# STRL — Sensor Truth Replay Lab

A Deterministic Dataflow Framework for Multi-Sensor Validation

**Jishnu Periya**

2026

## Abstract

Modern robotic, autonomous, and aerospace systems rely on multi-sensor perception pipelines operating under strict timing, synchronization, and robustness constraints. In production deployments, failures frequently arise not from algorithmic limitations, but from latency accumulation, timestamp misalignment, calibration drift, and unobservable system interactions.

STRL (Sensor Truth Replay Lab) is a lightweight, deterministic C++ framework designed to replay recorded multi-sensor data and enable controlled experiments on perception pipelines. STRL emphasizes reproducibility, traceability, and validation over end-to-end autonomy, allowing engineers to analyze system behavior under realistic constraints.

## 1 Motivation

Production systems face persistent challenges:

- Sensors operate on independent, drifting clocks
- Synchronization is approximate
- Calibration degrades over time
- Latency budgets are silently violated
- Failures are difficult to reproduce post-deployment

Existing tools—simulation environments, autonomy frameworks, and ML benchmarks—prioritize capability and realism, but provide limited support for deterministic replay and system-level validation. STRL addresses this gap.

## 2 Design Principles

STRL is guided by the following principles:

- **Determinism over realism**: Reproducibility is prioritized over photorealistic simulation.
- **Validation-first**: The framework exists to analyze and explain failures.
- **Minimalism**: Only components required to expose system behavior are included.
- **Explicit time modeling**: Sensor time and system time are first-class concepts.
- **Domain-agnostic**: Applicable to robotics, autonomous systems, and aerospace.

## 3 Replay as Controlled Experimentation

In STRL, replay is defined as deterministic re-execution of recorded sensor data under controlled, parameterized conditions. Replay is an experimental mechanism, not passive playback.

Controlled experiments include:

- Injecting latency, jitter, and frame drops

- Perturbing intrinsic and extrinsic calibration
- Modifying buffering and scheduling strategies
- Comparing pipeline implementations under identical inputs

This approach enables isolation of failure causes and regression analysis.

## 4 Architecture Overview

STRL follows a modular, ADTF-inspired dataflow architecture:

- **Ingestion**: Deterministic loading of recorded sensor data
- **Time & Synchronization**: Clock models, alignment, and jitter injection
- **Calibration**: Application and perturbation of sensor geometry
- **Processing**: Minimal vision and projection stages
- **Validation**: Latency, alignment, and throughput metrics
- **Replay & Analysis**: Deterministic re-execution and comparison

Components communicate via strongly typed, timestamped data streams with explicit ownership and execution order.

## 5 Time Model

Time is treated as an explicit, controllable entity:

- Separation of sensor time and system time
- Explicit clock domains
- Deterministic scheduling
- Configurable synchronization strategies

This enables repeatable experiments that are infeasible in live systems.

## 6 Validation Metrics

STRL focuses on behavioral and system-level metrics:

- End-to-end and per-stage latency
- Synchronization error
- Calibration sensitivity
- Throughput degradation
- Failure frequency under perturbation

Metrics are logged with sufficient context to support root-cause analysis.

## 7 Applications

STRL supports:

- Robotics perception validation
- Autonomous sensor fusion analysis
- Aerospace sensor replay and timing validation
- Generic real-time C++ dataflow experimentation

## 8 Conclusion

STRL externalizes a class of deterministic replay and validation tooling that typically exists only as proprietary, fragmented infrastructure within mature organizations. By prioritizing explicit time modeling, controlled experimentation, and reproducibility, STRL provides a reusable foundation for analyzing and validating multi-sensor perception systems under real-world constraints.

*STRL is designed as a validation laboratory, not an autonomy framework.*