Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Spiking Residual Control with Decoupled Event Allocation and Lyapunov-Constrained Execution

Mingyang HUANG

b2301335@ustb.edu.cn

University of Science and Technology Beijing

Research Article

Keywords: spiking neural networks, control Lyapunov functions, residual control, disturbance observers

Posted Date: September 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10746637/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Spiking Residual Control with Decoupled Event Allocation and Lyapunov-Constrained Execution

Authors: Mingyang HUANG[a, b,c] *
*Corresponding author (Mingyang HUANG, Email: b2301335@ustb.edu.cn)

Affiliations:
a Key Laboratory of Intelligent Bionic Unmanned Systems, Ministry of Education, University of Science and Technology Beijing, Beijing 100083, P. R. China;
b School of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, P. R. China;
c Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, P. R. China

†These authors contributed equally to this work

***Abstract*—Model-based controllers provide structured feedback, while learned residuals can improve tracking under unmodeled dynamics. Their combination across plants raises a practical question: how can a spiking residual remain meaningful in different command coordinates while preserving the nominal feedback action? We address this question with LyCo-SNN, which retains an Observer–LQR path and uses an action-equivalent disturbance estimate to condition a bipolar adaptive-threshold SNN. A task-specific map expresses the decoded residual as seven joint corrections for Franka or four rotor corrections for quadrotors. A closed-form Lyapunov-constrained affine directional filter then scales the residual before execution. We evaluated the controller on four continuous trajectory protocols using matched comparisons, component ablations, and ten held-out seeds per protocol. LyCo-SNN obtained 9.329 mm Cartesian RMS in generic Franka tracking, 24.603 mm under an external wrench, 0.1337 m position RMS in general flight, and 0.1093 m during 30% rotor-efficiency loss. Relative to Observer–LQR, the primary tracking error decreased in every protocol. LyCo-SNN achieved the lowest mean primary endpoint under the wrench and in both flight tests, while 99.90% of generic Franka samples remained within 25 mm. Flight event occupancy was 7.48% and 9.47%, below every neural comparator. Component ablations attributed the tracking gains to the residual, observer context, and bipolar event channel. Separating event generation, command-coordinate mapping, and directional execution therefore allows an SNN to augment structured feedback through a common residual-control rule across manipulation and flight interfaces.**

***Index Terms*—spiking neural networks, control Lyapunov functions, residual control, disturbance observers**

I. Introduction

Robotic tracking must remain accurate despite loads, parameter error, actuator degradation, and reference transients. Structured feedback gives a clear baseline, yet incomplete models and disturbance descriptions still degrade closed-loop behavior. Residual control learns a correction around that baseline [1].

Spiking neural networks (SNNs) are useful residual actors because their event activity can change with control demand [2], Event activity affects computation, so it should be measured beside tracking quality and energy use [4]. The interface matters as well. An actor designed for joint commands does not automatically transfer to rotor commands or virtual body torque.

A composite controller needs an inexpensive, interpretable nominal path and neural signals in the plant's command coordinates. Threshold adaptation decides when neurons fire. A separate runtime constraint limits the decoded correction. Predictive safety filters address this second problem [5].

LyCo-SNN combines a discrete-time LQR path, an action-equivalent disturbance observer, a bipolar adaptive-threshold SNN, and a task-coordinate residual interface. In Franka, joint and observer errors produce seven joint residuals. For general quadrotor tracking, the 12-dimensional tracking error and observer context produce four rotor residuals. Under rotor-efficiency loss, the adapter maps attitude and body-rate errors to a three-axis virtual torque for the common fault allocator. The Franka and both quadrotor controllers apply a closed-form affine directional scale to their decoded residuals.

We evaluate paired tracking performance in disturbance and fault windows. The analysis also reports realized control outputs, event activity, component ablations, and violation diagnostics. Four held-out continuous-trajectory protocols cover generic Cartesian and wrench-disturbed Franka tracking, general quadrotor tracking, and quadrotor tracking under rotor-efficiency loss.

The main contributions are:

- LyCo-SNN separates residual generation from control execution. The Observer–LQR action remains explicit, while an observer-conditioned bipolar adaptive-threshold SNN produces only a signed correction proposal rather than replacing the nominal feedback path.
- To turn that proposal into an admissible action, a closed-form directional filter selects the largest scale satisfying the affine Lyapunov decrement, residual-amplitude, and actuator-box conditions. Four theorems establish point-wise decrement, nominal affine convergence, relative-context composite decay, and practical boundedness under execution error.
- Task-coordinate interfaces carry the same proposal-to-execution rule into seven Franka joint corrections and four quadrotor rotor corrections. Candidate and executed residuals, the applied scale, event activity, input limits, and

Mingyang Huang is with the School of Artificial Intelligence, Key Laboratory of Intelligent Bionic Unmanned Systems (Ministry of Education), and Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China (email: b2301335@ustb.edu.cn).

[3]. Existing SNN control studies cover state encoding, spike-to-action decoding, adaptation, and closed-loop performance.

affine violations remain observable at every control step.

- Paired ten-seed evaluation across four continuous-trajectory protocols shows lower primary tracking error than Observer–LQR in every protocol and the lowest mean primary endpoint under the external wrench and in both flight tests. Component ablations isolate the roles of the residual, observer context, and bipolar event channel, while directional diagnostics connect the executed scale to the closed-loop response.

The remainder of this paper is organized as follows. Section II reviews structured feedback, SNN control, residual learning, and runtime filtering. Section III describes the LyCo-SNN architecture, task-coordinate interfaces, directional execution mechanism, and training procedure. Section IV specifies the four tasks, comparison controllers, evaluation metrics, and paired held-out protocol. Section V presents the tracking results, directional diagnostics, closed-loop trajectories, and component ablations. Section VI discusses task-dependent behavior, computational characteristics, and the scope of the evaluation. Section VII concludes the paper and outlines future experimental work.

II. Related Work

A. Structured Feedback and Constrained Robotic Control

PID and LQR remain common nominal choices because they are deterministic, computationally light, and easy to audit at the control interface. PID exposes proportional, integral, and derivative correction directly, including in spike-coded implementations [6]. LQR instead provides a state-feedback gain for a declared local model and a quadratic value matrix that can be reused in residual-action analysis. Neither choice by itself removes nonlinear mismatch, actuator faults, or persistent external loads.

More specialized structured controllers target these effects in different ways. Geometric control works directly with quadrotor configuration geometry [7]. Sliding-mode and terminal sliding-mode designs target robust or finite-time tracking, including fault-tolerant composite schemes [8]. Predictive filters enforce state or input requirements through online model evaluation [5], [9]. These methods motivate the structured backbone used here, but they are not interchangeable baselines. Our Observer–LQR comparison disables only the neural residual, thereby isolating what the learned correction adds to the shared nominal path.

B. SNN Controllers for Robotic and Continuous Control

SNN-based robotic control spans evolved task-specific circuits, biologically motivated motor models, neuromorphic implementations, and learned continuous policies [10]. Early real-robot work evolved spike-response networks online for wall following and compared them with standard genetic-algorithm and fuzzy-control alternatives [11]. Subsequent studies used compact SNNs to control nonlinear simulated plants and musculoskeletal arm models [12], [13], while end-to-end lane keeping combined event-camera input with transferred deep Q-learning or reward-modulated STDP in simulation [14]. These studies established several routes from sensed variables to spiking motor commands, but their evaluation remained task- and platform-specific.

Neuromorphic demonstrations expose deployment effects that software spike counts alone cannot capture. Spike-coded PID arithmetic has been implemented on Loihi for a wheeled robot and a one-degree-of-freedom UAV [6], and an FPGA-based spiking PID controlled four joints of a physical BioRob arm [15]. Nengo-based systems mapped a simulated rover and an adaptive arm component to Loihi and directly measured power and control-loop latency [16]. A later Loihi implementation reproduced an operational-space controller for a simulated seven-degree-of-freedom arm, reporting measured hardware energy and throughput while retaining a simulated plant [17]. At the system level, these examples support neuromorphic co-design across sensing, computation, communication, and actuation rather than inferring end-to-end efficiency from neural activity alone [18]. An imitation-trained SNN has also been deployed on a physical mobile robot, although its reported energy advantage was an operation-count estimate rather than an on-chip measurement [19].

For high-dimensional continuous control, PopSAN combines population encoding, LIF hidden layers, and population decoding [2], whereas ILC-SAN adds intralayer connections and membrane-based decoding [3]. A recent preprint further evaluates a population-coded actor on simulated Franka manipulation tasks in Isaac Gym [20]. These actor architectures provide practical routes from spikes to continuous commands, but they generate the complete policy action rather than a correction constrained relative to an existing nominal controller.

C. SNNs in Adaptive and Composite Control

Another line of work places an SNN inside a structured controller or adaptive motor model. The REACH model implemented task-space arm control with separate online adaptation to kinematic and dynamic changes in a large spiking network but evaluated the closed loop in simulation [21]. Pérez Fernández et al. reproduced an LQR path-following law with SNN associative maps and used error-modulated spike-timing-dependent plasticity to adapt to changes in vehicle adhesion [22]. Their stability study was empirical, using cell mapping over a discretized state region, followed by real-time high-fidelity vehicle simulation. Peng et al. combined a characteristic model, golden-section feedback, and an STDP-trained SNN for hex-rotor trajectory tracking, with numerical and outdoor prototype comparisons against the non-SNN compound controller [23]. Yu et al. instead used an event-driven-threshold-plasticity SNN to approximate an unknown nonlinear term within an adaptive fast-terminal sliding-mode fault-tolerant attitude controller [8]. Their finite-time practical tracking analysis concerns the complete composite controller, while the evaluation is limited to quadrotor simulations and identifies the implemented SNN as clock-driven. Collectively, these studies motivate combining spiking adaptation with structured feedback, but they do not separate internal spike allocation from runtime admissibility of a learned residual

action.

D. Residual Learning, Event Allocation, and Runtime Filtering

Residual learning retains a structured baseline and learns only the unmodeled correction [1]. Learned observers can estimate state and lumped uncertainty under explicit convergence conditions [24]; the observer used here is instead a fixed action-equivalent estimator and carries no learned-observer convergence claim. Classical event-triggered control reduces sensing, communication, or actuation updates through a plant-level trigger [25], while event-driven adaptive dynamic programming can trigger neural-weight updates [26]. These mechanisms operate at different levels from allocating spikes inside a fixed-period residual actor. The LyCo-SNN backbone treats neural event allocation and residual-action auditing as separate decisions: the spiking actor determines when events are spent, whereas the plant interface determines how the decoded proposal is bounded, filtered, or allocated. A complementary analytical line maps Kalman estimation and LQR control into spike-coding networks for fully identified linear systems [27]. That construction produces sparse, irregular activity and was tested on simulated spring–mass–damper and Cart–Pole systems, but it does not address a learned residual acting beside an existing nonlinear-task controller or report a hardware implementation.

Reviews of learned certificates and safe learning emphasize that formal claims remain tied to the declared model, domain, and verification assumptions [28], [29]. Learning-based control constraints are commonly imposed either during controller and certificate synthesis [30], [31], [32] or through a runtime correction such as a predictive safety filter or CLF–CBF quadratic program [5]. LyCo-SNN does not synthesize a nonlinear certificate. The Franka implementation applies a closed-form scalar filter to the residual direction under its declared affine prediction. The task-adapted quadrotor implementation applies the same calculation to its virtual-torque residual before fault allocation. Nonlinear execution is therefore evaluated empirically, and software spike activity is reported as a computation proxy rather than as measured hardware energy

III. LyCo-SNN Controller Architecture, Certificate Chain, and Deployment

This section develops the LyCo-SNN residual-control architecture and its certificate chain, then instantiates the shared formulation for the Franka and quadrotor control interfaces.

A. Nominal Affine Model and Execution Discrepancy

Let $T_s > 0$ denote the fixed controller period and $k \in \mathrm{Z}_{\geq 0}$ the control-sample index. Throughout, $\|\cdot\|_2$ denotes the Euclidean vector norm or induced matrix 2-norm, I_r is the $r \times r$ identity matrix, and $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ denote the extreme eigenvalues of a symmetric matrix M. The matrices and dimensions below are plant specific. We suppress the plant index unless a statement compares the three controlled objects. Consider the disturbed discrete-time plant

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + d_k, \\ x_k \in \mathrm{R}^n, u_k \in U \subset \mathrm{R}^m, d_k \in \mathrm{R}^n. \end{cases} \tag{1}$$

Here $A \in \mathrm{R}^{n\times n}$ and $B \in \mathrm{R}^{n\times m}$ are the declared nominal matrices, $\mathrm{U} = [\underline{u}, \bar{u}]$ is a component wise input box, and d_k collects the one-step effect of loads, model mismatch, reference motion, and unmatched disturbance. For tracking tasks, x_k denotes a local tracking-error state. Stabilizability of (A, B) is sufficient: choose $K \in \mathrm{R}^{m\times n}$ such that $F = A - BK$ is Schur, so every eigenvalue of F lies strictly inside the unit disk. The gain K may be obtained from a linear-quadratic regulator (LQR) design. Given any symmetric $Q \succ 0$, the discrete Lyapunov equation

$$F^{\top} PF - P = -Q. \tag{2}$$

has a unique solution $P \succ 0$. We define the nominal state energy as $V_x(x) = x^{\top} Px$.

With the observer compensation and residual correction introduced below, the declared recentered prediction is

$$\hat{x}_{k+1} = Fx_k + B\delta_k. \tag{3}$$

The known nominal offset and assumed matched compensation are removed in this recentered coordinate model; any imperfect cancellation enters $w_{i,k}$ below. For plant $i \in \mathrm{I} := \{\mathrm{cp}, \mathrm{F}, \mathrm{q}\}$, the measured nonlinear transition is represented as

$$x_{k+1} = Fx_k + B\delta_k + w_{i,k}, \tag{4}$$

where $w_{i,k} \in \mathrm{R}^n$ aggregates observer error, nonlinear model mismatch, actuator realization, reference-coordinate motion, and external disturbance. The nominal result sets $w_{i,k} = 0$. The practical stability result retains $w_{i,k}$ as the additive discrepancy between affine prediction and nonlinear execution.

B. Matched-disturbance Observer and Neural Context

For $k \geq 1$, the measured one-step innovation is

$$\iota_k = x_k - Ax_{k-1} - Bu_{k-1}. \tag{5}$$

Shifting the plant equation by one sample and subtracting the known terms yields the measurable residual in (5). The vector $L_B \iota_k$ is also the unique minimizer of $\min_g \left\{ \|Bg - \iota_k\|_2^2 + \varepsilon_B \|g\|_2^2 \right\}$. Let $\varepsilon_B > 0$ be a regularization constant and define $L_B = (B^{\top} B + \varepsilon_B I_m)^{-1} B^{\top}$. Component-wise clipping of $L_B \iota_k$ to $[-g_c, g_c]$ gives the instantaneous action-equivalent estimate g_k^{inst}, where $g_c > 0$ is the innovation-clipping bound. Let $\Pi_{\hat{g}_c}$ denote component-wise projection onto $[-\hat{g}_c, \hat{g}_c]^m$, where $\hat{g}_c > 0$ is the stored-estimate limit. With observer gain $\ell_o \in (0,1)$, initialize $\hat{g}_0 \in \mathrm{R}^m$ and define the slow estimate and fast innovation context by

$$\begin{cases} \hat{g}_k = \Pi_{\hat{g}_c}\left((1-\ell_o)\hat{g}_{k-1} + \ell_o g_k^{\mathrm{inst}} \right), \\ \xi_k = g_k^{\mathrm{inst}} - \hat{g}_{k-1}. \end{cases} \tag{6}$$

The slow observer estimate supplies the action offset $o_k = -\hat{g}_k$. The certificate-space nominal action is $u_k^{\mathrm{nom}} = o_k - Kx_k$. Task references and actuator mappings are composed outside this shared certificate interface and are specified in Subsection III-F. Imperfect compensation and execution mismatch remain in $w_{i,k}$. For a relative-context deployment, choose $L_c > 0$ and define the radial projection

$$\tilde{\xi}_k = \Pi_{L_c}(x_k, \xi_k) := \begin{cases} 0, & \|\xi_k\|_2 = 0, \\ \min\left\{1, \dfrac{L_c \|x_k\|_2}{\|\xi_k\|_2}\right\} \xi_k, & \|\xi_k\|_2 > 0, \end{cases} \tag{7}$$

$$\|\tilde{\xi}_k\|_2 \le L_c \|x_k\|_2 .$$

The stacked relative-context input has norm at most $\sqrt{1+L_c^2}\,\|x_k\|_2$, which closes the corresponding energy estimate. Without projection, ξ_k remains an explicit forcing term. The implementation uses a numerical zero guard only to avoid division by zero. Let $\xi_k^{\mathrm{net}} = \tilde{\xi}_k$ for the relative-context branch and $\xi_k^{\mathrm{net}} = \xi_k$ for the clipped raw-context branch. The selected branch is part of the deployment map in Subsection III-F. These branches lead to the distinct composite-energy bounds in Subsection III-E.

C. Lyapunov-coupled Bipolar Spiking Residual

The neural input is $y_k = [x_k^{\top}, (\xi_k^{\mathrm{net}})^{\top}]^{\top} \in \mathrm{R}^{d_y}$. Let $v_k \in \mathrm{R}^N$ be the membrane state, $\Lambda \in \mathrm{R}^{N\times N}$ the leak matrix, $W_{\mathrm{in}} \in \mathrm{R}^{N\times d_y}$ the encoder, and $W_{\mathrm{dec}} \in \mathrm{R}^{m\times N}$ the decoder. The bipolar leaky integrate-and-fire update, for neuron $j \in \{1,\dots,N\}$, is

$$\begin{cases} \bar{v}_k = \Lambda v_k + W_{\mathrm{in}} y_k, \\ s_{j,k} = \mathrm{sign}(\bar{v}_{j,k})\,\mathbf{1}\{|\bar{v}_{j,k}| \ge \vartheta_{j,k}\}, \\ v_{k+1} = \bar{v}_k \odot (\mathbf{1} - |s_k|), \\ z_k = W_{\mathrm{dec}} s_k . \end{cases} \tag{8}$$

Here v_k is the pre-reset membrane state, $s_k \in \{-1,0,1\}^N$, $\mathbf{1}\{\cdot\}$ is the indicator function, $\odot$ denotes elementwise multiplication, and $z_k \in \mathrm{R}^m$ is the decoded residual proposal. The explicitly evaluated composite scheduling energy is

$$\mathrm{V}_k = x_k^{\top} P x_k + \beta \|v_k\|_2^2, \beta > 0. \tag{9}$$

The plant-side Lyapunov quantity remains V_x. The composite quantity V_k only schedules neural events and supports membrane-boundedness analysis. In particular, the reset produces the exact jump

$$\beta\left(\|v_{k+1}\|_2^2 - \|\bar{v}_k\|_2^2\right) = -\beta \sum_{j:|s_{j,k}|=1} \bar{v}_{j,k}^2 \le 0. \tag{10}$$

Because (8) is evaluated once per fixed period T_s and each neuron emits at most one bipolar event per sample, the network produces at most N events per period. This sampling rule excludes finite-time event accumulation without requiring a continuous-time Zeno premise. For $0 < \rho_{\mathrm{safe}} < \rho_{\mathrm{warn}}$, $0 < \vartheta_{j,\min} \le \vartheta_{j,\max}$, and $p_\vartheta > 0$, define

$$\begin{cases} \varpi(V) = clip\left(\dfrac{\rho_{\mathrm{warn}} - \mathrm{V}}{\rho_{\mathrm{warn}} - \rho_{\mathrm{safe}}}, 0, 1\right), \\ \vartheta_{j,k} = \vartheta_{j,\min} + (\vartheta_{j,\max} - \vartheta_{j,\min})\,\varpi(\mathrm{V}_k)^{p_\vartheta}. \end{cases} \tag{11}$$

The scalar clipping function is $\mathrm{clip}(a,0,1) = \min\{1, \max\{0,a\}\}$. Low scheduling energy raises the threshold toward $\vartheta_{j,\max}$, whereas high energy lowers it toward $\vartheta_{j,\min}$. Threshold scheduling allocates events. The filter below enforces action admissibility by preserving the decoded direction and selecting its largest feasible scale.

D. Affine directional filter

The controller operates in certificate-space actions:

$$\begin{cases} u_k^{\mathrm{nom}} = o_k - Kx_k, u_k = u_k^{\mathrm{nom}} + \delta_k, \\ \delta_k = \eta_k z_k, \quad \eta_k \in [0,1]. \end{cases} \tag{12}$$

After recentering by the observer estimate, the declared affine prediction is $Fx_k + B\eta z_k$. For the current pair (x_k, z_k), define

$$\begin{cases} \alpha(z_k) = z_k^{\top} B^{\top} P B z_k, \\ \chi(x_k, z_k) = 2x_k^{\top} F^{\top} P B z_k, \\ Q(x_k) = x_k^{\top} Q x_k, \\ c(x_k) = (1-\sigma) Q(x_k), \quad \sigma \in (0,1), \end{cases} \tag{13}$$

$$\Delta_\eta(x_k, z_k) = \chi(x_k, z_k)^2 + 4\alpha(z_k) c(x_k). \tag{14}$$

The desired state decrement is equivalent to

$$\alpha(z_k)\eta^2 + \chi(x_k, z_k)\eta \le c(x_k). \tag{15}$$

Because $\eta = 0$ is feasible and the left-hand side is convex, the positive root terminates the nonnegative feasible interval when $\alpha(z_k) > 0$. The affine case is handled separately. The nonnegative boundary of the directional CLF-feasible interval is

$$\eta_+(x_k, z_k) = \begin{cases} 1, & \alpha(z_k) = 0, \\ \dfrac{-\chi(x_k, z_k) + \sqrt{\Delta_\eta(x_k, z_k)}}{2\alpha(z_k)}, & \alpha(z_k) > 0 \end{cases}. \tag{16}$$

Let $\eta_{\mathrm{clf},k} = \min\{1, \eta_+(x_k, z_k)\}$ and choose a residual gain limit $c_\delta > 0$. The amplitude scale is

$$\eta_{\mathrm{amp},k} = \begin{cases} 1, & \|z_k\|_2 = 0, \\ \min\left\{1, \dfrac{c_\delta \|x_k\|_2}{\|z_k\|_2}\right\}, & \|z_k\|_2 > 0 \end{cases}. \tag{17}$$

The amplitude constraint enforces $\delta_{k2} \le c_\delta x_{k2}$. If $u_k^{\mathrm{nom}} \in \mathrm{U}$, the box-feasibility scale is

$$\eta_{\mathrm{U},k} = \min\left\{1, \min_{j: z_{j,k} > 0} \frac{\bar{u}_j - u_{j,k}^{\mathrm{nom}}}{z_{j,k}}, \min_{j: z_{j,k} < 0} \frac{\underline{u}_j - u_{j,k}^{\mathrm{nom}}}{z_{j,k}}\right\}. \tag{18}$$

Here an empty inner minimum is $+\infty$. The applied scale and correction are

$$\eta_k = \min\left\{\eta_{\mathrm{clf},k}, \eta_{\mathrm{amp},k}, \eta_{\mathrm{U},k}\right\}, \delta_k = \eta_k z_k. \tag{19}$$

A decoded direction orthogonal to the affine Lyapunov descent direction has $\chi(x_k, z_k) = 0$, whereas a locally adverse direction has $\chi(x_k, z_k) > 0$. The CLF root still gives a positive scale for $x_k \neq 0$ unless the amplitude or input margin is exhausted. When the decoded direction is strongly adverse, η_k can approach zero and the controller reverts to the nominal command. This behavior preserves the declared affine-model inequality but can reduce the benefit of the learned residual. Floating-point execution uses a cancellation-safe evaluation of (16) and a conservative small-α branch.

Figure 1 summarizes this division of responsibilities. The upper path converts the measured state and observer innovation into a sparse proposal. The lower path combines the nominal command with the directionally filtered correction before actuation. The state schedules activity through V_k and independently enters the affine-model condition through V_x.

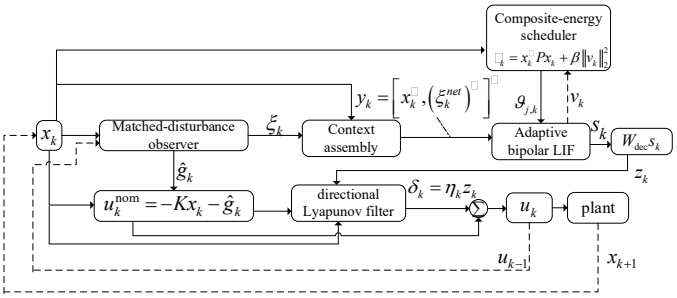


Figure 1. Observer-coupled LyCo-SNN forward pass. Composite scheduling energy sets the firing threshold. The affine directional filter scales the decoded residual before it is combined with the nominal command.

E. Certificate Chain for Convergence and Practical Boundedness

The first theorem establishes the one-step state decrement enforced by the directional scale. The second converts that decrement into nominal affine-model convergence. The third includes the membrane recursion under relative context. The fourth retains raw context and the execution discrepancy $w_{i,k}$ to give an input-to-state practical bound.

Theorem 1 (Pointwise state decrement). *For the recentered undisturbed nominal prediction, if $u_k^{\mathrm{nom}} \in \mathrm{U}$, then the scale in (19) satisfies*

$$\begin{cases} V_x\left(Fx_k + B\eta_k z_k\right) - V_x\left(x_k\right) \le -\sigma x_k^{\cdot} Q x_k, \\ u_k^{\mathrm{nom}} + \eta_k z_k \in \mathrm{U}. \end{cases} \tag{20}$$

Proof. Substituting $x_{k+1}^{nom} = Fx_k + B\eta_k z_k$ into V_x and using (2) gives

$$V_x\left(x_{k+1}^{nom}\right) - V_x\left(x_k\right) = -\mathrm{Q}\left(x_k\right) + \eta_k \chi\left(x_k, z_k\right) + \eta_k^2 \alpha\left(z_k\right). \tag{21}$$

Inequality (20) is therefore equivalent to $\alpha(z_k)\eta_k^2 + \chi(x_k, z_k)\eta_k \le c(x_k)$. Zero is feasible. If $\alpha(z_k) > 0$, the connected nonnegative feasible interval ends at (16). If $\alpha(z_k) = 0$, positive definiteness of P implies $Bz_k = 0$, so the candidate does not alter the predicted state. The additional amplitude and input minima preserve CLF feasibility and enforce the stated constraints.

Let $p_{\min} = \lambda_{\min}(P)$, $p_{\max} = \lambda_{\max}(P)$, $q_{\min} = \lambda_{\min}(Q)$, and

$$a = \frac{\sigma q_{\min}}{p_{\max}} \in (0,1) \tag{22}$$

Theorem 2 (Nominal affine-model convergence). Consider the declared recentered affine prediction $x_{k+1}^{\mathrm{nom}} = Fx_k^{\mathrm{nom}} + B\eta_k z_k$ with $x_0^{\mathrm{nom}} = x_0$. If $u_k^{\mathrm{nom}} \in \mathrm{U}$ for every k, then

$$\left\|x_k^{\mathrm{nom}}\right\|_2 \le \sqrt{\frac{p_{\max}}{p_{\min}}}(1-a)^{k/2}\left\|x_0\right\|_2. \tag{23}$$

Proof. Theorem 1 gives

$$V_x\left(x_{k+1}^{\mathrm{nom}}\right) \le V_x\left(x_k^{\mathrm{nom}}\right) - \sigma q_{\min}\left\|x_k^{\mathrm{nom}}\right\|_2^2 \le (1-a)V_x\left(x_k^{\mathrm{nom}}\right). \tag{24}$$

Iteration of (24), together with $p_{\min}\left\|x_k^{\mathrm{nom}}\right\|_2^2 \le V_x\left(x_k^{\mathrm{nom}}\right)$ and $V_x\left(x_0\right) \le p_{\max}\left\|x_0\right\|_2^2$, gives (23).

This theorem concerns the declared affine prediction. In the executed nonlinear transition, $w_{i,k}$ is generally nonzero and is handled by the execution-error ISpS theorem below.

Let $\lambda_\Lambda = \left\|\Lambda\right\|_2 < 1$. For relative context, $\left\|y_k\right\|_2 \le g_y\left\|x_k\right\|_2$ with $g_y = \sqrt{1+L_c^2}$. Choose $\dot{o}_v > 0$ such that $(1+\dot{o}_v)\lambda_\Lambda^2 < 1$, and define

$$a_v = 1 - (1+\dot{o}_v)\lambda_\Lambda^2, c_v = \left(1+\dot{o}_v^{-1}\right)\left\|W_{\mathrm{in}}\right\|_2^2 g_y^2. \tag{25}$$

Theorem 3 (Relative-context composite exponential decay). *If (7) holds and*

$$0 < \beta < \frac{\sigma\lambda_{\min}(Q)}{c_v}. \tag{26}$$

then every trajectory remaining in the nominal-feasible domain satisfies

$$\mathrm{V}_{k+1} - \mathrm{V}_k \le -\gamma \mathrm{V}_k,$$
$$\gamma = min\left\{\frac{\sigma\lambda_{\min}(Q) - \beta c_v}{\lambda_{\max}(P)}, a_v\right\} > 0. \tag{27}$$

Proof. Hard reset cannot increase the membrane norm. Young's inequality and (7) therefore give

$$\left\|v_{k+1}\right\|_2^2 \le (1+\dot{o}_v)\lambda_\Lambda^2\left\|v_k\right\|_2^2 + \left(1+\dot{o}_v^{-1}\right)\left\|W_{\mathrm{in}}\right\|_2^2 g_y^2\left\|x_k\right\|_2^2. \tag{28}$$

Adding β times the membrane-energy change implied by (28) to (20), and using $x^{\cdot} Qx \ge \lambda_{\min}(Q)\left\|x\right\|_2^2$, gives

$$\mathrm{V}_{k+1} - \mathrm{V}_k \le -\left[\sigma\lambda_{\min}(Q) - \beta c_v\right]\left\|x_k\right\|_2^2 - \beta a_v\left\|v_k\right\|_2^2. \tag{29}$$

Condition (26) makes both coefficients positive; comparison with $\mathrm{V}_k \le \lambda_{\max}(P)\left\|x_k\right\|_2^2 + \beta\left\|v_k\right\|_2^2$ yields γ in (27).

For raw context, partition $W_{\mathrm{in}} = \left[W_x\ W_\xi\right]$. Choose $\dot{o}_\xi > 0$ and define

$$\begin{cases} c_x^{\mathrm{raw}} = \left(1+\dot{o}_v^{-1}\right)\left(1+\dot{o}_\xi\right)\|W_x\|_2^2, \\ c_\xi^{\mathrm{raw}} = \left(1+\dot{o}_v^{-1}\right)\left(1+\dot{o}_\xi^{-1}\right)\|W\|_{\xi 2}^{\ 2}, \\ m_x^{\mathrm{raw}} = \sigma\lambda_{\min}(Q) - \beta c_x^{\mathrm{raw}}, \\ \gamma_{\mathrm{raw}} = min\left\{\dfrac{m_x^{\mathrm{raw}}}{\lambda_{\max}(P)}, a_v\right\}. \end{cases} \tag{30}$$

The same Young-inequality decomposition now gives

$$\|v_{k+1}\|_2^2 \le (1-a_v)\|v_k\|_2^2 + c_x^{\mathrm{raw}}\|x_k\|_2^2 + c_\xi^{\mathrm{raw}}\|\xi_k\|_2^2. \tag{31}$$

The last term has no state-relative bound and cannot enter the state coefficient. Let $w_{i,k} = x_{k+1} - (Fx_k + B\delta_k)$ denote the aggregate error remaining after observer recentering, and define $L_y = \|F\|_2 + \|B\|_2 c_\delta$, $c_1 = \|P\|_2 L_y / \sqrt{\lambda_{\min}(P)}$, and $C_w = 2c_1^2 / \gamma_{\mathrm{raw}} + \|P\|_2$.

Theorem 4 (Execution-error ISpS boundedness). *If* $m_x^{\mathrm{raw}} > 0$, $u_k^{\mathrm{nom}} \in \mathrm{U}$ *, and the trajectory remains in the declared local domain, then*

$$\mathrm{V}_{k+1} \le (1-\gamma_{\mathrm{raw}}/2)\mathrm{V}_k + C_w\|w_{i,k}\|_2^2 + \beta c_\xi^{\mathrm{raw}}\|\xi_k\|_2^2. \tag{32}$$

If $\|w_{i,k}\|_2 \le \bar{w}_i$ *and* $\|\xi_k\|_2 \le \bar{\xi}$ *, then*

$$\limsup_{k\to\infty}\|x_k\|_2 \le \sqrt{\frac{2(C_w\bar{w}_i^2 + \beta c_\xi^{\mathrm{raw}}\bar{\xi}^2}{\gamma_{\mathrm{raw}}\lambda_{\min}(P)}}. \tag{33}$$

Proof. Two applications of Young's inequality bound the raw-context membrane update by $(1-a_v)\|v_k\|_2^2 + c_x^{\mathrm{raw}}\|x_k\|_2^2 + c_\xi^{\mathrm{raw}}\|\xi_k\|_2^2$. Combining this inequality with (20) gives unforced decay at rate γ_{raw} plus the explicit context term. Expanding the state energy after adding $w_{i,k}$, and using Young's inequality once more, absorbs half of the decay margin and produces $C_w\|w_{i,k}\|_2^2$. Under the stated uniform bounds, define $b = C_w\bar{w}_i^2 + \beta c_\xi^{\mathrm{raw}}\bar{\xi}^2$. Iterating (32) gives

$$\mathrm{V}_k \le (1-\gamma_{\mathrm{raw}}/2)^k\mathrm{V}_0 + \frac{2b}{\gamma_{\mathrm{raw}}}. \tag{34}$$

Because $\lambda_{\min}(P)\|x_k\|_2^2 \le \mathrm{V}_k$, the limit superior gives (33). The bound is conditional on the stated operating domain, action feasibility, and discrepancy bounds.

F. Plant-specific deployment

The Franka controller takes the 14-dimensional joint position/velocity error and seven-dimensional observer context. Its 128-neuron actor returns a seven-joint residual. Both Franka protocols follow a continuous Cartesian reference generated through damped-least-squares inverse kinematics. The generic protocol has no external wrench. The second protocol adds a sustained wrench and a short wrench pulse. Separate frozen Franka checkpoints retain the same 21-dimensional input, seven-joint output, bounded action interface, and forward-kinematic Cartesian endpoint. The certificate action is a correction, $q_k^{\mathrm{cmd}} = q_k^{\mathrm{ref}} + a_k$; consequently, (18) is evaluated in correction space before the physical position command is composed.

For both quadrotor protocols, a 128-neuron actor takes the 12-dimensional tracking error and four-dimensional observer context and returns a four-rotor residual $z_k^{\mathrm{q}} \in \mathbb{R}^4$. The executed rotor command is

$$u_k^{\mathrm{q}} = u_k^{\mathrm{nom}} + \eta z_k^{\mathrm{q}}. \tag{35}$$

Observer–LQR uses the same nominal path and actuator box with $z_k^{\mathrm{q}} = 0$. The reference generator, checkpoint, observer, directional filter, and four-dimensional action coordinates are identical in the general- flight and fault-flight evaluations. This controller uses the relative-context branch with $L_c = 1$.

The fault protocol changes only the plant-side rotor effectiveness. Let $E_k = \mathrm{diag}(1,\rho_k,1,1)$, where $\rho_k = 0.70$ during the prescribed rotor-2 fault and $\rho_k = 1$ otherwise. Relative to the nominal certificate input Bu_k^{q} , this introduces

$$w_k^{\mathrm{fault}} = B(E_k - I)u_k^{\mathrm{q}} \tag{36}$$

The deployed rotor command is box bounded and $\|E_k - I\|_2 \le 0.30$, so $\|w_k^{\mathrm{fault}}\|_2 \le 0.30\|B\|_2\,\bar{u}$. The fault term therefore enters the bounded execution-error term of Theorem 4; the certificate-space controller and the conditions used by the first three theorems are unchanged.

Both reported base controllers satisfy the common conditions of Theorems 1 and 2. Their closed-loop matrices have $\rho(F) = 0.910753$ for Franka and 0.971644 for the quadrotor. With $Q = I$, the associated Lyapunov matrices satisfy $\lambda_{\min}(P) = 1.0000$ and 1.1066 , respectively, and the Lyapunov-equation residuals are 9.45×10^{-13} and 1.27×10^{-12} . The deployed values $\sigma = 0.025$ and 0.75 , together with $c_\delta = 1.5$, satisfy the required ranges, and the reported LyCo-SNN trajectories remain inside the certificate-action box. Hence the pointwise decrement is realized and $a = \sigma\lambda_{\min}(Q)/\lambda_{\max}(P) \in (0,1)$ for both affine predictions.

Both quadrotor protocols use relative context with $L_c = 1$, $\lambda_\Lambda = 0.8$, and $\dot{o}_v = 0.28125$, giving $a_v = 0.18 > 0$. Its deployed checkpoint satisfies $\beta = 5\times10^{-4} < \beta_{\max} = 0.2783$ and $\gamma = 1.54\times10^{-3} > 0$, so the conditions of Theorem 3 hold. Franka uses clipped raw context with $\bar{\xi} = 0.1852$; its deployed checkpoint gives $m_x^{\mathrm{raw}} = 0.02465 > 0$ and $\gamma_{\mathrm{raw}} = 2.14\times10^{-4} > 0$, satisfying the coefficient condition in Theorem 4.

G. Training and model selection

The Franka and general-flight checkpoints are trained in two stages. A bounded short-horizon MPC teacher first generates residual targets relative to the nominal LQR action. In Phase I, ridge regression initializes the decoder. The decoder remains frozen while the encoder is trained with imitation, affine-

margin, event-rate, safe-silence, and active rate-band terms. A surrogate derivative handles the nondifferentiable bipolar spike function during gradient updates [33]. After each optimizer step, the encoder is projected onto the implemented constraint set. Phase II refines closed-loop rollouts with task, affine-margin, filter-intervention, event, saturation, decoder-anchor, and terminal terms. The same projection follows each update. A plant-specific observer-context checkpoint is selected on the matching development and validation protocol. The generic and wrench Franka checkpoints are frozen before their respective held-out evaluations. One frozen quadrotor checkpoint is reused without refitting in both the general-flight and rotor efficiency-loss protocols.

Each comparison actor is fitted on task-specific development data and frozen before evaluation. LyCo-SNN keeps signed events, so its readout can add to or subtract from the nominal command. At deployment, a forward pass updates the observer, generates spikes, decodes the residual, maps it to task coordinates, and applies directional scaling when the controller uses it. No online weight update, decoder refit, or held-out model selection is performed.

IV. Experimental Tasks and Protocol

The study uses four nonlinear continuous-trajectory tasks: generic Cartesian tracking with a Franka Panda, Franka tracking under an external wrench, continuous quadrotor trajectory tracking, and the same flight trajectory during rotor-efficiency loss. All controllers run at $T_s = 0.02$ s and are frozen before held-out evaluation. Within each protocol, the reference, initial condition, plant realization, disturbances, actuator limits, and seed are shared across controllers. The resulting comparisons are paired at the seed level.

A. Franka continuous Cartesian tracking

Both Franka protocols use the official MuJoCo Menagerie Panda model and a continuous end-effector path. Controllers act through seven joint coordinates, and forward kinematics maps the executed joint state to the reported Cartesian error. The reference is

$$p_k^{\mathrm{ref}} = p_0 + \begin{bmatrix} 0.035\sin\phi_k \\ 0.050\sin(2\phi_k) \\ 0.025(\cos\phi_k - 1) \end{bmatrix}, \qquad \phi_k = \frac{2\pi k}{300}. \tag{37}$$

A warm-started damped-least-squares inverse-kinematics solver converts the path to a continuous seven-joint reference. Its maximum Cartesian residual is 0.500 mm. Each protocol uses a 300-sample horizon and ten held-out seeds (6501–6510), with independent position and velocity perturbations at reset.

The generic Franka protocol contains no external wrench. The second protocol uses the same reference construction and adds a world-frame wrench during samples 90–164 and a three-sample impulse during samples 195–197. The two protocols therefore separate nominal Cartesian tracking from rejection of a prescribed external load while retaining the same joint command interface.

B. Quadrotor continuous and fault tracking

Both flight tasks share a cyclic C^2 position trajectory through $(0,0,0)$, $(0.25,-0.20,0.20)$, $(-0.20,0.25,0.30)$, and $(0.20,0.20,0.10)$ m. Each 80-sample segment uses the quintic blend $b(\tau) = 10\tau^3 - 15\tau^4 + 6\tau^5$, giving $H = 320$ samples. The general protocol uses zero yaw, randomizes mass, inertia, and rotor efficiency, and applies no external disturbance. Each held-out seed contains 16 independently randomized episodes.

The fault protocol retains the same trajectory, zero-yaw reference, nominal controller, observer, residual coordinates, actuator box, and 16 episodes per held-out seed. LyCo-SNN uses the same pretrained actor with a validation-selected decoder scale of 3.25, $\sigma = 0.35$, $\rho_{\mathrm{warn}} = 0.5$, and schedule power 2; these values are frozen before held-out evaluation. Only the plant transition changes: rotor 2 operates at 70% of its randomized baseline efficiency during samples 120–229. An 80-ms body-torque impulse $[0.020,-0.015,0.010]^{\top}$ N m is retained during samples 175–178. Escape occurs when position error exceeds 2 m, roll or pitch exceeds 1.05 rad, or angular rate exceeds 8 rad s^{-1}.

C. Baselines and adaptation policy

The Franka comparison contains Observer–LQR, SNN-PID, PopSAN, ILC-SAN, and LyCo-SNN. The quadrotor comparison adds EDTP-SNN-FTSMC. Observer–LQR retains the common nominal controller and observer but disables the neural residual. SNN-PID uses population spikes and a fitted decoder to reproduce a task- coordinate PID correction, following the spike-coded PID principle in the original controller [6]. Pop-SAN and ILC-SAN retain the published actor structures [2], [3]. Their decoders are fitted for the present control interfaces and then frozen.

EDTP-SNN-FTSMC follows the published combination of an event-driven-threshold- plasticity SNN and adaptive fast-terminal sliding-mode control [8]. We adapt its input and output coordinates to the quadrotor reference, actuator limits, and common allocation path. All neural controllers use disjoint development, validation, and held-out partitions. No held-out rollout changes a decoder, checkpoint, gain, threshold, or adapter selection.

D. Metrics and aggregation

For Franka, Cartesian end-effector RMS is the primary endpoint. We also report Position peak, defined as the maximum Euclidean Cartesian error, the within-25-mm fraction, wrench-window RMS where applicable, event occupancy, and Peak force. In Franka, Peak force is the maximum norm of the seven simulated actuator-force entries and is interpreted as a joint-torque norm for the revolute joints. For flight, position RMS is primary. We report attitude RMS, Position peak (the maximum Euclidean position error), Peak force (the maximum effective thrust of an individual rotor), tail-band fraction, fault-window position and attitude RMS, occupancy, and latency. For LyCo-SNN, events per episode

count the absolute bipolar spikes emitted in one rollout. Event occupancy divides this count by network size and episode length. Events per neuron per second also accounts for the 0.02-s sampling period. These measures quantify activity in the software implementation

Three violation rates separate execution stages. The raw affine rate is the fraction of sampled transitions in which the unscaled neural proposal fails the directional-feasibility calculation. The applied affine rate repeats the test after directional scaling and actuator clipping. The simulator decrement rate evaluates the executed transition in the quadratic decrement diagnostic. The last quantity includes plant mismatch, observer error, disturbance, actuator realization, and reference motion.

TABLE I–TABLE IV report the mean and sample standard deviation of ten seed-level values. For general flight, episode metrics are first averaged within each seed. Relative reductions are computed within paired seeds, and 95% intervals use 10,000 bootstrap resamples of the ten paired contrasts. The time-series figures show one fixed held-out rollout from each protocol; the tables provide the aggregate estimates.

Each common ablation removes the SNN residual, negative events, or the observer from the frozen controller. The remaining execution path is unchanged. Table VI reports three task-relevant metrics for all four protocols. Complete and ablated rollouts use identical references, initialization distributions, seeds, and actuator limits within each protocol.

V. Results

A. Aggregate performance and controller diagnostics

On generic Franka Cartesian tracking (TABLE I), LyCo-SNN reached 9.329 mm Cartesian RMS and reduced the corresponding Observer–LQR value by 17.33%. Pop-SAN reached the lowest RMS in this protocol at 8.910 mm. LyCo-SNN retained 99.90% of samples within 25 mm, the largest fraction among the compared controllers. These two endpoints separate mean precision from error containment: Pop-SAN minimizes average error, whereas LyCo-SNN keeps more of the trajectory inside the prescribed band. LyCo-SNN uses more events and latency than SNN-PID to obtain this tracking gain, but its 1.259-ms p95 latency remains below Pop-SAN and ILC-SAN.

Under the external-wrench protocol (TABLE II), LyCo-SNN achieved 24.603 mm Cartesian RMS and 42.828 mm wrench-window RMS. Its Cartesian RMS was 11.187% below Pop-SAN, the strongest SNN comparator in this protocol. It also achieved the lowest Position peak and kept 77.50% of samples within 25 mm. Improvement across the full rollout, load window, and peak error shows that the residual correction acts throughout the loaded segment rather than at one isolated sample. Its 29.53% occupancy lies between SNN-PID at 14.14% and the denser Pop-SAN and ILC-SAN responses at 66.24% and 35.90%.

Actuator demand qualifies the Franka tracking gains. In generic tracking, LyCo-SNN reaches 54.802 ± 2.091 N m, above Observer–LQR and Pop-SAN but below ILC-SAN. Its tighter tracking therefore uses stronger transient joint corrections than the nominal controller. Under the wrench, the LyCo-SNN peak of 99.563±0.505 N m remains within the comparator range and below ILC-SAN at 110.423 ± 11.148 N m. The wrench result is distinguished by how the command is applied over the load interval, not by the largest peak demand.

On general flight (TABLE III), LyCo-SNN reduced position RMS from the SNN-PID value of 0.1397 to 0.1337 m, a paired reduction of 4.345%. It also achieved the lowest Position peak and the largest tail-band fraction. ILC-SAN instead achieved the lowest attitude RMS, 4.165°, compared with 4.331° for LyCo-SNN. This split shows that the LyCo-SNN advantage lies in translational tracking rather than attitude regulation, consistent with the component ablation in Table VI. Its 7.48% occupancy is the lowest among the neural controllers, so the position gain is obtained without increasing event activity.

During the 30% rotor-efficiency loss (TABLE IV), LyCo-SNN achieved the lowest episode-wide position RMS, 0.1093 m, followed by SNN-PID at 0.1281 m, and extended this lead to the fault window (0.1280 versus 0.1561 m). Its 0.2277-m Position peak and 58.11% tail-band fraction show that the improvement persists beyond the average endpoint. Peak force rises only from 3.827 ± 0.144 N for Observer–LQR to 3.941 ± 0.191 N for LyCo-SNN, linking the position gain to residual allocation rather than a large increase in individual-rotor thrust. The attitude ordering remains different: ILC-SAN leads episode-wide attitude RMS, while SNN-PID leads inside the fault window. EDTP-SNN-FTSMC combines 61.85% occupancy with the largest position errors, showing that higher event activity alone does not yield more accurate fault tracking.

TABLE V connects these task-level results to LyCo-SNN event allocation and directional scaling.

The generic Franka controller emitted 20,405±4,115 events per episode at 53.14±10.72% occupancy. The wrench protocol emitted 11,338 ± 233 events at 29.53 ± 0.61%. Their task-specific checkpoints therefore use different event budgets despite sharing the same residual architecture. The flight pair gives a direct within-actor comparison. Rotor loss increases events from 3,066 ± 125 to 3,879 ± 108 and occupancy from7.48 ± 0.31% to 9.47 ± 0.26%. The frozen 128-neuron actor thus responds to actuator loss by raising event activity by approximately 27%.

The executed directional scale was 0.209 ± 0.061 in generic Franka tracking, 0.487 ± 0.011 under the wrench, and 0.997 ±0.000 in general flight. Under rotor loss, candidate and executed values were 0.976 ± 0.005. The low Franka scales show that event generation and executed residual magnitude are distinct: frequent spikes can still produce a strongly attenuated correction. Both flight scales remain near one, so their performance difference is driven mainly by the changed event-coded residual rather than repeated directional attenuation. Applied affine violations remain zero in all four protocols. The simulator decrement rate increases from 0.51%±0.53% in general flight to 6.88%±3.76% under rotor loss, separating the affine filter outcome from the downstream plant-side efficiency change.

TABLE I

FRANKA GENERIC CARTESIAN TRACKING WITHOUT AN EXTERNAL WRENCH. VALUES ARE SEED-LEVEL MEAN ± S.D. OVER TEN HELD-OUT SEEDS. THE 25-MM BAND IS THE FRACTION OF SAMPLES WITHIN THE PRESCRIBED CARTESIAN-ERROR BOUND. PEAK FORCE IS THE NORM OF THE ACTUATOR-FORCE VECTOR.

Controller	Cartesian RMS (mm)↓	Peak error (mm)↓	Within 25 mm (%)↑	SNN occupancy (%)↓	Peak force (N·m)↓	P95 latency (ms)↓
Observer-LQR	11.291 ± 0.395	17.393 ± 3.546	99.77 ± 0.74	–	**41.024 ± 0.222**	**0.002 ± 0.000**
SNN-PID	12.677 ± 1.840	21.052 ± 3.563	98.63 ± 4.32	**12.96 ± 0.85**	47.683 ± 0.945	0.217 ± 0.027
PopSAN	**8.910 ± 0.416**	**15.508 ± 4.346**	99.83 ± 0.53	66.85 ± 0.25	43.612 ± 0.811	3.716 ± 0.422
ILC-SAN	12.129 ± 0.928	24.081 ± 2.877	99.43 ± 0.79	36.74 ± 0.16	72.709 ± 34.525	4.273 ± 0.108
LyCo-SNN	9.329 ± 0.615	15.782 ± 4.218	**99.90 ± 0.32**	53.14 ± 10.72	54.802 ± 2.091	1.259 ± 0.080

TABLE II

FRANKA CARTESIAN TRACKING UNDER THE FIXED EXTERNAL-WRENCH PROTOCOL. VALUES ARE SEED-LEVEL MEAN ± S.D. OVER TEN HELD-OUT SEEDS. PEAK FORCE IS THE NORM OF THE ACTUATOR-FORCE VECTOR.

Controller	Cartesian RMS (mm)↓	Wrench RMS (mm)↓	Peak error (mm)↓	Within 25 mm (%)↓	Peak force (N·m)↓	SNN occupancy (%)↓	P95 latency (ms)↓
Observer-LQR	29.801 ± 0.154	52.246 ± 0.000	61.577 ± 0.000	73.43 ± 0.74	**95.511 ± 0.001**	–	**0.002 ± 0.000**
SNN-PID	27.883 ± 2.046	48.159 ± 4.298	56.449 ± 5.523	72.70 ± 3.06	97.845 ± 1.313	**14.14 ± 0.72**	0.236 ± 0.020
PopSAN	27.703 ± 0.202	49.822 ± 0.175	58.616 ± 0.429	74.37 ± 0.51	96.606 ± 0.671	66.24 ± 0.19	6.046 ± 1.583
ILC-SAN	27.768 ± 0.580	48.620 ± 1.142	62.856 ± 1.331	74.17 ± 0.85	110.423 ± 11.148	35.90 ± 0.18	7.190 ± 0.334
LyCo-SNN	**24.603 ± 0.228**	**42.828 ± 0.065**	**56.157 ± 0.008**	**77.50 ± 0.45**	99.563 ± 0.505	29.53 ± 0.61	1.377 ± 0.103

TABLE III

QUADROTOR CONTINUOUS C^2 TRAJECTORY TRACKING WITH RANDOMIZED PLANT PARAMETERS AND NO EXTERNAL DISTURBANCE. VALUES ARE SEED-LEVEL MEAN ± S.D. OVER TEN HELD-OUT SEEDS; EACH SEED CONTAINS 16 EPISODES. ALL RUNS COMPLETED WITHOUT ESCAPE.

Controller	Position RMS (m)↓	Attitude RMS (°)↓	Position peak (m)↓	Peak force (N)	Tail band (%)↑	SNN occupancy (%)↓	P95 latency (ms)↓
Observer-LQR	0.1411 ± 0.0011	4.246 ± 0.049	0.2705 ± 0.0025	**3.572 ± 0.253**	40.00 ± 0.00	–	**0.008 ± 0.000**
SNN-PDF	0.1397 ± 0.0012	4.222 ± 0.042	0.2670 ± 0.0019	3.926 ± 0.288	39.55 ± 0.12	18.21 ± 0.14	0.026 ± 0.003
PopSAN	0.1566 ± 0.0016	4.179 ± 0.050	0.2942 ± 0.0028	3.908 ± 0.307	26.02 ± 0.71	70.34 ± 0.29	0.498 ± 0.022
ILC-SAN	0.1521 ± 0.0013	**4.165 ± 0.047**	0.2921 ± 0.0022	3.910 ± 0.298	26.80 ± 0.53	56.73 ± 0.43	0.621 ± 0.028
EDTP-SNN-FTSMC	0.1847 ± 0.0028	5.586 ± 0.118	0.3266 ± 0.0044	4.528 ± 0.251	38.53 ± 1.03	25.96 ± 2.03	0.167 ± 0.012
LyCo-SNN	**0.1337 ± 0.0010**	4.331 ± 0.045	**0.2560 ± 0.0022**	3.605 ± 0.271	**42.50 ± 0.00**	**7.48 ± 0.31**	0.167 ± 0.012

TABLE IV

QUADROTOR CONTINUOUS C^2 TRAJECTORY TRACKING DURING A 30% PLANT-SIDE ROTOR-2 EFFICIENCY LOSS AND A BODY-TORQUE IMPULSE. THE CONTROLLER STRUCTURE, PRETRAINED ACTOR, TRAJECTORY, INITIAL CONDITIONS, AND PLANT RANDOMIZATION MATCH TABLE III; THE LYCO-SNN DECODER AND FIRING SCHEDULE ARE SELECTED ON THE DISJOINT VALIDATION SPLIT AND FROZEN BEFORE EVALUATION. VALUES ARE SEED-LEVEL MEAN ± S.D. OVER TEN HELD-OUT SEEDS, EACH CONTAINING 16 EPISODES. PEAK FORCE IS THE MAXIMUM EFFECTIVE INDIVIDUAL-ROTOR THRUST.

Controller	Position RMS (m)↓	Attitude RMS (°)↓	Position peak (m)↓	Fault-position RMS (m) ↓	Fault-attitude RMS (◦) ↓	Peak force (N)	Tail band (%)↑	SNN occupancy (%)↓	P95 latency (ms)↓
Observer-LQR	0.1305 ± 0.0024	4.261 ± 0.073	0.2757 ± 0.0025	0.1583 ± 0.0051	4.499 ± 0.197	**3.827 ± 0.144**	40.50 ± 1.97	–	**0.007 ± 0.000**
SNN-PDF	0.1281 ± 0.0024	4.130 ± 0.043	0.2713 ± 0.0018	0.1561 ± 0.0065	**4.239 ± 0.065**	4.012 ± 0.206	41.66 ± 1.22	18.51 ± 0.11	0.021 ± 0.001
PopSAN	0.1407 ± 0.0037	4.171 ± 0.047	0.2997 ± 0.0027	0.1764 ± 0.0084	4.396 ± 0.096	4.035 ± 0.218	33.80 ± 1.92	70.80 ± 0.55	0.420 ± 0.004
ILC-SAN	0.1392 ± 0.0024	**4.107 ± 0.055**	0.2976 ± 0.0022	0.1732 ± 0.0066	4.295 ± 0.107	4.024 ± 0.194	33.83 ± 1.89	57.13 ± 0.85	0.536 ± 0.044
EDTP-SNN-FTSMC	0.2947 ± 0.0571	6.725 ± 0.474	0.5111 ± 0.0896	0.3488 ± 0.1081	5.061 ± 0.882	4.528 ± 0.251	0.00 ± 0.00	61.85 ± 4.91	0.136 ± 0.005
LyCo-SNN	**0.1093 ± 0.0025**	4.382 ± 0.036	**0.2277 ± 0.0021**	**0.1280 ± 0.0068**	4.601 ± 0.044	3.941 ± 0.191	**58.11 ± 1.04**	**9.47 ± 0.26**	0.123 ± 0.015

TABLE V

LYCO-SNN NEURAL ACTIVITY AND DIRECTIONAL-EXECUTION DIAGNOSTICS. VALUES ARE SEED-LEVEL MEAN ± S.D. OVER TEN HELD-OUT SEEDS. CANDIDATE AND EXECUTED H COINCIDE WHEN THE DIRECTIONAL SCALE IS APPLIED AT THE PLANT INTERFACE.

Task	Neurons	Events/ episode	Occupancy (%)	Events/ neuron/s	Candidate η	Executed η
Franka generic Cartesian	128	20405 ± 4115	53.14 ± 10.72	26.57 ± 5.36	0.209 ± 0.061	0.209 ± 0.061
Franka Cartesian + wrench	128	11338 ± 233	29.53 ± 0.61	14.76 ± 0.30	0.487 ± 0.011	0.487 ± 0.011
Quadrotor continuous	128	3066 ± 125	7.48 ± 0.31	3.74 ± 0.15	0.997 ± 0.000	0.997 ± 0.000
Quadrotor + rotor fault	128	3879 ± 108	9.47 ± 0.26	4.73 ± 0.13	0.976 ± 0.005	0.976 ± 0.005

B. Closed-loop trajectories and command responses

Figure 2–Figure 5 trace one fixed held-out rollout from each continuous-reference protocol. They complement the aggregate tables with paths, errors, velocities, and applied commands from the same closed-loop run. The command coordinates differ by task. Franka uses seven joint actuators, while both flight protocols use the same four rotor commands.

Figure 2 compares all controller paths against the same reference. In generic Franka tracking, PopSAN and LyCoSNN stay closest to the Cartesian path, followed by ILCSAN, Observer–LQR, and SNN-PID. Under the wrench, the LyCo-SNN path has the smallest visible departure from the reference, consistent with its lowest aggregate Cartesian RMS (24.603 mm) and wrench-window RMS (42.828 mm) in Table II. The quadrotor paths show the same comparison in flight

coordinates. In general flight, LyCo-SNN remains closest to the cyclic reference, SNN-PID and Observer–LQR form the next group, Pop-SAN and ILC-SAN deviate more, and EDTP-SNN-FTSMC has the widest path. With rotor-2 at 70% efficiency and the torque impulse applied, LyCo-SNN still has the smallest translational departure. SNN-PID and Observer– LQR follow, Pop-SAN and ILC-SAN spread farther from the path, and EDTP-SNN-FTSMC produces the largest excursion. These representative paths agree with the position-RMS and Position-peak rankings in Tables III and IV.

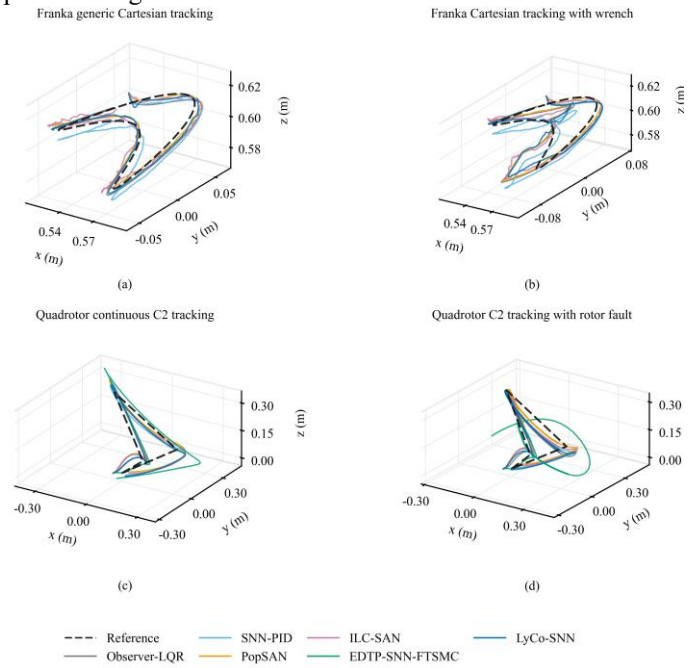


Figure 2 Fixed held-out three-dimensional reference and controller trajectories for (a) Franka generic Cartesian tracking without an external wrench, (b) Franka Cartesian tracking with the prescribed wrench and pulse, (c) general quadrotor tracking, and (d) quadrotor tracking with rotor-2 efficiency loss and a torque impulse.

Figure 3 resolves the trajectory differences over time. For generic Franka tracking, Pop-SAN has the lowest representative Cartesian-error trace and LyCo-SNN is next; both remain below the Observer–LQR and SNN-PID traces for most of the rollout. During the wrench interval, LyCo-SNN recovers below the other four Franka controllers and retains the lowest aggregate error. In general flight, LyCo-SNN gives the smallest position-error trace, whereas ILC-SAN gives the lower attitude-error trace. In fault flight, LyCo-SNN again gives the smallest position-error trace through the fault window, SNNPID has the lower attitude error, and EDTP-SNN-FTSMC shows the largest position and attitude excursions.

Figure 4 compares the realized motion behind these tracking errors. In generic Franka tracking, Observer–LQR, Pop-SAN, and SNN-PID remain near the reference-speed range, while LyCo-SNN and ILC-SAN exhibit brief higher peaks. Under the wrench, all five controllers accelerate during the load interval; their peak speeds occupy a narrow 0.48–0.51 m s^{-1} range, with LyCo-SNN returning to the common post-load range. In general flight, Observer–LQR, SNN-PID, Pop-SAN, ILCSAN, and LyCo-SNN have closely overlapping linear- and angular-speed profiles. EDTP-SNN-FTSMC has the largest linear-speed excursion. The rotor-loss protocol separates the profiles more clearly: the five LQR-based and neural-residual controllers remain in a similar range, whereas EDTP-SNNFTSMC reaches the largest linear and angular speeds. LyCo-SNN follows the shared range while achieving the lowest position error in Figure 3.

Applied-control histories in Figure 5 show the command demand associated with each response. In generic Franka tracking, LyCo-SNN reaches a 53.08 N m peak, above Observer– LQR, Pop-SAN, and SNN-PID and close to ILC-SAN. Under the wrench, all five command norms rise sharply. LyCo-SNN peaks at 99.36 N m, between Observer–LQR, SNN-PID, and Pop-SAN below it and ILC-SAN above it, while producing the lowest wrench-window error. In general flight, the LyCo-SNN command norm peaks at 1.49 N, close to Observer–LQR at 1.47 N and below SNN-PID Pop-SAN, and ILC-SAN. The EDTP-SNN-FTSMC trace has a distinct 4.34 N peak. During rotor loss, LyCo-SNN increases its peak command norm to 1.97 N; Observer–LQR, SNN-PID, Pop-SAN, and ILC-SAN remain between 1.50 and 1.80 N, while EDTP-SNN-FTSMC again reaches 4.34 N. These values are four-rotor command norms; the Peak force columns in TABLE III and TABLE IV report the maximum effective thrust of one rotor.

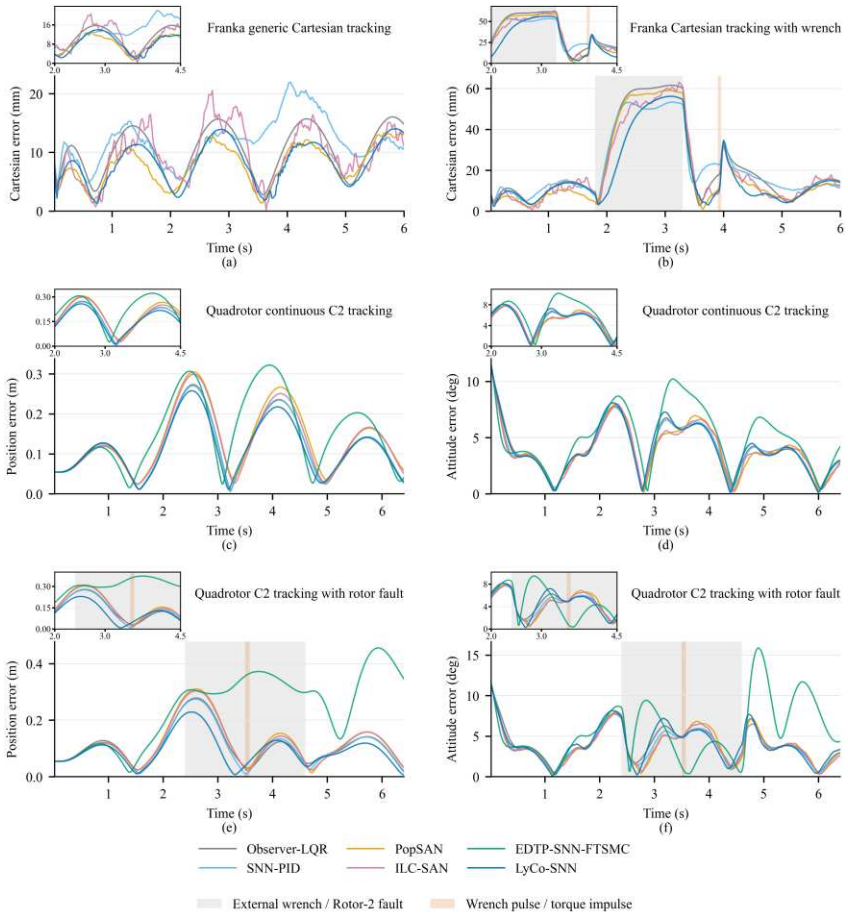


Figure 3. Representative tracking-error histories for all compared controllers. The Franka panels show Cartesian error norms. The quadrotor panels show position and attitude errors. The shaded intervals identify the prescribed wrench, rotor fault, or moment impulse in the affected protocols.

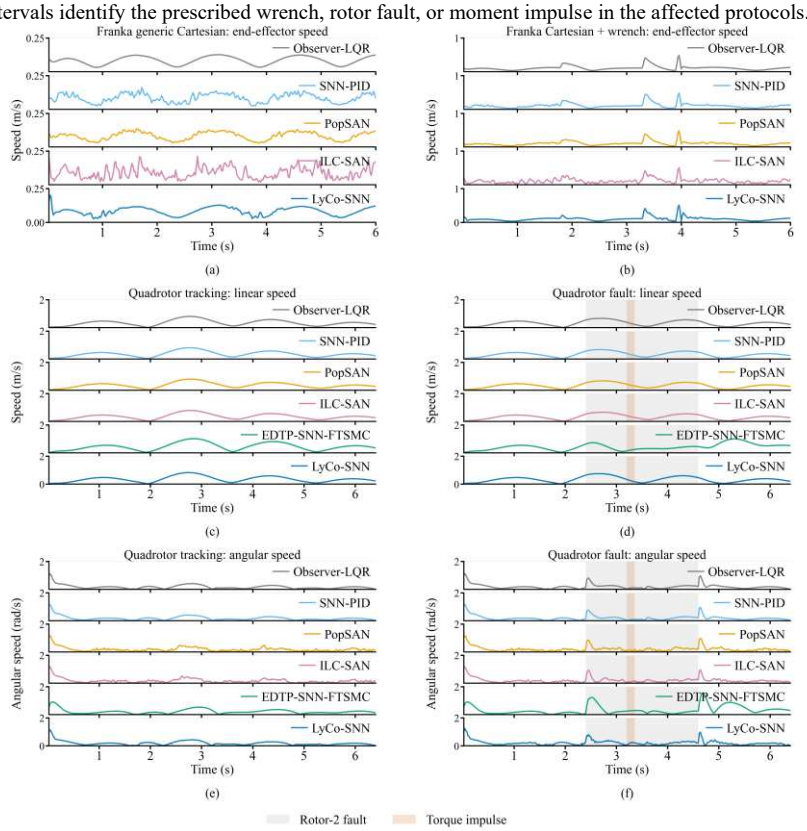


Figure 4. Representative realized velocity outputs for all compared controllers. The Franka panels show end-effector speed. The quadrotor panels show linear and angular speed. Only controller-specific velocity magnitudes are plotted.

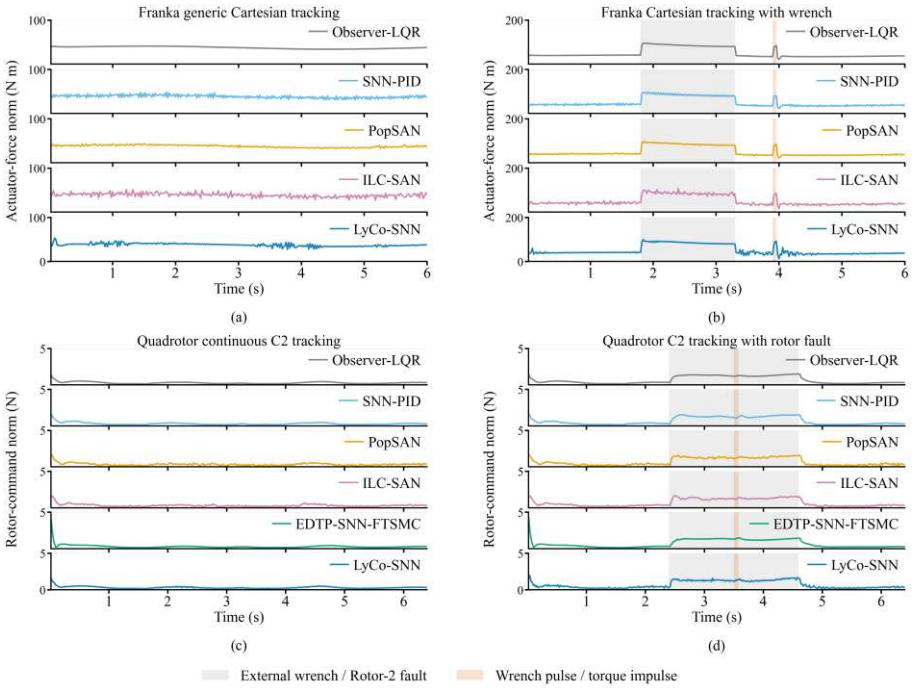


Figure 5. Representative applied-control histories for all compared controllers. Franka effort is the Euclidean norm of the seven actuator-force values. Quadrotor effort is the Euclidean norm of the four applied rotor-command values.

C. Component ablation and neural activity

TABLE VI tests each execution component under matched conditions. Removing the SNN residual increased generic Franka Cartesian RMS from 9.329 to 11.291 mm (21.03%) and Position peak from 15.782 to 17.393 mm (10.21%), while Within 25 mm remained nearly unchanged at 99.77%. Under the wrench, Cartesian RMS and Wrench RMS increased by 21.13% and 21.99%, while Within 25 mm decreased from 77.50% to 73.43%. The residual therefore improved fine generic-task Cartesian tracking and preserved tracking accuracy and in-band operation under the wrench.

The quadrotor results showed the same position-oriented effect. In general flight, residual removal increased Position RMS and Position peak by 5.53% and 5.66%, while Tail band decreased from 42.50% to 40.00%. Under rotor-efficiency loss, episode-wide Position RMS increased from 0.1093 to 0.1305 m, Fault-position RMS from 0.1280 to 0.1583 m, and Position peak from 0.2277 to 0.2757 m; Tail band fell from 58.11% to 40.50%. Conversely, episode-wide Attitude RMS decreased from 4.382° to 4.261°, Fault-attitude RMS from 4.601° to 4.499°, and Peak force from 3.941 to 3.827 N. The complete controller thus exchanged small increases in attitude error and rotor thrust for substantially better translational tracking and fault-window recovery.

Restricting the network to positive events produced a smaller but consistent loss of translational accuracy. Generic Franka Cartesian RMS and Position peak increased by 5.69% and 7.00%, while Within 25 mm remained at 99.90%. Under the wrench, Cartesian RMS and Wrench RMS increased by 6.62% and 7.93%, while Within 25 mm decreased to 74.83%. In general flight, Position RMS and Position peak increased by 2.62% and 2.38%, while Tail band decreased to 40.00%. Under rotor-efficiency loss, episode-wide Position RMS, Fault-position RMS, and Position peak increased by 6.22%, 8.28%, and 7.03%, while Tail band decreased from 58.11% to 54.00%. The positive-only variant achieved slightly lower attitude errors and nearly the same Peak force, indicating that the negative-event channel mainly strengthened translational correction.

Observer removal caused the largest and broadest degradation. In generic Franka tracking, Cartesian RMS and Position peak increased by 73.51% and 49.27%, while Within 25 mm remained saturated at 99.90%. Under the wrench, Cartesian RMS and Wrench RMS increased by 41.58% and 34.49%, while Within 25 mm fell to 72.77%. In general flight, Position RMS and Position peak increased by 10.02% and 5.90%, while Tail band decreased from 42.50% to 22.81%. Under rotor-efficiency loss, episode-wide Position RMS and Attitude RMS increased by 41.26% and 14.99%; Fault-position RMS, Fault-attitude RMS, and Position peak increased to 0.1551 m, 5.750°, and 0.2675 m; and Tail band decreased to 37.77%. Peak force changed only from 3.941 to 3.952 N. These results show that the observer primarily compensated for reference and execution mismatch without materially increasing peak actuator demand.

TABLE VI

PROTOCOL-MATCHED LYCO-SNN COMPONENT ABLATIONS ACROSS ALL FOUR CONTINUOUS-TRAJECTORY TASKS. VALUES ARE MEAN ± S.D. OVER TEN HELD-OUT SEEDS. THE COMPLETE CONTROLLER USES THE FROZEN TASK CHECKPOINT. EACH ABLATION CHANGES ONLY THE NAMED EXECUTION COMPONENT AND KEEPS THE REFERENCE, INITIAL CONDITIONS, PLANT REALIZATION, AND ACTUATOR LIMITS FIXED. GENERAL-FLIGHT METRICS COVER OVERALL POSITION AND ATTITUDE TRACKING, WORST-CASE POSITION ERROR, AND PEAK INDIVIDUAL-ROTOR THRUST. THE FAULT BLOCK REUSES THE SAME FROZEN QUADROTOR CHECKPOINT AND FOUR-ROTOR CONTROLLER; ONLY THE PLANT-SIDE EFFICIENCY LOSS AND PRESCRIBED TORQUE IMPULSE ARE ADDED. FAULT-FLIGHT RMS VALUES ARE RESTRICTED TO THE FAULT WINDOW, WHILE POSITION PEAK AND PEAK FORCE COVER THE COMPLETE ROLLOUT.

Variant	Franka generic Cartesian			Franka Cartesian + wrench		
	Cartesian RMS (mm) ↓	Position peak (mm) ↓	Within 25 mm (%) ↑	Cartesian RMS (mm) ↓	Wrench RMS (mm) ↓	Within 25 mm (%) ↑
Complete LyCo-SNN	**9.329 ± 0.615**	**15.782 ± 4.218**	**99.90 ± 0.32**	**24.603 ± 0.228**	**42.828 ± 0.065**	**77.50 ± 0.45**
w/o SNN residual	11.291 ± 0.395	17.393 ± 3.546	99.77 ± 0.74	29.801 ± 0.154	52.246 ± 0.000	73.43 ± 0.74
positive-only events	9.860 ± 0.511	16.886 ± 4.148	99.90 ± 0.32	26.232 ± 0.154	46.226 ± 0.104	74.83 ± 0.42
w/o observer	16.187 ± 1.033	23.558 ± 1.502	99.90 ± 0.32	34.834 ± 0.142	57.599 ± 0.021	72.77 ± 0.69

	Quadrotor continuous				
	Position RMS (m) ↓	Attitude RMS (◦) ↓	Position peak (m) ↓	Peak force (N) ↓	Tail band (%) ↑
Complete LyCo-SNN	**0.1337 ± 0.0010**	4.331 ± 0.045	**0.2560 ± 0.0022**	3.605 ± 0.271	**42.50 ± 0.00**
w/o SNN residual	0.1411 ± 0.0011	**4.246 ± 0.049**	0.2705 ± 0.0025	**3.572 ± 0.253**	40.00 ± 0.00
positive-only events	0.1372 ± 0.0010	4.290 ± 0.047	0.2621 ± 0.0023	3.590 ± 0.261	40.00 ± 0.00
w/o observer	0.1471 ± 0.0078	4.374 ± 0.095	0.2711 ± 0.0242	3.605 ± 0.271	22.81 ± 20.45

	Quadrotor + rotor-2 efficiency loss						
	Position RMS (m) ↓	Attitude RMS (◦) ↓	Fault-position RMS (m) ↓	Fault-attitude RMS (◦) ↓	Position peak (m) ↓	Peak force (N) ↓	Tail band (%) ↑
Complete LyCo-SNN	**0.1093 ± 0.0025**	4.382 ± 0.036	**0.1280 ± 0.0068**	4.601 ± 0.044	**0.2277 ± 0.0021**	3.941 ± 0.191	**58.11 ± 1.04**
w/o SNN residual	0.1305 ± 0.0024	**4.261 ± 0.073**	0.1583 ± 0.0051	4.499 ± 0.197	0.2757 ± 0.0025	**3.827 ± 0.144**	40.50 ± 1.97
positive-only events	0.1161 ± 0.0026	4.289 ± 0.047	0.1386 ± 0.0069	**4.446 ± 0.093**	0.2437 ± 0.0019	3.926 ± 0.185	54.00 ± 1.29
w/o observer	0.1544 ± 0.0223	5.039 ± 0.133	0.1551 ± 0.0310	5.750 ± 0.241	0.2675 ± 0.0440	3.952 ± 0.156	37.77 ± 15.58

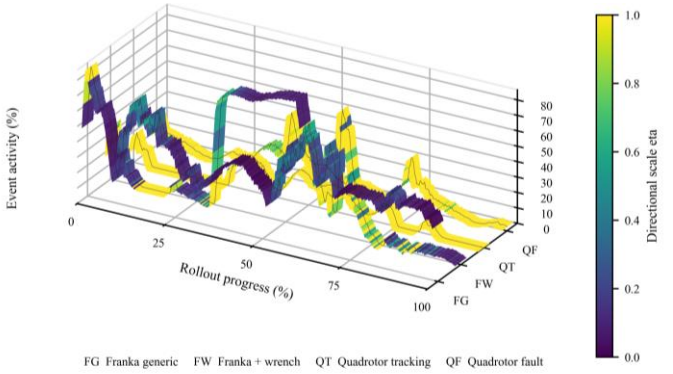


Figure 6. Fixed held-out LyCo-SNN internal activity across the four tasks. Surface height denotes per-step event activity and color denotes the directional scale η. Both flight lanes use the same 128-neuron actor and four-rotor residual coordinates.

Temporal activity in Figure 6 varied across the rollout and task. The generic and wrench-loaded Franka protocols averaged 53.14% and 29.53% occupancy, versus 7.48% in general flight and 9.47% under rotor loss. Event activity and executed scale followed different temporal patterns. Mean executed scales were 0.209 and 0.487 for Franka, 0.997 for general flight, and 0.976 under rotor loss. Event generation determines when the neural residual is active, whereas directional scaling determines how much reaches the plant. Because both flight protocols use the same 128-neuron actor, their occupancy and scale values are directly comparable.

Figure 7 resolves this behavior at seed level. Panel (a) shows that occupancy changed with within-task demand ranking, although profiles differed between protocols. Separate rankings were required because joint-space and rotor-space demands have different physical scales. Panel (b) shows substantial attenuation in Franka, while flight scales remained close to one. In both flight protocols, scaling preceded plant execution; under rotor loss, the efficiency reduction followed.

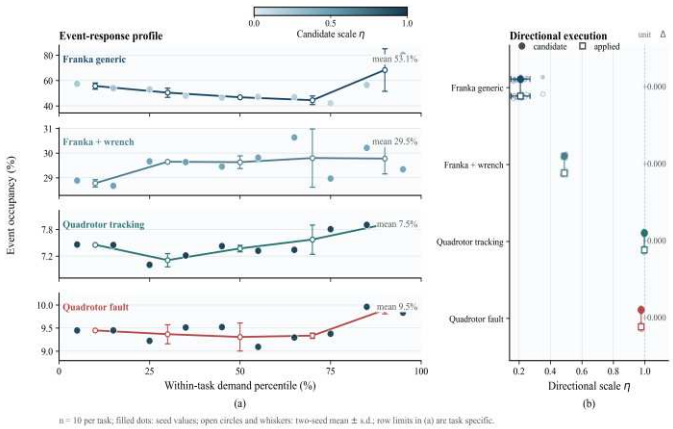


Figure 7. Cross-protocol LyCo-SNN activity and directional execution. (a) Seed-level occupancy ordered by a task-specific tracking-demand metric. (b) Seedlevel directional scales with mean ± s.d. over ten held-out seeds. FG, FW, QT, and QF denote Franka generic, Franka with wrench, quadrotor tracking, and quadrotor tracking with rotor loss.

VI. Discussion

The experiments linked LyCo-SNN performance to a specific division of labor. The observer estimated an action-equivalent mismatch, bipolar events encoded a signed residual, and the affine directional filter determined how much reached the nominal controller. Removing the residual or observer increased primary tracking error in every task, while positive-only events weakened translational correction. Mean directional scales of 0.209 and 0.487 showed substantial attenuation in Franka, whereas values of 0.997 and 0.976 admitted most flight residuals. The architecture therefore adapts through event allocation and execution scaling, but their roles depend on the command interface and operating condition.

The most consequential result was retained tracking accuracy under external load and actuator loss. LyCo-SNN achieved the lowest Cartesian and wrench-window RMS in loaded Franka tracking, at 24.603 and 42.828 mm. Position RMS reached

0.1337 m in general flight and 0.1093 m under rotor loss, with 0.1280 m in the fault window. Fault-protocol peak force was 3.941 N for LyCo-SNN and 3.827 N for Observer–LQR. These findings indicate that the residual primarily improves translational correction without replacing structured feedback or relying on the largest individual-rotor thrust. The benefit was not uniform: Pop-SAN retained the lowest generic Franka RMS, and other spiking controllers retained lower flight attitude error.

Several observations agree with earlier research. Residual learning was introduced to preserve a structured controller while learning only the missing correction [1]; the Observer–LQR and residual-removal comparisons supported this principle in all four protocols. Pop-SAN and ILC-SAN established population-coded spiking actors as practical continuous-action models [2], [3]. Their strong generic Franka and attitude-tracking results supported that conclusion under the present interfaces. SNN-PID also retained competitive attitude regulation during rotor loss, consistent with demonstrations that spike-coded PID preserves a structured feedback law [6]. LyCo-SNN extends these ideas by assigning the spiking network a residual role and auditing its decoded action before execution.

The differences from prior results were task dependent. Pop-SAN achieved lower generic Franka RMS, whereas LyCo-SNN led under the external wrench. This reversal suggests that a full population-coded actor is effective for a smooth disturbance-free path, while observer-conditioned residual correction becomes more useful when the nominal model omits a sustained load. ILC-SAN and SNN-PID achieved lower flight attitude error, and the LyCo-SNN ablation showed the same tension: removing the residual slightly improved attitude metrics but degraded position tracking. The residual allocation therefore favored translational correction in the reported flight objective. EDTP-SNN-FTSMC was originally formulated as an SNN approximation within a complete adaptive fast-terminal sliding-mode attitude controller [8]. Here it was mapped to a shared rotor interface and evaluated on simultaneous three-dimensional position and attitude tracking. The original study assessed a complete attitude-control composite, whereas the present comparison tests its adapted form within a common trajectory-tracking interface.

The directional diagnostics clarify the reported certificate's scope. The filter produced zero applied affine violations in every protocol, consistent with runtime correction methods enforcing a declared model-level constraint [5]. Under rotor loss, the simulator decrement rate rose from 0.51% to 6.88% even though the applied affine rate remained zero. This separation shows that the filter controlled the certificate-space residual as designed, while downstream effectiveness changes altered the realized plant transition. It also agrees with prior assessments that learned-certificate claims remain tied to their model, domain, and verification assumptions [28], [29].

The evidence is confined to four simulated protocols, their trajectories and randomization ranges, and ten held-out seeds. Software occupancy measures neural activity rather than platform energy, while host latency does not represent neuromorphic deployment. The comparator actors retain their published structures but use task-adapted decoders and interfaces; the tables therefore compare complete deployed controllers rather than isolated network architectures. Pop-SAN was more accurate in generic Franka tracking, ILC-SAN and SNN-PID were stronger on the reported attitude endpoints, and LyCo-SNN used higher peak joint demand than Observer–LQR in generic tracking. These trade-offs bound the main claim to observer-conditioned residual correction of Cartesian and translational tracking under the evaluated disturbances and actuator loss.

VII. Conclusion

Spiking residual control poses a coordination problem: a learned correction must remain meaningful across plant command coordinates while preserving nominal feedback action. This paper addressed that problem by separating observer-conditioned residual generation, bipolar event allocation, task-coordinate mapping, and Lyapunov-constrained directional scaling. Within the evaluated tasks, this separation allowed one residual-control principle to operate through seven Franka joint corrections and four quadrotor rotor corrections. The evidence showed that LyCo-SNN is most effective as a structured correction. It reduced Observer–LQR's primary tracking error in all four protocols and led under the external wrench and rotor-efficiency loss. Ablations tied these gains to the residual, observer context, and negative-event channel. Franka required substantial directional attenuation, whereas both flight tasks executed residuals near their candidate scale. Event allocation and execution scaling therefore serve distinct functions.

The main contribution is a controller architecture in which an SNN augments structured feedback while keeping its decoded action visible at the plant interface. Recording the candidate residual, event activity, applied scale, and affine diagnostic connects continuous spiking control with residual learning and runtime action auditing. The observed position-attitude trade-off defines the method's practical role: Cartesian and translational correction under mismatch, rather than uniform improvement of every metric.

The conclusions are bounded by four simulated protocols, task-adapted comparison interfaces, and software measurements of event activity and latency. Hardware measurements can establish how recorded event allocation translates into platform energy and timing, while broader trajectory and actuator-loss regimes can test how far the residual principle extends beyond the reported conditions.

Acknowledgements

This work was supported by no Fund.

Conflict-of-Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] T. Johannink *et al.*, "Residual reinforcement learning for robot control," in *2019 international conference on robotics and automation (ICRA)*, May 2019, pp. 6023–6029. doi: 10.1109/ICRA.2019.8794127.

[2] G. Tang, N. Kumar, R. Yoo, and K. P. Michmizos, "Deep reinforcement learning with population-coded spiking neural network for continuous control," in *Proceedings of the 2020 conference on robot learning*, in Proceedings of machine learning research, vol. 155. PMLR, 2021, pp. 2016–2029.

[3] D. Chen, P. Peng, T. Huang, and Y. Tian, "Fully spiking actor network with intralayer connections for reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 2, pp. 2881–2893, Feb. 2025, doi: 10.1109/TNNLS.2024.3352653.

[4] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel, "Backpropagation-based learning techniques for deep spiking neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11906–11921, Sep. 2024, doi: 10.1109/TNNLS.2023.3263008.

[5] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, Jul. 2021, doi: 10.1016/j.automatica.2021.109597.

[6] R. K. Stagsted, A. Vitale, J. Binz, A. Renner, L. B. Larsen, and Y. Sandamirskaya, "Towards neuromorphic control: A spiking neural network based PID controller for UAV," in *Robotics: Science and systems XVI*, Jul. 2020. doi: 10.15607/RSS.2020.XVI.074.

[7] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE conference on decision and control*, 2010, pp. 5420–5425. doi: 10.1109/CDC.2010.5717652.

[8] W. Yu *et al.*, "Fault-tolerant attitude tracking control driven by spiking NNs for unmanned aerial vehicles," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 2, pp. 3773–3785, Feb. 2025, doi: 10.1109/TNNLS.2023.3342078.

[9] B. Tearle, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "A predictive safety filter for learning-based racing control," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7635–7642, 2021, doi: 10.1109/LRA.2021.3097073.

[10] Z. Bing, C. Meschede, F. Röhrbein, K. Huang, and A. C. Knoll, "A survey of robotics control based on learning-inspired spiking neural networks," *Frontiers in Neurorobotics*, vol. 12, p. 35, Jul. 2018, doi: 10.3389/fnbot.2018.00035.

[11] H. Hagras, A. Pounds-Cornish, M. Colley, V. Callaghan, and G. Clarke, "Evolving spiking neural network controllers for autonomous robots," in *Proceedings of the 2004 IEEE international conference on robotics and automation*, 2004, pp. 4620–4626. doi: 10.1109/ROBOT.2004.1302446.

[12] J. Pérez, J. A. Cabrera, J. J. Castillo, and J. M. Velasco, "Bio-inspired spiking neural network for nonlinear systems control," *Neural Networks*, vol. 104, pp. 15–25, Aug. 2018, doi: 10.1016/j.neunet.2018.04.002.

[13] J. P. Fernández, M. A. Vargas, J. M. V. García, J. A. C. Carrillo, and J. J. C. Aguilar, "A biological-like controller using improved spiking neural networks," *Neurocomputing*, vol. 463, pp. 237–250, Nov. 2021, doi: 10.1016/j.neucom.2021.08.005.

[14] Z. Bing, C. Meschede, G. Chen, A. Knoll, and K. Huang, "Indirect and direct training of spiking neural networks for end-to-end control of a lane-keeping vehicle," *Neural Networks*, vol. 121, pp. 21–36, Jan. 2020, doi: 10.1016/j.neunet.2019.05.019.

[15] A. Linares-Barranco, F. Perez-Peña, A. Jimenez-Fernandez, and E. Chicca, "ED-BioRob: A neuromorphic robotic arm with FPGA-based infrastructure for bio-inspired spiking motor controllers," *Frontiers in Neurorobotics*, vol. 14, p. 590163, Nov. 2020, doi: 10.3389/fnbot.2020.590163.

[16] T. DeWolf, P. Jaworski, and C. Eliasmith, "Nengo and low-power AI hardware for robust, embedded neurorobotics," *Frontiers in Neurorobotics*, vol. 14, p. 568359, Oct. 2020, doi: 10.3389/fnbot.2020.568359.

[17] T. DeWolf, K. Patel, P. Jaworski, R. Leontie, J. Hays, and C. Eliasmith, "Neuromorphic control of a simulated 7-DOF arm using Loihi," *Neuromorphic Computing and Engineering*, vol. 3, no. 1, p. 014007, Feb. 2023, doi: 10.1088/2634-4386/acb286.

[18] C. Bartolozzi, G. Indiveri, and E. Donati, "Embodied neuromorphic intelligence," *Nature Communications*, vol. 13, no. 1, p. 1024, Feb. 2022, doi: 10.1038/s41467-022-28487-2.

[19] L. G. Troconis, F. Vella, Q. Chen, A. Freddi, O. Genova, and A. Monteriù, "Spiking neural network control for mobile robot path following," in *2026 34th mediterranean conference on control and automation (MED)*, Jun. 2026, pp. 601–606. doi: 10.1109/MED70602.2026.11598557.

[20] K. Jaisankar, X. Jiang, F. Liao, and J. S. Amuthan, "Population-coded spiking neural networks for high-dimensional robotic control." 2025. Available: https://arxiv.org/abs/2510.10516

[21] T. DeWolf, T. C. Stewart, J.-J. Slotine, and C. Eliasmith, "A spiking neural model of adaptive arm control," *Proceedings of the Royal Society B: Biological Sciences*, vol. 283, no. 1843, p. 20162134, Nov. 2016, doi: 10.1098/rspb.2016.2134.

[22] J. P. Fernández, M. A. Vargas, J. A. C. Carrillo, J. J. C. Aguilar, and B. Shyrokau, "Path-following control using spiking neural networks associative maps," *Robotics and Autonomous Systems*, vol. 193, p. 105077, 2025, doi: 10.1016/j.robot.2025.105077.

[23] C. Peng, G. Qiao, and B. Ge, "Intelligent compound control based on spiking neural network for hex-rotor unmanned aerial vehicle," in *2024 international annual conference on complex systems and intelligent science (CSIS-IAC)*, Sep. 2024, pp. 126–131. doi: 10.1109/CSIS-IAC63491.2024.10919458.

[24] S. Chen, S. Cai, T. Chen, C. Xu, and J. Chu, "Neural observer with lyapunov stability guarantee for uncertain nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 11527–11541, Aug. 2024, doi: 10.1109/TNNLS.2023.3262820.

[25] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *2012 IEEE 51st conference on decision and control (CDC)*, Dec. 2012, pp. 3270–3285. doi: 10.1109/CDC.2012.6425820.

[26] Q. Zhao, J. Si, and J. Sun, "Online reinforcement learning control by direct heuristic dynamic programming: From time-driven to event-driven," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 4139–4144, Aug. 2022, doi: 10.1109/TNNLS.2021.3053037.

[27] F. S. Slijkhuis, S. W. Keemink, and P. Lanillos, "Closed-form control with spike coding networks," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 16, no. 5, pp. 1677–1687, Oct. 2024, doi: 10.1109/TCDS.2023.3320251.

[28] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1749–1767, Jun. 2023, doi: 10.1109/TRO.2022.3232542.

[29] L. Brunke *et al.*, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022, doi: 10.1146/annurev-control-042920-020211.

[30] J. Wu, A. Clark, Y. Kantaros, and Y. Vorobeychik, "Neural lyapunov control for discrete-time systems," in *Advances in neural information processing systems*, Curran Associates, Inc., 2023, pp. 2939–2955.

[31] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, "Lyapunov-stable neural-network control," in *Robotics: Science and systems XVII*, 2021. doi: 10.15607/RSS.2021.XVII.063.

[32] A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo, "Formal synthesis of lyapunov neural networks," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 773–778, 2021, doi: 10.1109/LCSYS.2020.3005328.

[33] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, Nov. 2019, doi: 10.1109/MSP.2019.2931595.