Kinetic: A Precision Stage Control and Error-Modeling Laboratory - An Evidence-Aware Architecture, Evaluation Plan, and Integration Roadmap

Aaron Singh

Abstract **- Kinetic is a deterministic motion-stage simulator and controller bench for backlash, hysteresis, repeatability error, vibration, saturation, travel limits, and settling behavior. It isolates motion control from optical tasks so controllers and safety invariants can be tested before a physical stage is connected. This paper defines the system model, evidence contract, risks, and staged validation plan. No physical-system or performance result is claimed without a committed reproducibility artifact.**

Index Terms - precision motion, stage control, backlash, model predictive control, safety interlocks

I. INTRODUCTION

Optical alignment and inspection depend on motion whose commanded and realized positions are not identical. Direction reversal, load, encoder error, vibration, and settling can dominate the final result. Kinetic exposes these effects as configurable models with deterministic traces and explicit safe-state behavior.

Research contribution

The contribution is not a performance claim. It is a reproducible reference architecture, a falsifiable evaluation protocol, and an evidence schema that records inputs, decisions, outputs, failure modes, and claim boundaries.

II. TECHNICAL CONTEXT

Precision motion is governed by the gap between commanded and realized position. Kinetic isolates that gap so backlash, hysteresis, encoder noise, saturation, vibration, and settling can be varied independently before any optical task or physical stage is introduced.

III. RESEARCH QUESTIONS

1 Which compensation model best corrects direction-dependent backlash?

2 How do settling constraints trade speed for positioning accuracy?

3 Can controller or encoder faults be detected before travel limits are exceeded?

IV. SYSTEM MODEL AND ARCHITECTURE

Commands pass through policy limits, trajectory generation, actuator dynamics, and disturbance models. A simulated encoder returns noisy position estimates to PID or model-predictive controllers. An interlock layer can reject commands or transition the system to a safe stopped state.

$$x(t+1) = f(x(t), u(t), d(t)); \text{ measured position } y(t) = x(t) + \text{encoder error,}$$

with travel, velocity, and acceleration constraints.

The model is a design and test abstraction. It establishes invariants and expected-value checks; it does not convert simulated variables into measured hardware behavior.

V. DETAILED OPERATION AND RATIONALE

1 Implement reference axis dynamics and legal motion-state transitions.

2 Add backlash, hysteresis, noise, saturation, vibration, and settling models.

3 Compare PID, feed-forward compensation, and model-predictive control.

4 Inject encoder, limit, and controller faults while checking safety invariants.

VI. DATA AND EVIDENCE CONTRACT

Initial evidence consists of synthetic command, position, encoder, and fault traces plus simulated calibration grids. Future hardware step-response captures remain a separate evidence class tied to identified devices and calibration records.

Each run must preserve a machine-readable configuration, random seed where applicable, source revision, environment description, raw result, summary method, and known limitations. Derived plots must point back to the exact records used to produce them.

VII. IMPLEMENTATION BLUEPRINT

Trajectory generator. Produces bounded position, velocity, and acceleration commands. Its acceptance artifact is command trace and limit checks.

Axis dynamics. Models inertia, saturation, backlash, and hysteresis. Its acceptance artifact is deterministic realized-position trace.

Controller suite. Compares PID, feed-forward, and predictive control. Its acceptance artifact is error and control-effort series.

Safety supervisor. Rejects illegal motion and latches faults. Its acceptance artifact is interlock state and recovery log.

VIII. EXPERIMENTAL METHODOLOGY

Evaluation proceeds in four gates. Gate 1 checks deterministic behavior and invariants. Gate 2 applies controlled noise, tolerance, and injected-failure sweeps. Gate 3 repeats measurements and reports distributions rather than a favorable single run. Gate 4 introduces an external data source or bounded lab adapter only after the first three gates are reviewable.

Primary metrics are reported with sample counts, aggregation rules, uncertainty, and failure cases.

IX. REPRODUCIBILITY AND ACCEPTANCE CRITERIA

A result is accepted only when another clean environment can reproduce the reference behavior from a named revision. Every comparison uses identical budgets and held-out conditions. Failed runs remain part of the evidence record. Unknown hardware, calibration, and production conditions are labeled unknown rather than assigned synthetic values.

X. FAULT INJECTION AND NEGATIVE TESTING

Negative testing is part of the main evaluation rather than an appendix. Each risk is converted into an explicit scenario with an expected detection, rejection, abstention, or safe-state outcome. A silent success

under missing or contradictory state is treated as a failed test.

Model mismatch. The failure consequence is simulation tuning fails on hardware. The required control is parameter sweeps and identification.

Integrator windup. The failure consequence is overshoot after saturation. The required control is anti-windup tests.

Encoder fault. The failure consequence is controller drives from false position. The required control is plausibility checks and safe stop.

Limit failure. The failure consequence is travel envelope is exceeded. The required control is independent hard-limit model.

XI. ANALYSIS AND INTERPRETATION

Analysis reports distributions, failure counts, and representative traces rather than a single favorable score. Baselines share the same inputs, budgets, stopping rules, and held-out conditions. When multiple configurations are explored, the selection procedure is recorded so the test set is not used as a tuning signal.

The primary metric families are Absolute position error, Repeatability, Overshoot and settling time, Trajectory duration, Limit and interlock violations. Each is reported with its unit, aggregation rule, sample count, spread, and failed or undefined scenarios. An improvement is meaningful only when reference correctness is preserved and the result repeats across declared evaluation seeds or groups.

XII. INTEGRATION INTERFACES

Integration occurs through versioned records rather than hidden shared state. Inputs declare units, coordinate frames or topology identifiers, timestamps, validity windows, and provenance. Outputs include a status, result payload, limitation flags, and a stable link to the run configuration.

Adapters are introduced in read-only or simulated form first. A live adapter cannot inherit the trust assigned to the reference model; it requires separate authentication, calibration, timeout, retry, and safe-state evidence.

XIII. EXTERNAL VALIDITY AND RELEASE CRITERIA

The initial conclusions are bounded by the declared simulator, public data, protocol mocks, and sampled conditions. They do not automatically generalize to other equipment, optical assemblies, wafers, stages, power systems, operating ranges, or production environments. External validity is evaluated as a new experiment, not assumed from architectural similarity.

A portfolio release requires the paper source, generated PDF, configuration schema, reproducible entry point, tests, raw or referenced evidence, and a concise limitations record. If one of these artifacts is missing, the project may remain a research design but cannot be presented as an evidence-backed implementation.

XIV. CLAIM BOUNDARIES, SAFETY, AND RISKS

No physical stage accuracy, repeatability, bandwidth, or safety certification is claimed. The simulator is not a replacement for manufacturer limits, electrical interlocks, or an emergency-stop design.

The primary research risk is a category error: treating simulation, public benchmark data, or protocol mocks as proof of physical-system performance. The mitigation is provenance at every boundary, fail-closed behavior for missing state, and an explicit review before any live interface is enabled.

XV. GOALS, MILESTONES, AND INTEGRATION

1 Implement deterministic single-axis dynamics.

2 Add non-ideal actuator and encoder models.

3 Establish PID and compensation baselines.

4 Integrate the model with Aperture.

5 Compare bounded simulation against a documented lab stage.

XVI. CONCLUSION

Kinetic now has a research plan comparable in rigor and presentation to the established portfolio papers. Its architecture is inspectable, its evaluation is falsifiable, and its limitations are explicit. Credibility will come from completing the staged evidence artifacts, not from promoting unverified numbers.

REFERENCES

[1] HS Kempten, 'relign: Python software for simulating the active alignment of optical systems,' GitHub.

[2] Aaron Singh, 'Kinetic research specification,' local portfolio repository, 2026.

TABLE I
SYSTEM COMPONENTS AND EVIDENCE RESPONSIBILITIES

Component	Responsibility	Required evidence
Trajectory generator	Produces bounded position, velocity, and acceleration commands	Command trace and limit checks
Axis dynamics	Models inertia, saturation, backlash, and hysteresis	Deterministic realized-position trace
Controller suite	Compares PID, feed-forward, and predictive control	Error and control-effort series
Safety supervisor	Rejects illegal motion and latches faults	Interlock state and recovery log

TABLE II
PLANNED EVALUATION MATRIX

Metric family	Acceptance record
Absolute position error	Repeated result with configuration, seed, revision, and limitations
Repeatability	Repeated result with configuration, seed, revision, and limitations
Overshoot and settling time	Repeated result with configuration, seed, revision, and limitations
Trajectory duration	Repeated result with configuration, seed, revision, and limitations
Limit and interlock violations	Repeated result with configuration, seed, revision, and limitations

TABLE III
RISK REGISTER AND CONTROLS

Risk	Failure consequence	Control
Model mismatch	Simulation tuning fails on hardware	Parameter sweeps and identification
Integrator windup	Overshoot after saturation	Anti-windup tests
Encoder fault	Controller drives from false position	Plausibility checks and safe stop
Limit failure	Travel envelope is exceeded	Independent hard-limit model

TABLE IV
PROJECT GOALS AND REQUIRED PROOF

ID	Goal	Completion evidence
G1	Implement deterministic single-axis dynamics.	Reviewable source, test, trace, or measurement artifact
G2	Add non-ideal actuator and encoder models.	Reviewable source, test, trace, or measurement artifact
G3	Establish PID and compensation baselines.	Reviewable source, test, trace, or measurement artifact
G4	Integrate the model with Aperture.	Reviewable source, test, trace, or measurement artifact
G5	Compare bounded simulation against a documented lab stage.	Reviewable source, test, trace, or measurement artifact