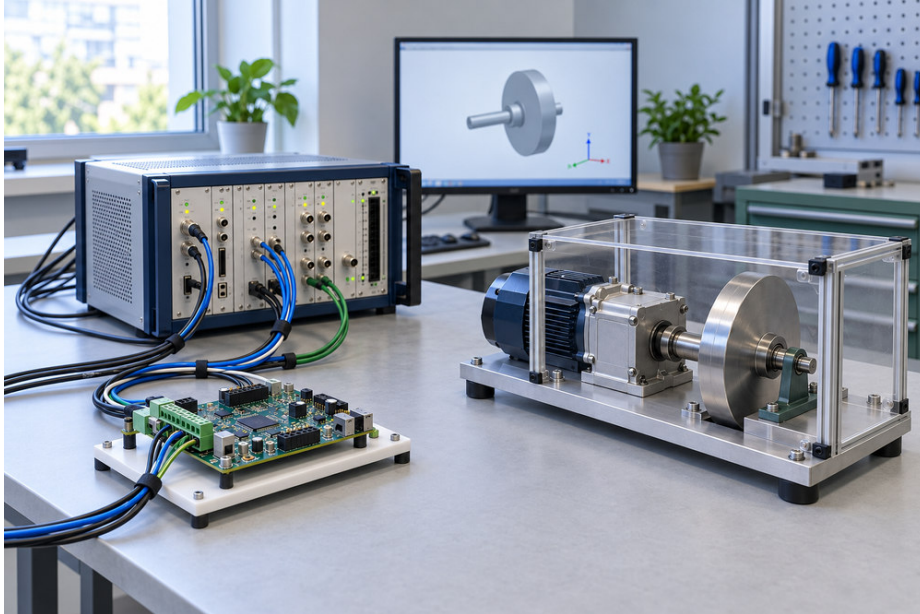


Your Controller Passed HIL. Did the Model Hide the Fault?

The controller is real. Its world may be too kind.



The controller receives a simulated world; the separate mechanism supplies the comparison that gives that world meaning.

A positioning axis makes a tiny correction, pauses, then reverses. On the simulator, it follows smoothly. On the assembled mechanism, nothing happens at first; then it jumps. The controller board, firmware and command can be identical in both tests. The missing ingredient may be the model's description of contact and friction.

Hardware-in-the-loop, or HIL, connects real controller hardware to a simulation that supplies the signals a mechanism would produce. The simulated mechanism is often called the plant. The useful question is specific: can this combination reproduce the behaviour this test is intended to expose?

Friction shows why that matters. A viscous-friction model resists motion in proportion to speed; at rest, its resistance is zero. Static friction can balance an applied torque while the shaft remains still. Movement begins only when that balance

can no longer hold. After movement starts, the resisting torque can change, creating a difficult transition for a controller making very small corrections.

Research on [servo systems with friction and backlash](#) treats these nonlinear effects explicitly. “Nonlinear” here means that doubling the command need not simply double the response. A model fitted to ordinary continuous movement may miss the transition from standing still to moving.

A passing software test and a credible mechanism model are separate achievements. [NASA's model standard](#) distinguishes checking that a simulation implements its design from comparing its results with the system it represents. Faster hardware and more repeated runs do not repair an omitted physical effect.

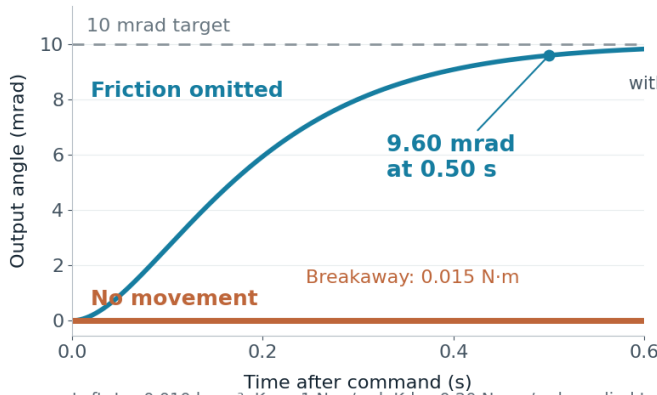
A real controller can pass a test against an unrealistically easy mechanism.

Why the axis stays stuck

A smooth simulated move can hide a stuck axis

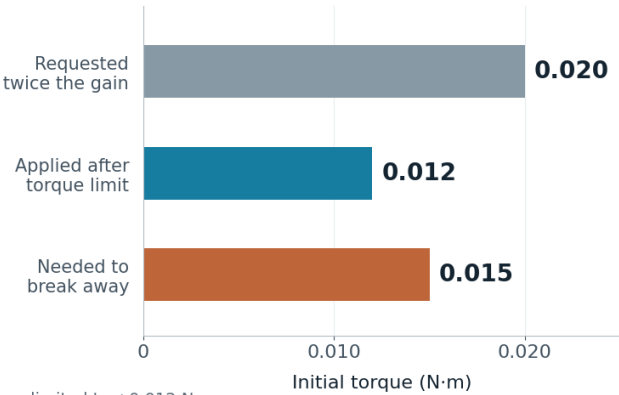
Calculated example · Zero initial speed · No external load torque

Same loop, different plant model



Right: K_p doubles to $2 \text{ N}\cdot\text{m}/\text{rad}$ at the same 10 mrad error. The $0.012 \text{ N}\cdot\text{m}$ limit remains below breakaway.

More gain cannot lift a hard limit



A drive capped at $0.012 \text{ N}\cdot\text{m}$ cannot overcome the assumed $0.015 \text{ N}\cdot\text{m}$ breakaway torque.

Consider a calculated rotary-axis example, starting from rest with no external load torque. Let its inertia J be $0.010 \text{ kg}\cdot\text{m}^2$ and its requested angle r be 0.010 rad , or 10 mrad . One milliradian is one thousandth of a radian.

Use a simple proportional-plus-damping controller:

$$\tau_{\text{request}} = K_p(r - \theta) - K_d\omega$$

Here θ is angle, ω is angular speed, $K_p = 1 \text{ N}\cdot\text{m}/\text{rad}$ and $K_d = 0.20 \text{ N}\cdot\text{m}\cdot\text{s}/\text{rad}$. The drive clips the applied torque to $\pm 0.012 \text{ N}\cdot\text{m}$. This clipping is saturation: asking for more does not produce more.

With friction omitted, acceleration follows $J\ddot{\theta} = \tau$. These choices give the smooth response $\theta = r[1 - (1 + 10t)e^{-10t}]$, with t in seconds. At 0.50 s , the calculated angle is 9.60 mrad . The peak torque magnitude is $0.010 \text{ N}\cdot\text{m}$, so this movement never reaches the limit.

Now include a hard static-friction rule with a breakaway torque of $0.015 \text{ N}\cdot\text{m}$. The initial $0.010 \text{ N}\cdot\text{m}$ request is balanced by friction.

The shaft stays at zero, leaving the full 10 mrad error.

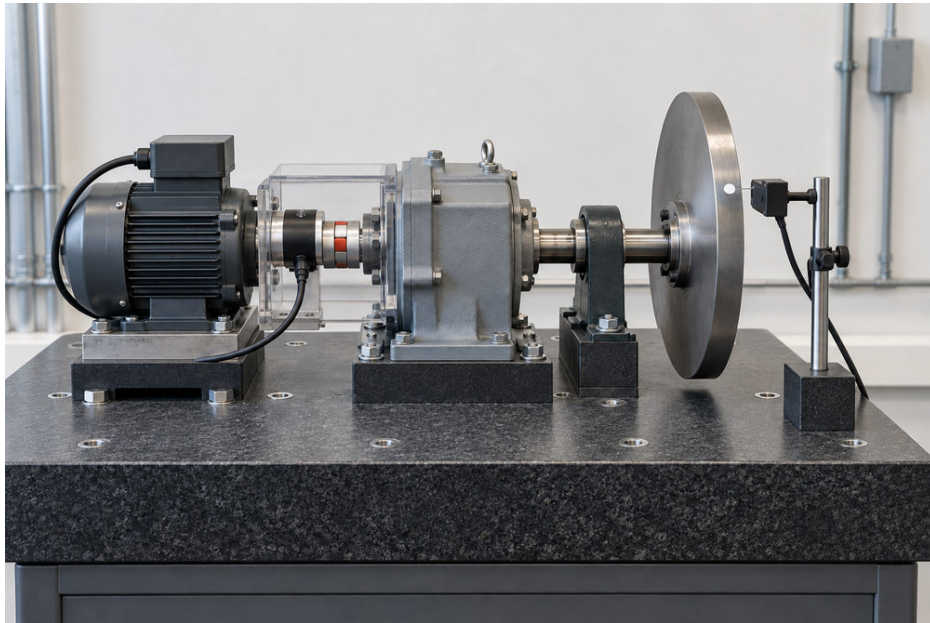
Double K_p . The initial request becomes $0.020 \text{ N}\cdot\text{m}$, but the drive still applies only $0.012 \text{ N}\cdot\text{m}$ - below breakaway. Increasing this gain cannot start the axis under these assumptions.

Adding integral action cannot create unavailable torque either. It can accumulate an increasingly large internal request while the output stays capped. When conditions change, that stored request can delay recovery; [MIT's anti-windup notes](#) explain the mechanism. Anti-windup limits that accumulation, but does not increase actuator capacity.

These chosen values isolate one failure mechanism. They describe no particular Inventra S³ axis, and omit electrical dynamics, flexible parts and digital sampling.

A torque request above breakaway is useless if the applied torque remains below it.

A reversal reveals what a long sweep can miss



Motor-side sensing and a separate output marker expose motion on opposite sides of the transmission.

Backlash creates another kind of waiting. When direction reverses, clearance between mating parts may need to be crossed before torque reaches the load through the opposite contact. During that interval, the motor can move while the output remains still. [MIT's motor-servo example](#) models motor and load separately, with no transmitted contact torque inside the clearance band.

Suppose the full gap, expressed at the output, is 0.8 mrad. If the output is held still while the equivalent motor angle crosses it at 4 mrad/s, traversing the gap takes 0.20 s. This is a geometric calculation, not a predicted system delay. Actual contact stiffness, loading and acceleration determine what happens during a real reversal. A sensor on the motor alone cannot establish the output's movement through that interval.

The useful comparison therefore includes an independent observation of output motion. Run a slow forward approach, a pause and a small reverse command. Record motor position, output position and applied current

or torque where measurable. Keep the payload, orientation and temperature known. A large one-direction sweep is a poor substitute: it may keep the same gear faces engaged throughout.

Use some movements to estimate model parameters, then reserve different movements for checking predictions. For example, estimate friction from repeated slow starts, then predict an untried small reversal without retuning first. Compare the start of output motion and the remaining position error, not just a graph's overall appearance.

If the motor moves first, examine clearance and compliance. If both remain still while torque rises, examine breakaway and the drive limit. These are hypotheses to test, not diagnoses from one trace. Vary uncertain parameters over defensible measured ranges; changing only their nominal values can hide the condition that matters.

Measure the output independently when the model includes drivetrain clearance.

Change the mechanism model; corrupt the feedback separately

A friction sweep asks whether the controller tolerates a plausible mechanism. A frozen-feedback test asks what it does when its evidence becomes wrong. Keep those questions separate initially, so a failure has a traceable cause.

In HIL, continue calculating the plant's true position while holding the reported position constant. Keep other interface behaviour normal. Now the controller can receive apparently healthy communication carrying a stale value. This is different from unplugging a cable. [NASA's software-fault bulletin](#) stresses testing erroneous behaviour and unexpected sensor input, alongside sensor HIL tests.

Record the simulated true angle, delivered feedback, requested torque, applied torque and controller fault state together. A stationary reported angle cannot itself prove a fault - the shaft might actually be stationary. Detection needs the system's specified plausibility checks or independent evidence. Set the expected response from the design requirements: which condition triggers a fault, what output is permitted afterwards, and how recovery occurs.

The simulator needs checking too. Hold the controller's sample period fixed while reducing the plant solver's integration step offline. If the predicted start of movement or reversal changes materially, investigate numerical error before accepting the result. Then confirm that the selected real-time implementation completes its calculations and delivers feedback on schedule. An accurate offline model is insufficient if the live simulator misses its deadlines.

Also inspect the friction equation near zero speed. A [smooth friction approximation](#) can be convenient numerically yet permit slow creep without a separate sticking rule, where hard static friction predicts no movement. Its suitability depends on the question being tested.

[Inventra S³ lists HIL and microprocessor control verification among its analysis capabilities](#). For a proposed motion test, ask one concrete question: which measured behaviour shows that this model can reveal the failure we are trying to catch?

Model uncertainty, sensor faults and simulator errors need different evidence.

Sources and further reading

[Standard for Models and Simulations, NASA-STD-7009B](#)

NASA | 2024

[Characteristic Modeling and Control of Servo Systems with Backlash and Friction](#)

Nanjing University of Science and Technology; Mathematical Problems in Engineering | 2014

[16.30 Topic 23: Anti-windup](#)

Massachusetts Institute of Technology | 2010

[Motor Servo with Backlash](#)

Massachusetts Institute of Technology | 2009

[Considerations for Software Fault Prevention and Tolerance, Technical Bulletin 23-06](#)

NASA Engineering and Safety Center | 2023

[Analysis](#)

Inventra Inc.

[Control software verified before the prototype: model-based design and HIL](#)

Inventra Inc. | 2026

[Read this article online](#)

Original text and artwork: Inventra Inc. Referenced sources retain their rights.