Closed-form control with spike coding networks

Filip S. Slijkhuis, Sander W. Keemink*, and Pablo Lanillos*
*These authors contributed equally

Neural optimal feedback control with local learning rules

Johannes Friedrich [1] Siavash Golkar [1] Shiva Farashahi [1]
Alexander Genkin [5] Anirvan M. Sengupta [2,3,4] Dmitri B. Chklovskii [1,5]

brain-marl

Berke Altiparmak

Agenda

Closed-form control with spike coding networks

Neural optimal feedback control with local learning rules

Closed-form control with spike coding networks

Filip S. Slijkhuis, Sander W. Keemink*, and Pablo Lanillos*
*These authors contributed equally

Introduction

Problem: Existing SNN systems often require dense, regular spiking patterns and learning-based optimization -> not biologically plausible or hardware-efficient.

Solution: A new method that extends Spike Coding Network (SCN) theory to incorporate closed-form optimal estimation and control, effectively turning SNNs into spiking versions of Linear-Quadratic-Gaussian (LQG) controllers.

1) Integrates Kalman filtering and LQR control into SCNs
2) Produces sparse, irregular, robust spiking.
3) Offers a fully analytical, interpretable framework suitable for low-power neuromorphic controllers.

Methods

- Work based on "Predictive Coding of Dynamical Variables in Balanced Spiking Networks" by Boerlin et al.
- The system's state x(t) is partially observed (y(t)), and a spiking neural network (SNN) must produce a control signal u(t). The network comprises Leaky Integrate-and-Fire (LIF) neurons with both fast and slow recurrent connections, capturing instantaneous and long-term effects.

Tracking a Fully Observable System

$$\dot{v} = -\lambda v + D^\top(\dot{x} + \lambda x) + \Omega_f s$$

Implementing a Dynamical System

$$\dot{v} = -\lambda v + D^\top(Ax + \lambda x) - D^\top D s$$

Replacing x with $\hat{x} = Dr$ gives a closed recurrent system with:

- Slow recurrent weights $\Omega_s = D^\top(A + \lambda I)D$
- Fast weights $\Omega_f = -D^\top D$

Variable	Meaning
$x(t)$	True system state (K-dimensional)
$\hat{x}(t) = Dr(t)$	Estimate of the state decoded from spikes
$D \in \mathbb{R}^{N \times K}$	Decoder: maps filtered spikes to state estimate
$s(t) \in \mathbb{R}^N$	Raw spike train (0 or 1 for each neuron)
$r(t) \in \mathbb{R}^N$	Filtered spike train (low-pass filtered version of $s(t)$)
$v(t)$	Voltage/membrane potential of each neuron
λ	Leak rate (decay constant)
A	Dynamics matrix for the system (how the state evolves)
$\Omega_f = -D^\top D$	Fast recurrent (inhibitory) weights
$\Omega_s = D^\top(A + \lambda I)D$	Slow recurrent weights (to encode system dynamics)

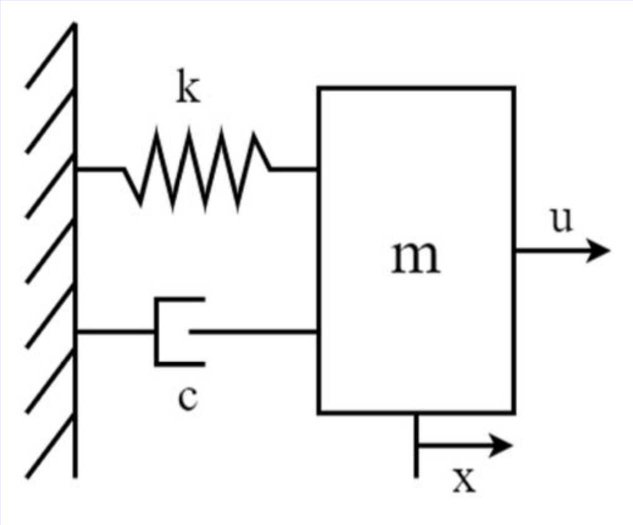
Closed-form Opt. Estim. and Control with SNNs

- Classical Control Problem: $\dot{x}(t) = Ax(t) + Bu(t) + \eta_d, \quad y(t) = Cx(t) + \eta_n$
- Goal: Estimate x from y (Kalman filter), then compute u u (LQR control).
- A continuous Kalman Filter: $\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - C\hat{x})$
- With LQR, optimal control law: $u = -K_c(x - z)$

Assuming that K_f is known, we can directly implement the optimal filtering under the SCN framework resulting in the following voltage update rule:

$$\dot{\mathbf{v}} = -\lambda\mathbf{v} + \underbrace{\mathbf{\Omega}_s\mathbf{r} + \mathbf{\Omega}_f\mathbf{s} + \mathbf{F}_i\mathbf{u}}_{\text{System estimate}} + \underbrace{\mathbf{\Omega}_k\mathbf{r} + \mathbf{F}_k\mathbf{y}}_{\text{Kalman update}} + \eta_V$$

Results



A. Spring-Mass-Damper (SMD) System

- 2D system: Position x_1 and velocity x_2
- Observed: Only position with noise

1. Estimation: SCN Kalman filter accurately reconstructs both states from noisy observations.
2. Control: Target signal is stair-wise. SCN tracks it well. Performance matches ideal Kalman+LQR controller.
3. Robustness:
   - Progressive neuron silencing (50 -> 5 neurons).
   - Network maintains control accuracy until final few neurons.
   - Other neurons compensate -> showing coordinated spiking.
4. Noise and Perturbations
   - Experiments with increasing sensor noise and external force pulses.
   - SCN shows performance on par with ideal controller.

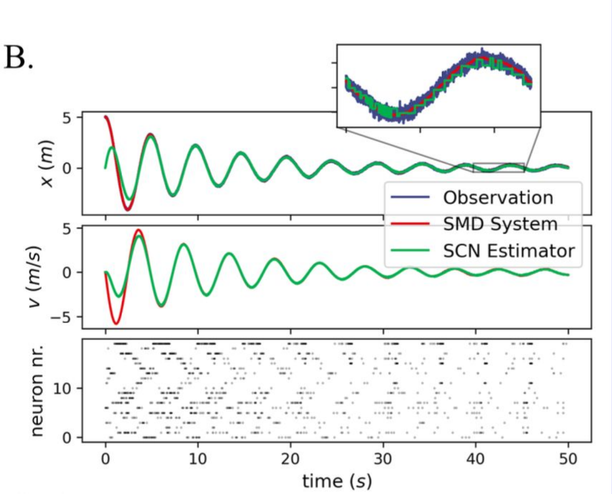
Results

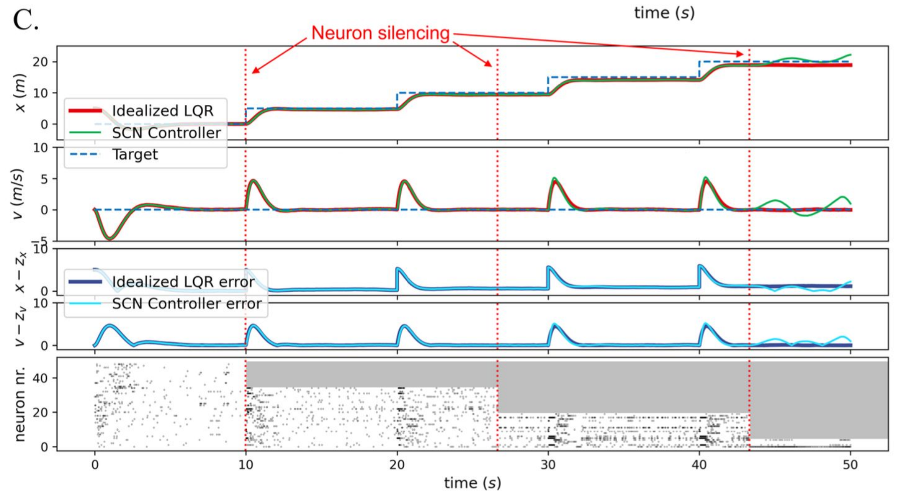
B. Cartpole System

- 4D nonlinear system: cart position/velocity, pole angle/angular velocity.
- Only cart position observed (with noise).
- SCN maintains control and keeps pole upright.
- Spiking increases during control-demanding periods, remains sparse&irregular.

C. Parameter Settings

Detailed hyperparameters for each experiment are given. Neuron counts: 20–100, λ (leak constant), decoding norm, noise levels, Q/R matrices, etc.





Advantages and Limitations of SCN Control

1. Hardware Implementation
   - Uses standard LIF neurons (compatible with Loihi, SpiNNaker) – Assumes zero synaptic delay – Requires fast + slow synapses
2. Explainability
   - Fully interpretable through control theory and SCN equations.
   - Model parameters easily tunable for sparsity, coding, etc.

3. Biological Fidelity
   - Sparse, irregular spiking
   - Matches biological patterns like excitation-inhibition balance

4. Robustness
   - Distributed computation -> tolerant to neuron loss
   - Ideal for harsh environments (e.g., radiation-exposed neuromorphic systems)

5. Energy Efficiency
   - Temporal sparsity = lower energy

Neural optimal feedback control with local learning rules

Johannes Friedrich [1] **Siavash Golkar** [1] **Shiva Farashahi** [1]

Alexander Genkin [5] **Anirvan M. Sengupta** [2,3,4] **Dmitri B. Chklovskii** [1,5]

[1] Center for Computational Neuroscience, Flatiron Institute
[2] Center for Computational Mathematics, Flatiron Institute
[3] Center for Computational Quantum Physics, Flatiron Institute
[4] Department of Physics and Astronomy, Rutgers University
[5] Neuroscience Institute, NYU Medical Center

Introduction

Problem: 1) Sensory input is noisy. 2) Sensory feedback is delayed.

Optimal Feedback Control (OFC) integrates internal model predictions and noisy observations using Kalman filtering, then computes optimal control. Yet, prior models of OFC are limited: 1) Ignore delays 2) Require noise covariances 3)Often not biologically plausible

Solution: Bio-OFC, a circuit that:

- handles delayed sensory feedback.
- performs Kalman filtering and control jointly, not in isolated phases.
- requires no prior knowledge of noise covariances or system dynamics.
- uses local synaptic plasticity rules, aligning with biological plausibility.

Achieves this by combining adaptive Kalman filtering with model-free control via policy gradient, implemented as an online algorithm in a neural network.

	[4]	[6]	[5]	[7]	[8]	Bio-OFC
delayed sensory feedback	✗	✗	✗	✗	✗	✓
control included	✗	✓	✗	✗	✗	✓
noise covariance agnostic	✗	✓	✗	✗	✗	✓
online system identification	✗	✓	✗	✓	✓✗	✓
local learning rules	N/A	✓	N/A	✗	✓	✓
tractable latent size	✗	✓	✗	✓	✓	✓
absence of inner loop	✓	✗	✓	✓	✗	✓
single phase learning/execution	N/A	✗	N/A	✓	✓	✓

Background

1. Problem Formulation

dynamics: $\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{v}_t$

observation: $\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{w}_t$

$$\boldsymbol{v}_t \sim \mathcal{N}(0; \boldsymbol{V})$$
$$\boldsymbol{w}_t \sim \mathcal{N}(0, \boldsymbol{W})$$
$$\boldsymbol{x}_0 \sim \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_0)$$

Goal: Estimate the latent state x_hat to design a control u that minimizes:

expected cost: $J = \mathbb{E}\left[\sum_{t=0}^{T} c(\boldsymbol{x}_t, \boldsymbol{u}_t)\right]$

control: $\boldsymbol{u}_t = k(\hat{\boldsymbol{x}}_t) = \arg\min J$

As the environment dynamics is not known to the animal a priori, the parameters A, B, C must be learned online.

2. Kalman Estimation and Control

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + L_t(y_t - C\hat{x}_t)$$
$$L_t = A\Sigma_t C^\top (C\Sigma_t C^\top + W)^{-1}$$

, where for quadratic costs, the optimal control is: $\boldsymbol{u}_t = -\boldsymbol{K}\hat{\boldsymbol{x}}_t$

(K is the LQR gain found via the Riccati eq.)

Neural Network Representation for OFC

Inference

The neural circuit has populations representing $\hat{x}_t$, prediction error $e_t = y_t - C\hat{x}_t$, and u_t.

Connections implement matrices A, B, C, L, K.

To account for delay:

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + L(y_{t+1-\tau} - C\hat{x}_{t+1-\tau})$$

This avoids storing full history or forward-simulating states — **biologically plausible**.

Kalman Gain Learning

Use online least squares to minimize prediction error and update:

$$\Delta\hat{A}_t \propto Le_t\hat{x}_{t-\tau}^\top$$
$$\Delta\hat{B}_t \propto Le_t u_{t-\tau}^\top$$
$$\Delta\hat{C}_t \propto e_t\hat{x}_{t+1-\tau}^\top$$
$$\Delta L_t \propto Le_t e_{t-\tau}^\top$$

These updates are local in the circuit. Also, no need to know V or W (covariances)

Control Learning

Use policy gradient to learn control weights K:

$$u_t = -K\hat{x}_t - \xi_t \quad (\xi_t \sim \mathcal{N}(0, \sigma^2 I))$$

This avoids solving the Riccati equation and supports local, online learning.

Experiments

1. Double Integrator

Goal: Drive a mass from an initial position to the target position under a force while minimizing the cost:

$$J = \sum_{t=0}^{T} x_t^\top Q x_t + \sum_{t=0}^{T-1} u_t^\top R u_t$$

, where

$x_t = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$

u_t: control force

$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$: penalizes position error

$R = 1$: penalizes control effort

$T = 10$: finite horizon

1. Changing observation noise (adaptive filtering)
   - Observation noise matrix W is altered mid-training.
   - Bio-OFC adapts Kalman gain online to compensate.
2. Delayed observations
   - Feedback delay τ (e.g., 1, 2, 3 timesteps).
   - Bio-OFC uses delayed prediction error to update state.
3. Partial observations
   - Not all state variables are directly observed.
   - Bio-OFC must estimate full state x using noisy, partial y.
4. Closed-loop learning
   - Learns estimation and control simultaneously while interacting with the environment. Performs nearly as well as optimal LQG controllers

Experiments

2. Reaching Movements (Biological Relevance)

Goal: Reproduce results from a human motor control experiment (Shadmehr & Mussa-Ivaldi, 1994)

- Simulate reaching under a force field
- Use 6D state (position, velocity, acceleration in x & y)
- Sensory delay = 50ms
- Bio-OFC learns to correct trajectories and matches human behavior
- Even with signal-dependent noise, results remain robust
- Learning mainly occurs in the estimator, not controller

3. Winged Flight (OpenAI Gym Environment)

Goal: Simulate flying toward a target against wind and gravity using 2D simplified physics

- Control = flapping frequency of left and right wings
- Observations delayed by 100ms
- Cost = L1 distance + L1 control, non-quadratic
- Bio-OFC flies directly to target; policy gradient alone overshoots
- Bio-OFC handles delays and generalizes beyond quadratic cost assumptions

Discussion

- Bio-OFC is a biologically plausible, online, and delay-resilient implementation of optimal feedback control.
- Uses adaptive Kalman filtering + model-free control with local plasticity.
- It avoids needing noise covariances, full dynamics knowledge, or multi-phase learning.
- Limitations
    - **Currently restricted to linear systems (LQR)**
    - **Suggests extension using locally linearized dynamics or neural networks for nonlinear mapping**
    - **Future exploration -> multiple sensory delays, learning the delays, combining with model-based control**

Feature	Bio-OFC
Handles delayed feedback	✅
Joint estimation and control	✅
Online learning	✅
No noise covariance needed	✅
Local synaptic updates	✅
Robust to model mismatch	✅
Matches biological behavior	✅

We have discussed:

Closed-form control with spike coding networks

01 Introduction

02 Methods

03 Closed-form Optimal Estimation and Control with SNNs

04 Results

05 Advantages and Limitations of SCN Control

Neural optimal feedback control with local learning rules

01 Introduction

02 Background

03 Neural Network Representation for OFC

04 Experiments

05 Discussion