BLADE-AV Governance Simulator

Simulation User Guide

Version: v2.2
Document: BLADEAV-SIM-UG-001
Date: April 2026
Revision: 1.0

Author: Burak Oktenli
Institution: Georgetown University, School of Continuing Studies
Program: M.P.S. Applied Intelligence (STEM)
ORCID: 0009-0001-8573-1667

Related Artifact: DOI: 10.5281/zenodo.19232130

Document Control

Document ID	BLADEAV-SIM-UG-001
Simulation Version	v2.2
Author	Burak Oktenli — Georgetown University — ORCID 0009-0001-8573-1667
Date / Revision	April 2026 / Rev 1.0 — Initial Release
Classification	UNCONTROLLED — Research Artifact
Related Artifact	DOI: 10.5281/zenodo.19232130

Table of Contents

1. Purpose, Scope, and Assumptions

This guide provides operating procedures for the BLADE-AV (Autonomous Vehicle Drive-by-Wire Governance) simulation. It implements the governance pipeline for civilian autonomous vehicles with 6 sensor channels (ARS540 radar, OS1-64 LiDAR, GMSL2 camera, ZED-F9R GNSS, SMI230 IMU, C-V2X), 12 attack scenarios (E1-E12), KILOVAC LEV200 electromechanical relay modeling, and three-leg redundant fail-safe circuit simulation.

Intended Audience: EB2-NIW petition evaluators, defense/aerospace reviewers, academic peers, and technical collaborators seeking independent verification of governance pipeline behavior.

Scope: Operation of the BLADE-AV Governance Simulator simulation. Does not cover mathematical theory (see published paper) or hardware specifications (see Blueprint.am).

1.1 Assumptions and Constraints

- The user has a modern browser with JavaScript enabled and HTTPS access.
- All parameter values are synthetic research placeholders, not calibrated against physical hardware.
- The simulation models governance logic only. Physical dynamics (aerodynamics, acoustics, electromagnetics) are simplified.
- Cryptographic operations (SHA-256 audit chain) use the WebCrypto API, not hardware TPM/HSM.
- Results are valid for demonstrating architectural behavior, not for operational safety assessment.

> **IMPORTANT**
> This simulation is a research prototype. Not for operational planning, safety-critical decisions, or system certification. All parameters are synthetic.

2. Quick Start

Step 1. Open blade-av-simulation.html in Chrome over HTTPS.

Step 2. Select attack scenarios using checkboxes E1-E12 or click preset (ALL, SENSOR, CYBER).

Step 3. Click RUN to start, then INJECT to inject the selected attacks.

Step 4. Watch the KILOVAC relay state — when authority drops, the relay OPENS (safe state).

Step 5. Click MC x100 for statistical analysis across 100 automated trials.

> **NOTE**
> All computation runs client-side. No data leaves your browser. Requires HTTPS (not file://).

3. System Requirements and Security Considerations

Browser	Chrome 90+, Firefox 88+, Safari 15+, Edge 90+
Protocol	HTTPS required (WebCrypto API for SHA-256 audit hashing)

Display	Min 1280x720; recommended 1920x1080+
CPU/Memory	Any modern processor. Monte Carlo (100+ runs): multi-core recommended, ~200MB RAM peak
GPU	WebGL-capable recommended for 3D visualizations (Three.js)
Network	Internet for initial CDN load (~500KB). All computation client-side after load.
Installation	None — zero install, no login, no backend, no database, no cookies

3.1 Security Considerations

• **No data exfiltration:** All computation runs in the browser. No data is sent to any server.

• **CDN dependencies:** React, ReactDOM, and Babel load from cdnjs.cloudflare.com (Cloudflare CDN with SRI hashes where available).

• **Audit integrity:** SHA-256 hash chain via WebCrypto API. Each audit entry links to the previous entry's hash. VERIFY button recomputes the entire chain.

• **No authentication:** The simulation has no login system. All state is ephemeral in browser memory.

4. Interface Layout and Navigation

4.1 Panel Layout

The interface uses a compact dashboard layout for autonomous vehicle governance. **Top Panel:** Attack scenario checkboxes (E1-E12) in a grid, preset buttons (ALL, NONE, SENSOR, CYBER), and control buttons (RUN, INJECT, RESET, MC x100, EXPORT). **Center Panel:** 9-module pipeline status visualization showing PASS/FAIL/SKIP per module, sensor trust array (6 channels with weights), authority gauge, and KILOVAC relay state indicator (OPEN/CLOSED). **Bottom Panel:** BYPASS GOVERNANCE toggle with warning label. Speed slider. **Monte Carlo Panel:** Per-scenario statistics table after MC run.

> **TIP**
> Hover over interface elements for tooltips. Most gauges include ARIA labels for screen reader accessibility.

5. Operating Procedures

5.1 Startup

1. Navigate to the simulation URL or click Launch Simulation from burakoktenli.com.
2. Wait for loading (2-5 seconds). CDN scripts load from cdnjs.cloudflare.com.
3. Verify interface loads completely. All panels should be visible.

5.2 Standard Operation

1. Open the simulation via Launch Simulation or navigate to blade-av-simulation.html.

2. Select attack scenarios using the checkboxes (E1-E12) or click preset buttons (ALL, NONE, SENSOR, CYBER).

3. Click RUN to start the simulation. The governance pipeline processes sensor inputs through all 9 modules.

4. Click INJECT to inject the selected attack at the current simulation tick.

5. Observe the sensor trust array: attacked sensors show trust degradation. SATA cross-validation identifies spoofed channels.

6. Monitor the KILOVAC relay state: when authority drops below threshold, the normally-open relay cuts drive-by-wire authority in hardware.

7. Click MC x100 to run 100 Monte Carlo trials. Review per-scenario statistics.

8. Click BYPASS GOVERNANCE to see unprotected behavior. Compare authority trajectories with and without governance.

9. Click EXPORT to download results as JSON for independent analysis.

10. Click RESET to restore all parameters to defaults and clear history.

5.3 Shutdown

1. Export session data. **2.** Close browser tab (all state discarded).

> **IMPORTANT**
> State is not persisted. Export before closing to preserve results.

6. Parameter Reference

Speed	Slider spd	0 - 120 km/h	60	Vehicle speed affecting sensor update rates

> **NOTE**
> All defaults are synthetic. Replace with empirically derived values before operational use.

7. Scenario Reference

E1	GPS Spoofing	Coordinated GNSS manipulation	SATA detects via cross-validation
E2	Camera Blinding	Camera sensor saturated	Camera trust drops, authority reduces
E3	LiDAR Spoofing	Reflected laser attack	LiDAR trust drops, cross-validation triggers
E4	IMU Injection	Accelerometer data manipulation	IMU trust drops, EKF degrades
E5	CAN Bus Injection	Vehicle network message injection	ADARA detects anomalous messages

E6	OTA Firmware	Over-the-air firmware manipulation	Attestation check fails, trust drops
E7	ADARA Adversarial ML	ML-based adversarial perturbation	DPE detects distribution anomaly
E8	FLAME Timeout	Deliberation window exceeded	Circuit breaker transitions to HOLD
E9	Byzantine Fault	Conflicting sensor reports	MAIVA consensus detects conflict
E10	Replay Attack	Replayed sensor data	Nonce/timestamp check detects replay
E11	PCIe Bus Fault	Hardware bus communication failure	Sensor channel drops to zero trust
E12	Sensor Dropout	Complete sensor channel loss	Channel removed from fusion, authority adjusts

8. Metrics, Formulas, and Verification

8.1 Key Metrics

Authority Level

Drive-by-wire authority from 0.0 to 1.0. Below KILOVAC threshold, relay opens and cuts actuator power.

Sensor Trust Array

6 channels: ARS540 Radar (w=0.25), OS1-64 LiDAR (w=0.25), GMSL2 Camera (w=0.15), ZED-F9R GNSS (w=0.15), SMI230 IMU (w=0.10), C-V2X (w=0.10).

Relay State

KILOVAC LEV200 normally-open relay. CLOSED = drive-by-wire active. OPEN = actuator power cut (safe state). 25ms actuation delay modeled.

Watchdog Status

Dual MAX16161 watchdog timers. Both must be fed within timeout window or relay opens.

Pipeline Status

9-module pass/fail: SATA, ADARA, IFF, HMAA, MAIVA, FLAME, CARA, BDA, EFFECTOR.

8.2 Verification Checklist

Perform the following checks to verify correct simulation behavior:

Start simulation (RUN/START)	Interface loads. Governance pipeline begins processing.
Observe default state	Authority at nominal level. All pipeline stages PASS.
Inject a fault or attack	Authority reduces proportionally. Affected stage shows FAIL.
Monitor recovery	If CARA active, observe GREP recovery phases.
Export session data (JSON)	File downloads with parameters, history, and audit trail.
Reload and verify reproducibility	Same seed + params = identical outputs.

9. Data Export and Reproducibility

Click export/download to save session JSON with parameters, history, and audit trail.

Verification: 1) Export JSON. 2) Note PRNG seed. 3) Reload with same seed/params. 4) Verify bit-exact match.

9.1 Reproducibility Guarantee

PRNG	Mulberry32 (32-bit seeded)
Math.random()	Zero calls in computation paths
Cross-Browser	Verified: Chrome, Firefox, Safari, Edge
Cross-Platform	Verified: Windows, macOS, Linux
Audit Chain	SHA-256 via WebCrypto (SubtleCrypto API)

10. Limitations and Threat Considerations

Simulation-Only Evidence	Browser-based computation. No physical sensor data or hardware measurements.
Uncalibrated Parameters	All values are synthetic research parameters, not empirically derived.
No Real-Time Guarantees	JavaScript engine provides no timing guarantees for safety-critical operations.
Simulated Cryptography	SHA-256 uses WebCrypto. TPM/HSM operations are modeled, not hardware-backed.
Single-Session State	All state held in memory. Closing the tab discards all data.

10.1 Threat Considerations

• **CDN compromise:** React/Babel load from cdnjs.cloudflare.com. A CDN compromise could inject malicious code. Mitigation: Subresource Integrity (SRI) hashes on script tags where available.

• **Browser extensions:** Malicious extensions could modify simulation DOM/state. Mitigation: test in Incognito mode for clean results.

• **Local modification:** Users can modify simulation code via DevTools. Exported data should be verified against the published source on burakoktenli.com.

11. Troubleshooting

Black screen after loading	React render error or CSP violation	Open F12 Console for error details. Try Chrome Incognito mode.

Simulation runs slowly	CPU-intensive Monte Carlo or 3D rendering	Close other browser tabs. Reduce sample count.
Controls not responding	Browser tab lost focus	Click inside the simulation window. Ensure tab is active.
Export button not working	Pop-up/download blocked	Allow downloads from the simulation domain in browser settings.
Loading screen never completes	CDN scripts blocked by firewall/extension	Disable ad blockers. Allow cdnjs.cloudflare.com.

12. Glossary and References

12.1 Glossary

BLADE-AV	BLADE-AV (Autonomous Vehicle Drive-by-Wire Governance)
SATA	Sensor Attestation and Trust Anchoring — trust fusion
HMAA	Human-Machine Authority Architecture — authority computation
CARA	Control Authority Regulation Architecture — recovery protocol
Authority Level	Computed governance authority (0.0-1.0) governing operational actions
PRNG	Pseudo-Random Number Generator — Mulberry32 seeded for reproducibility
Governance Pipeline	Sequential processing chain: SATA -> HMAA -> MAIVA -> FLAME -> CARA

12.2 References

[1] Oktenli, B. (2026). BLADE-AV Governance Simulator. DOI: 10.5281/zenodo.19232130.

[2] Oktenli, B. (2026). Simulation artifact. https://burakoktenli.com/blade-av-simulation.html

[3] Oktenli, B. (2026). Research Portfolio. https://burakoktenli.com

[4] U.S. DoD (2023). Directive 3000.09: Autonomy in Weapon Systems.

[5] NIST (2023). AI Risk Management Framework (AI RMF 1.0).

12.3 Contact

Burak Oktenli — Georgetown University, School of Continuing Studies
Website: **burakoktenli.com** | ORCID: **0009-0001-8573-1667**
For questions about this simulation or the governance architecture research program, use the contact form at burakoktenli.com.

End of Document