Neural Control: Adjoint Learning Through Equilibrium Constraints

Forty-Third International Conference on Machine Learning (ICML 2026) · Seoul, South Korea · July 2026

Dezhong Tong, Jiawen Wang, Hengyi Zhou, Yinglong Shen, Xiaonan Huang, M. Khalid Jawed
University of Michigan & University of California, Los Angeles

UNIVERSITY OF MICHIGAN · UCLA

https://structures.computer/

1 · Motivation & problem

Implicit equilibrium is multi-stable: under the same boundary conditions z, the realized shape can depend on actuation history and solver warm start.

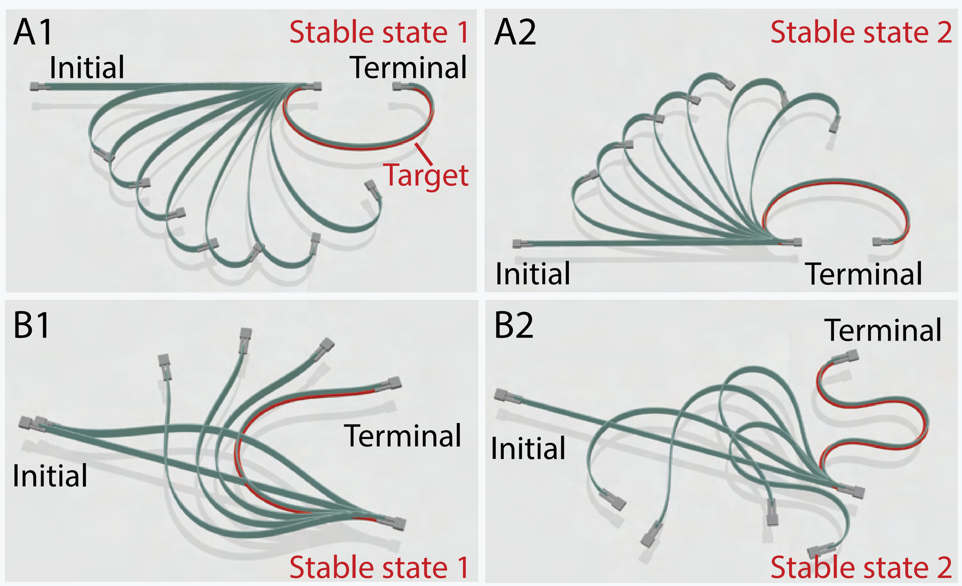


Fig. 1 · Multiple stable equilibria under identical boundary conditions.

States defined by equilibrium constraints

Equilibrium constraint
$G(x(\lambda), z(\lambda)) = 0$
Typical mechanics form
$G(x,z) = \nabla_x E(x,z) - F^{ext}$
Continuation control
$u(\lambda)=u\Theta(\lambda), \quad dz/d\lambda=f(z,u)$

Key challenge

$x(\lambda)$ is obtained only implicitly by an iterative solver and may be multi-stable, so direct backpropagation through the solver is expensive and brittle.

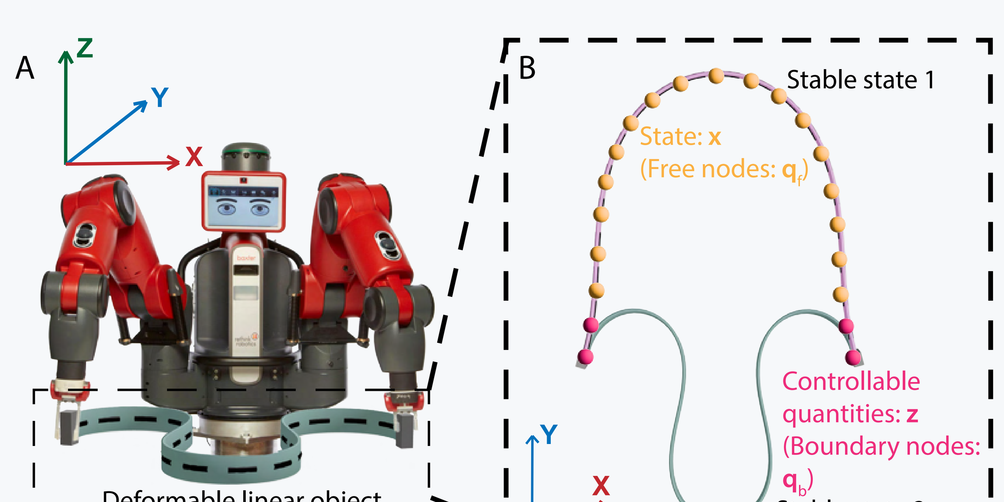


Fig. 2 · Boundary nodes z are controlled; free nodes x are solved implicitly from equilibrium.

This poster focuses on learning continuation controls without unrolling Newton/CG iterations.

Matrix-free adjoint product for scalable gradients.
Receding-horizon continuation to stay on realized branches.

2 · Adjoint learning through equilibrium

Proxy adjoint gradients via implicit differentiation

Local equilibrium sensitivity
$S(\lambda)=\partial x/\partial z, \quad G_x S = -G_z$
Matrix-free adjoint product
$G_x^T p = v \;\Rightarrow\; S^T v = -G_z^T p$
Proxy sensitivity dynamics
$dx/d\lambda = S(\lambda)\, f(z(\lambda), u(\lambda))$

Forward pass: solve $G(x,z)=0$ to convergence.
Backward pass: use frozen-tangent proxy dynamics.
No unrolling of Newton/CG iterations is required.

Receding-horizon continuation(RHC)

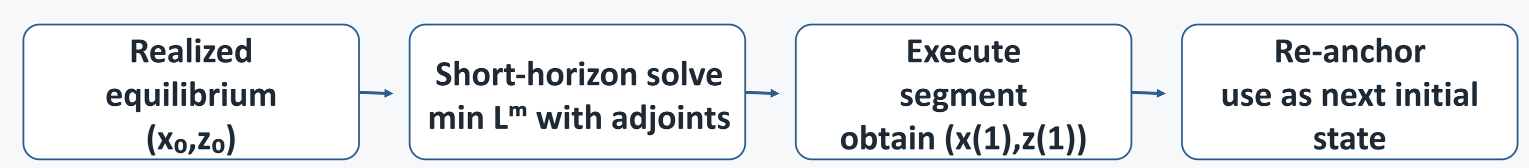


Fig. 3 · Optimize a short segment, execute it, then re-anchor from the equilibrium actually reached.

Localizes proxy-gradient error to short continuation windows.
Reduces basin drift by replanning from the reached equilibrium.
Most beneficial when multiple stable branches coexist.

Experimental protocol

Task 1 · Point-to-point targeting
Terminal loss for a selected strip node.
Task 2 · Midpoint trajectory tracking
Integral loss over realized intermediate equilibria.
Task 3 · Multistable shape formation
Curvature matching to reach a target branch.

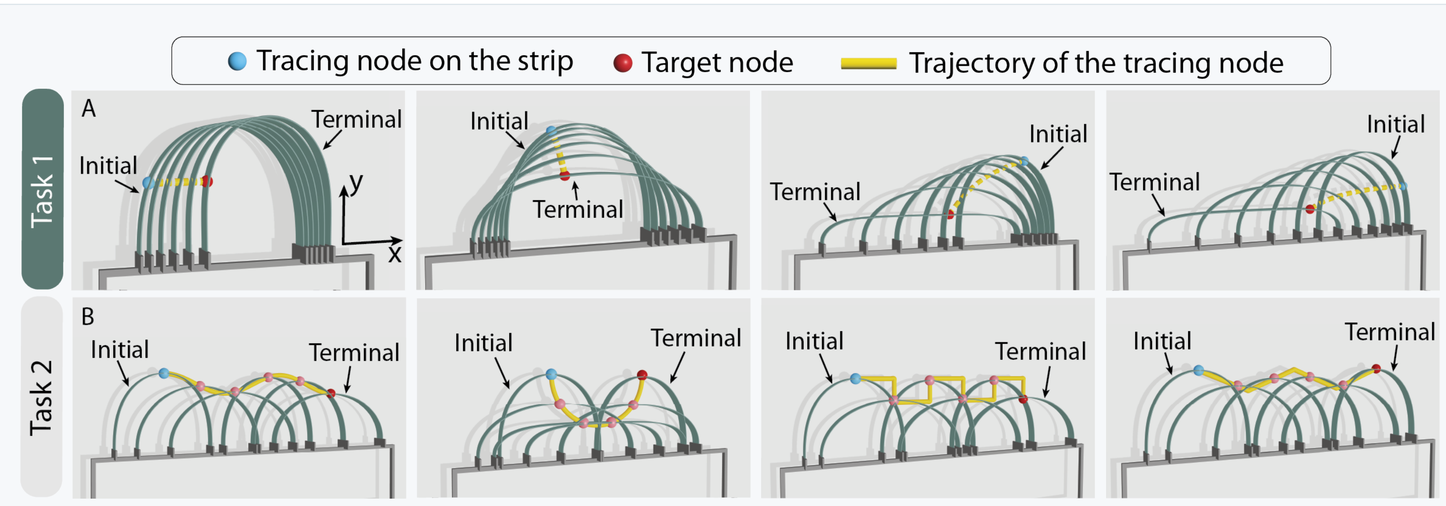


Fig. 4 · Simulated rollouts for point-to-point and trajectory tracking under two-end boundary actuation.

Baselines: SPSA and CEM · Ablation: Adjoint-only vs. Adjoint+RHC

3 · Results & validation

Convergence: gradients and RHC are both important

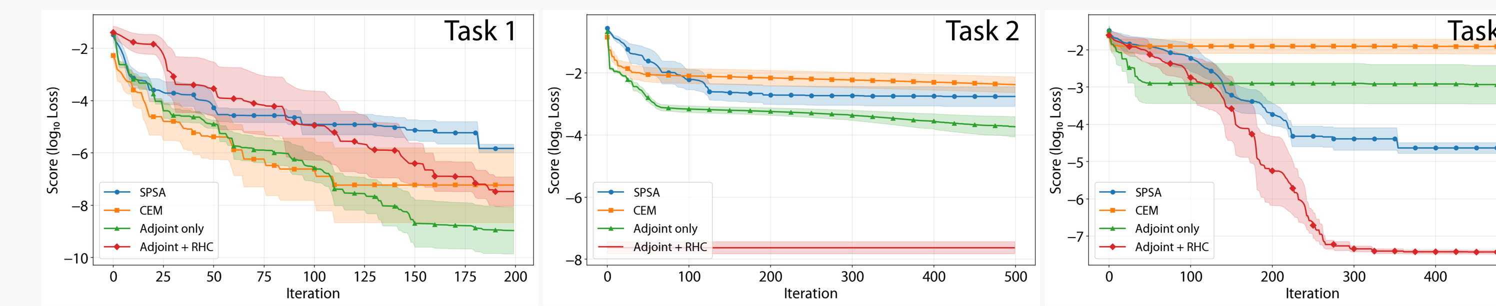


Fig. 5 · Best-so-far loss vs. optimizer iterations for Tasks 1–3 (mean ± std. err.).

Task 1: adjoint gradients converge rapidly in a short-horizon regime.
Task 2: RHC improves robustness for trajectory objectives.
Task 3: RHC avoids wrong-branch traps and reaches the lowest curvature loss.

Efficiency comparison

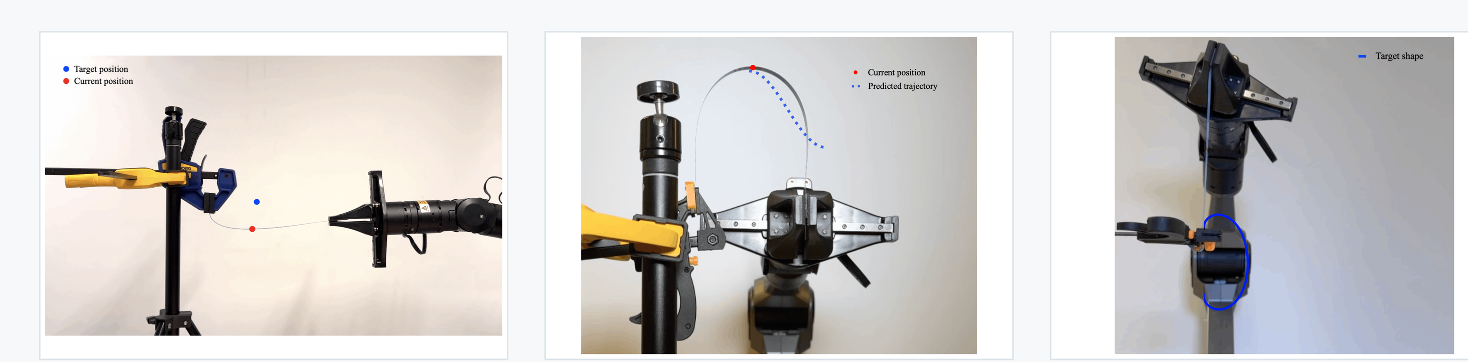
19–127× faster

Adjoint+RHC is faster than SPSA/CEM across tasks while achieving lower best loss.

Method	Theory (per optimizer update) Time/update	Theory (per optimizer update) Memory/update	Task 1 (200 updates) Time (s)↓	Task 1 (200 updates) Best loss↓
SPSA	$\mathcal{O}(2K\,C_{eq})$	$\mathcal{O}(n_\Theta)$	727.3±108.8s	2.0e-6±1.8e-6
CEM	$\mathcal{O}(PK\,C_{eq})$	$\mathcal{O}(P\,n_\Theta)$	2043.2±73.0s	6.7e-4±1.2e-3
Adjoint-only	$\mathcal{O}(K\,C_{eq} + K\,C_{lin}) \approx \mathcal{O}(K\,C_{eq})$	$\mathcal{O}(K\,(n_x+n_z)+n_\Theta)$	176.0±53.1s	3.2e-7±5.6e-7
Adjoint+RHC	$\mathcal{O}(H\,C_{eq} + H\,C_{lin}) \approx \mathcal{O}(H\,C_{eq})$	$\mathcal{O}(H\,(n_x+n_z)+n_\Theta)$	**16.1±2.8s**	**2.3e-7±3.0e-7**

Task 2 (500 updates) Time (s)↓	Task 2 (500 updates) Best loss↓	Task 3 (500 updates) Time (s)↓	Task 3 (500 updates) Best loss↓
3612.9±1231.9s	5.4e-3±7.7e-3	3967.3±417.9s	2.8e-5±1.5e-5
5358.9±208.7s	6.4e-3±3.9e-3	3176.4±232.8s	1.6e-2±8.5e-3
1111.2±83.1s	4.3e-4±5.4e-4	935.2±164.7s	8.4e-3±1.2e-2
186.9±24.8s	**3.6e-8±3.6e-8**	**50.9±6.1s**	**3.8e-8±7.4e-9**

Real-world validation



Task 1 · Point-to-point · Task 2 · Trajectory tracking · Task 3 · Shape control

Fig. 6 · Hardware rollouts on a 6-DoF arm showing sim-to-real transfer under gravity.

Takeaways

Implicit-differentiation gradients make equilibrium-constrained control practical.
RHC turns brittle long-horizon optimization into short, re-anchored solves.
The method transfers from simulation to hardware on three DLO manipulation tasks.