.

Actionable Outputs

Provides a **probabilistic safety guarantee** with a user-specified confidence level.

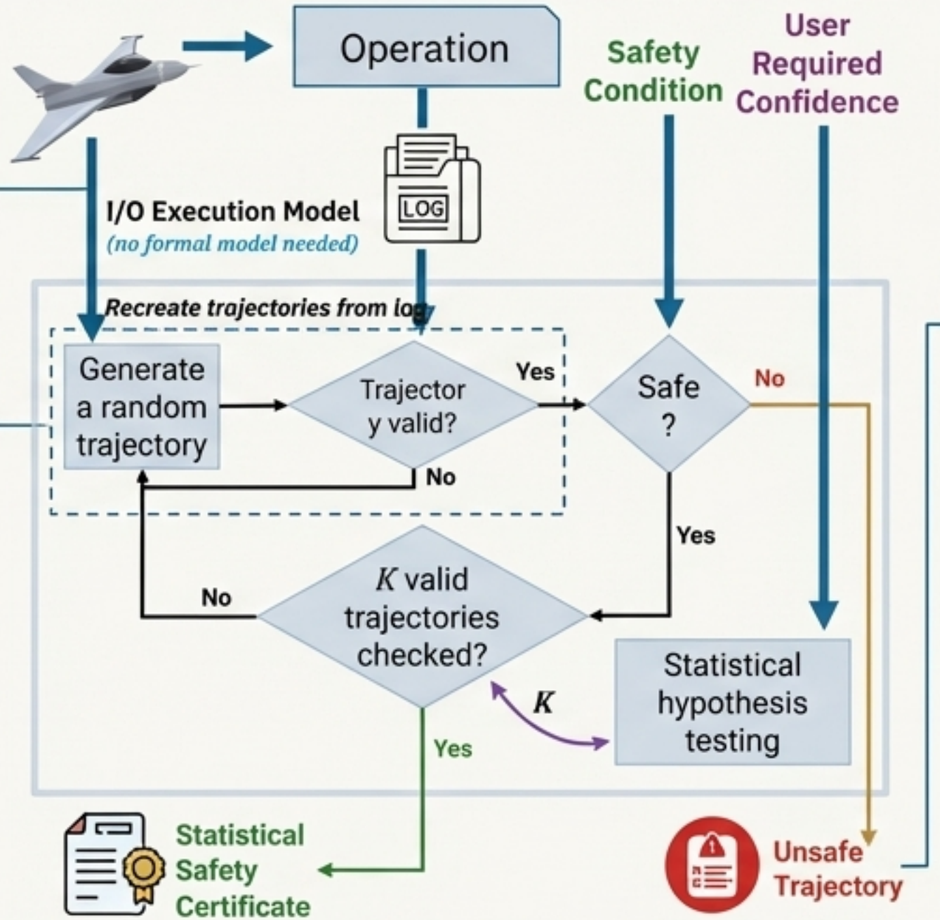
Generates a **concrete counterexample** if an unsafe behavior is discovered.

The Posto Approach: From Log Data to Safety Certificate

Posto ingests system data and user requirements to statistically test for safety and deliver a clear verdict.

1. Inputs:
The process begins with four key inputs: the I/O Execution Model, an Execution Log from the system's operation, the defined Safety Condition, and the User-Required User-Required Confidence level.

2. Verification Loop:
Posto recreates and validates thousands of trajectories from the log. It needs to check a specific number of valid trajectories, K, which is determined by statistical hypothesis testing.



3. Outputs:
The process results in one of two clear outcomes: a formal "Statistical Safety Certificate" if all checked trajectories are safe, or a specific "Unsafe Trajectory" that serves as a counterexample if a violation is found.

Built on Peer-Reviewed Academic Research

Posto’s methodology is grounded in rigorous research from the formal methods community.

Probabilistic Safety Verification of Autonomous Systems: A Statistical Approach for Monitoring

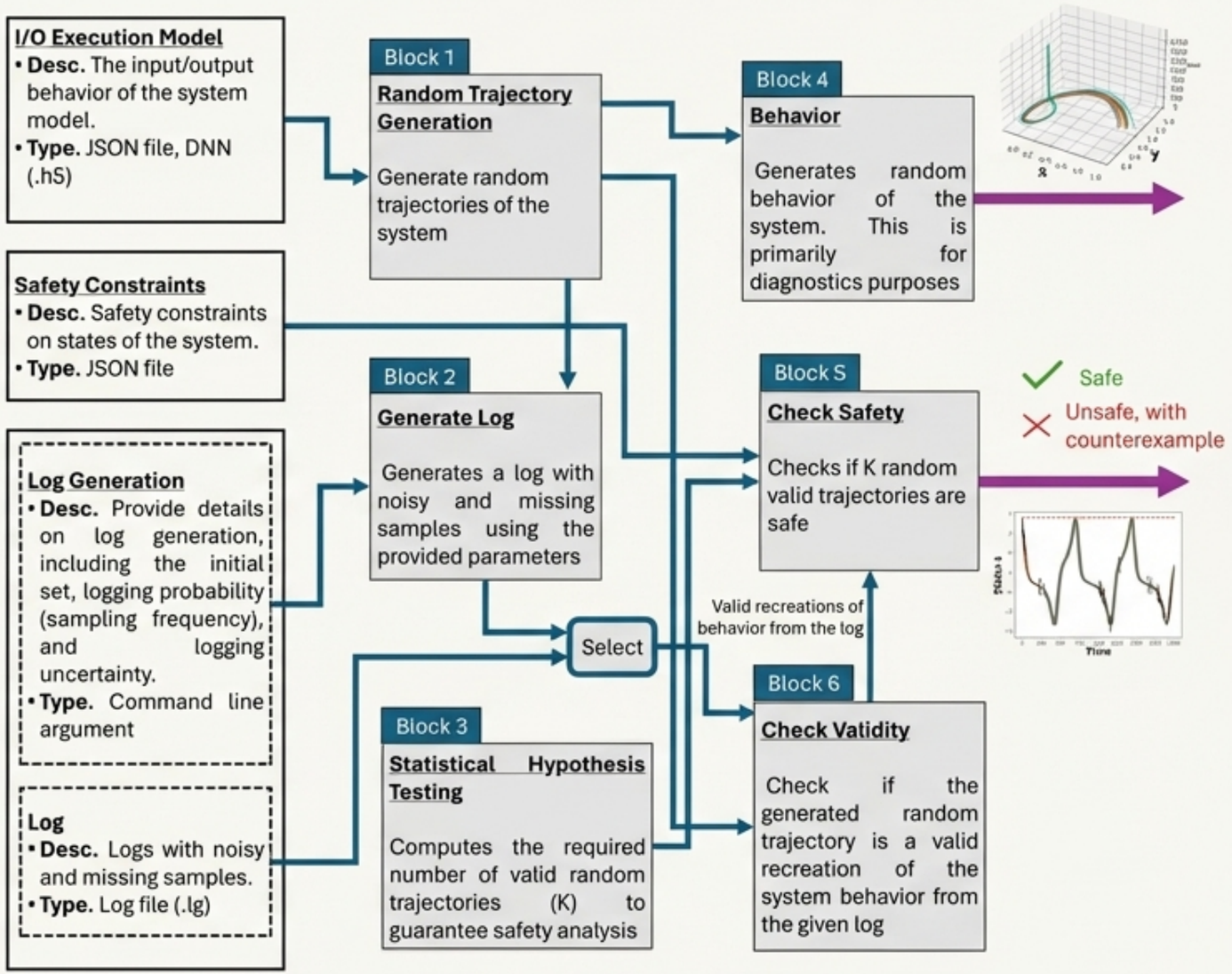
Authors: Bineet Ghosh, Étienne André

Published In: International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE), 2025.

Under the Hood: The Posto Architecture

Posto's modular architecture enables a systematic process from data ingestion and trajectory generation to final safety validation.

1. **Inputs:** The system accepts an I/O Execution Model (e.g., JSON file, DNN), Safety Constraints, and an Execution Log. Log generation parameters can also be provided.
2. **Generation Phase (Blocks 1 & 2):** It generates random trajectories based on the model and can create synthetic logs with noise and missing samples.
3. **Statistical Core (Block 3):** It performs statistical hypothesis testing to compute K, the required number of valid trajectories to guarantee the desired confidence level.
4. **Verification Loop (Blocks 6 & 5):** For each trajectory, it checks if it's a valid recreation from the given log and then checks if it violates the safety constraints.
5. **Outputs:** The process concludes with a binary output: "Safe" or "Unsafe, with counterexample".



Posto's Core Functionalities

Posto provides a suite of functionalities for verification, simulation, and diagnostics.

Statistical Safety Verification

The primary function to provide probabilistic safety guarantees from log data.

Log Generation

Synthetically create realistic logs with noise and missing data to test system robustness.

Behavior Generation & Visualization

Generate and analyze potential system behaviors for diagnostics and deeper understanding.

Function in Focus: Statistical Safety Verification

Posto systematically checks a statistically significant number of trajectories to provide a confident safety assessment.

Hypothesis Testing

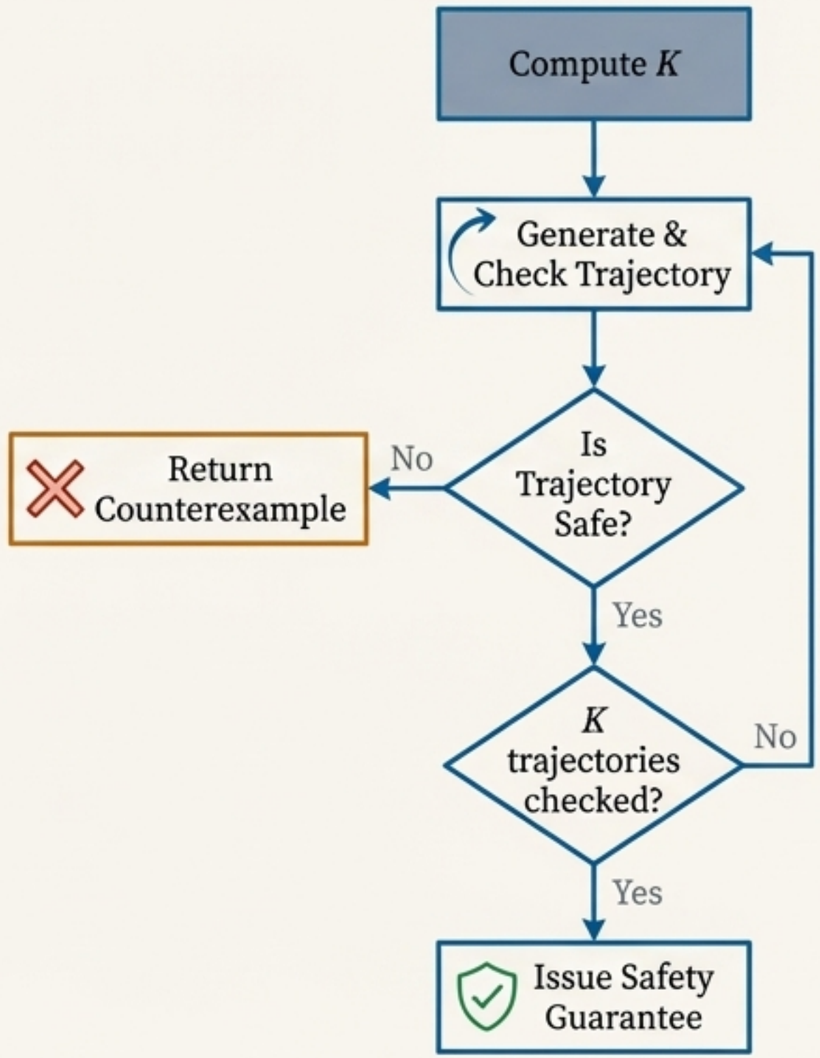
First, Posto computes K, the number of valid trajectories that must be checked to satisfy the user's specified confidence requirement.

Iterative Checking

The tool then generates and validates trajectories one by one. Each valid trajectory is checked against the defined safety property.

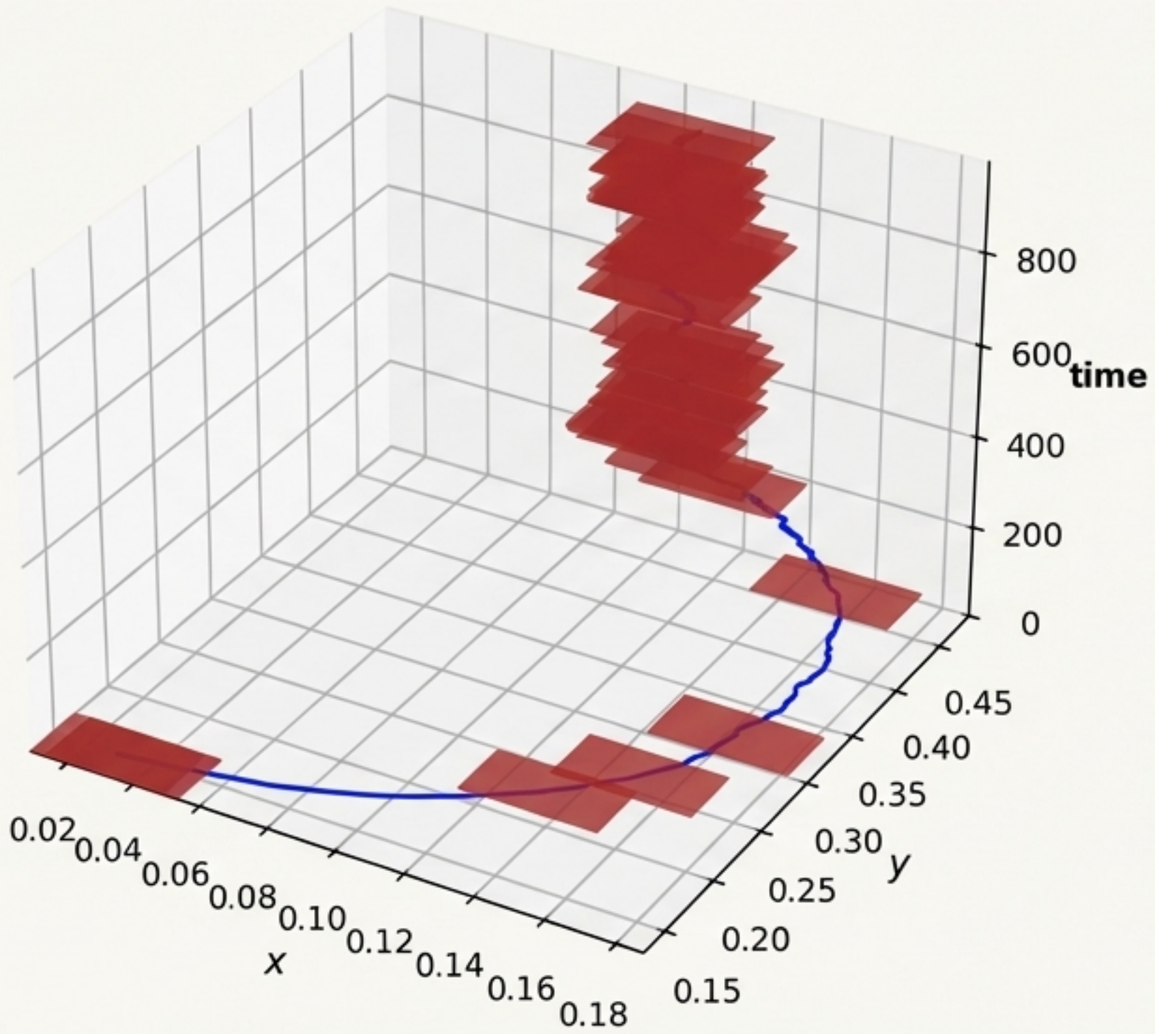
Binary Outcome

- If an unsafe trajectory is found at any point, the process halts immediately and returns it as a **concrete counterexample**.
- If K valid trajectories are checked and all are safe, Posto issues a **probabilistic guarantee of safety.**



Function in Focus: Simulating Real-World Conditions

Posto can synthetically generate logs with noise and missing samples to test system robustness.



Purpose

Allows for rigorous testing and verification even when real-world execution logs are unavailable or insufficient.

Simulating Imperfection

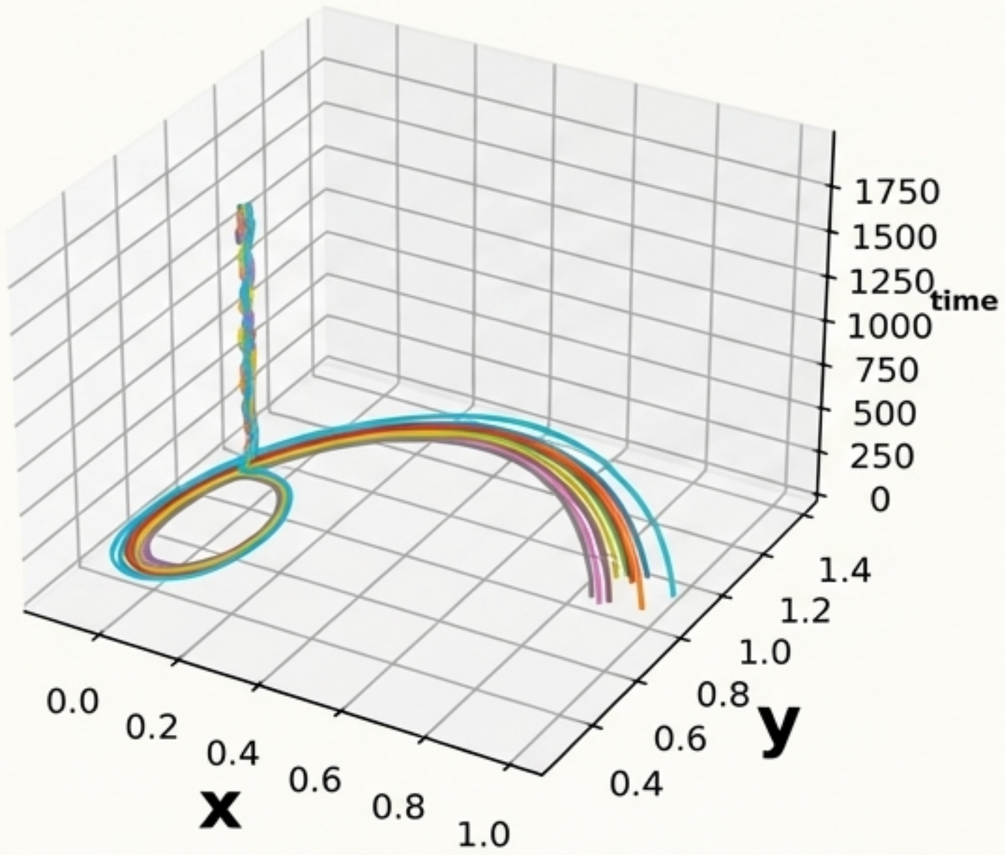
The tool simulates real-world data collection issues by programmatically injecting noise and creating missing samples in the synthetic log.

What the Image Shows

The blue line represents the system's true, continuous trajectory. The red blocks show how this trajectory is captured in a synthetically generated log file—discretely, with uncertainty (larger blocks), and with missing data points.

Function in Focus: Visualizing System Behavior

Generate and visualize random system trajectories to diagnose issues and understand behavior under uncertainty.



Diagnostic Tool

This function helps developers and researchers understand the full range of possible system behaviors allowed by the I/O model.

Robustness Analysis

By visualizing many random trajectories, one can explore how the system behaves under different valid conditions and identify potential edge-case scenarios.

What the Image Shows

Multiple possible system trajectories are generated and plotted over time. This collection of paths illustrates the behavioral envelope of the system.

Why Posto Matters: A New Paradigm for Verification

Posto shifts from traditional formal model-based verification to a more practical, data-driven statistical approach.

Solves a Modern Problem

Directly addresses the verification challenge for complex, AI-driven systems where creating formal models is intractable.

Works with Imperfect Data

Uniquely designed to operate on noisy, incomplete logs—the type of data available from real-world systems.

Provides Actionable Guarantees

Delivers both probabilistic safety certificates for confident deployment and concrete counterexamples for effective debugging.

Get Started with Posto

Posto is an open-source tool with comprehensive documentation to help you begin monitoring your own systems.

Features for Users

Easy to Use

Comes with ready-to-use scripts to reproduce experiments.

Extendable

Designed to be extended for custom applications and monitoring needs.

Resources

Project Repository

[Link to GitHub/Project Page]

Full Documentation

Read the Posto Manual for complete setup, installation, and command options.