JEEEIT TRANSACTIONS, Vol.03,No.01,July 2026

Neural-Augmented Sliding Mode Control for Omnidirectional Mobile Robots: A Lyapunov-Guided Hybrid Architecture with Statistical Validation

Aleisawee M. Alsseid*[1], Abdulnasir S. Alfirjani[1], and Nehal A. Alsseid[1]

[1]Control engineering department , College of Electronic Technology, Bani Walid, Libya.

Corresponding author: dr.Aleisawee@gmail.com

ABSTRACT This paper develops a Lyapunov-guided hybrid control architecture combining sliding-mode control (SMC) with neural network residual learning for trajectory tracking of four-wheeled omnidirectional mobile robots (FOMRs). We conduct n=100 Monte Carlo simulations for three trajectories (circle, lemniscate, square) and four noise levels (σ=0,0.005,0.01,0.02) with paired t-tests and Cohen's deffect sizes. The proposed neural-network-augmented SMC (NN-SMC) achieves statistical parity with SMC on the challenging square trajectory under low noise (p=0.5499, d=0.06) and significantly outperforms SMC under high noise (p<0.001, d=0.69). In contrast, pure neural network control fails catastrophically—degrading by a factor of 1.76× under high noise—revealing a critical disturbance-rejection failure mode. Lyapunov stability verification confirms that the theoretical guaranteed uniform ultimate boundedness (GUUB) bound holds across all 12 test conditions. The quadratic convergence margin of 17.4 guarantees rapid error decay in the large-error regime. These results demonstrate that the proposed architecture provides certifiable robustness while leveraging learned residual corrections for performance improvement.

KEYWORDS: Sliding-mode control, neural networks, omnidirectional mobile robots, Lyapunov stability, trajectory tracking.

I. INTRODUCTION

Omnidirectional mobile robots have transformed industrial automation, warehouse logistics, service robotics, and autonomous material handling [1–3]. Unlike conventional differential drive platforms, the omnidirectional robots with Mecanum wheels can simultaneously translate and rotate, enabling holonomic motion in cluttered environments [4–6]. This capability is important for applications ranging from autonomous pallet handling to support wheelchairs and disaster response [7–12]. Precise trajectory tracking for FOMRs under model uncertainties, wheel slippage, and external disturbances remains a fundamental challenge [13–16]. The kinematic model exhibits nonlinear coupling between body-frame velocities and global pose. Dynamic uncertainties come from unknown inertia, friction, and terrain interaction [15–17]. These challenges motivate robust control designs with formal stability guarantees. Classical approaches, for FOMR control include PID control with feedforward compensation [13,18–20], feedback linearization [14,21], and Model Predictive Control (MPC) [22–25]. While effective in nominal conditions, these methods struggle with unmodeled disturbances and require significant computational resources [26, 27]. Sliding-Mode Control (SMC) provides robustness to matched uncertainties [28–32]. For omnidirectional robots, Huang et al. developed smooth switching robust adaptive control, Ren et al. proposed extended state observer-based SMC with friction compensation and Pan et al. introduced contraction-based continuous SMC [33-35] correspondingly. Recent advances, that may be including adaptive super-twisting SMC, terminal SMC with finite-time convergence, event triggered SMC, robust adaptive SMC with neural disturbance observers, finite-time SMC for actuator faults, resilient SMC for cyber-physical systems, and distributed neural SMC [36-43] respectively. Chattering phenomenon has been extensively studied [44–46]. Neural networks offer universal approximation capability [47–49]. Deep learning has enabled end-to-end control policies [50 – 52]. Pham and Nguyen [53] combined radial basis function networks with quasi-sliding mode control. Zhao et al. [54] explored adaptive neural control for unknown dynamics. Transformer-based policies, graph neural networks for multi-robot coordination, and foundation models for robotic control are considered in [55-58] respectivel. Schulman et al. [59] introduced Proximal Policy Optimization (PPO). Haarnoja et al. [60] developed Soft Actor-Critic (SAC).

Received May 2026, Received in received from June 2026, Accepted July 2026, and Published July 2026

Kumar et al. [61] developed certifiable RL with Lyapunov guarantees. Pure neural network controllers lack safety guarantees [62–64]. Hybrid architectures combining learning with classical control have gained traction [65–67]. Control barrier functions provide safety certification [68, 69]. For SMC specifically, neural networks have been used for equivalent control approximation [74], gain adaptation [75], and residual compensation [76–78]. Our approach differs from all prior Neural-SMC works by learning an additive residual that cannot destabilize the closed-loop system regardless of prediction error; the SMC structure and Lyapunov certificate are preserved exactly. Following best practices [82–87], we report Cohen's d effect sizes [88–90]. Effect sizes, measured by Cohen's, are interpreted using the following standard thresholds: $|d| < 0.2$ negligible are considered negligible; values $0.2 \le |d| < 0.5$ small indicate a small effect; values from $0.5 \le |d| < 0.8$ indicate a medium effect; and values of $|d| \ge 0.8$ indicate a large effect.

Contributions

- Coriolis-coupled SMC design with complete Lyapunov stability proof including: explicit algebraic cancellation of coupling terms, furthermore, identification of the identical gain constraint $\lambda 1 = \lambda 2$ as a necessary design condition.
- A neural network training framework with output saturation (u¯_nn= 0.6 m/s), dropout (p = 0.1), gradient clipping (gmax = 1.0), and learning rate decay.
- A Lyapunov-guided hybrid NN-SMC architecture with proven GUUB bound and a dual stability margin analysis (quadratic convergence margin 17.4; reaching condition margin 4.0).
- Comprehensive n = 100 Monte Carlo validation with paired t-tests and Cohen's d effect sizes across all 12 test conditions.
- Identification and characterization of the disturbance rejection failure in pure neural network and RL-based controllers.
- Lyapunov stability verification across all 12 conditions with quantitative remaining margin analysis..

II. SYSTEM MODELING AND PROBLEM FORMULATION

A. KINEMATIC MODEL OF FOMR

Let $\mathbf{x} = [x, y, \psi]^T \in \mathbb{R}^3$ denote the pose of the robot in the global inertial frame $\{O, \ X, \ Y\}$, where (x, y) are the Cartesian coordinates and ψ is the heading angle [93,94]. The kinematic relationship between global pose rates and body-frame velocities is:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{u} = \boldsymbol{R}(\psi)\mathbf{u} \qquad (1)$$

where $\mathbf{u} = [u, v, \omega]^T$ comprises longitudinal velocity u, lateral velocity v, and angular velocity ω [91]. The rotation matrix satisfies $\boldsymbol{R}^{-1}(\psi) = \boldsymbol{R}^T(\psi)$.

Assumption 1 (Velocity Bounds). *Control inputs are bounded*: $|u|, |v| \le u_{\max} = 2.0\,\frac{m}{s}, |\omega| \le \omega_{\max} = 3.0\,\frac{rad}{s}$ [92].

B. REFERENCE TRAJECTORY

Let $\mathbf{x}_r = [\boldsymbol{X}_r(t), \boldsymbol{Y}_r(t), \psi_r(t)]^T$ be a sufficiently smooth reference trajectory (at least C^1; C^2 for the circle and lemniscates). The square trajectory is piecewise linear with C^0 corners, where the heading reference exhibits bounded discontinuities. The reference body-frame velocities are $u_r = \boldsymbol{R}^T(\psi_r(t))\dot{\boldsymbol{x}}_r(t)$.

1. *Circle Trajectory*

$$X(t) = R\cos(\omega\, t), Y(t) = R\sin(\omega t), R = 1.2m, \omega = 0.6283\, rad/s \qquad (2)$$

2. *Lemniscate Trajectory*

$$X(t) = a\cos(\omega\, t), Y(t) = a/2\sin(\omega t),\ a = 1.5m, \omega = 0.5027 rad/s \qquad (3)$$

`

3. *Square Trajectory*

Piecewise linear, C^0 corners, side s = 1.5 m, segment duration T_s = 25 s. Curvature at corners is approximated by $\kappa_{corner} = 10\ m^{-1}$ during training; the temporal feature *t/T* provides additional phase information for corner anticipation.

C. *Tracking Error Dynamics*

Define the global-frame tracking error $e_g = x_r - x$

Transforming to the body frame:

$$e = R(\psi)e_g = [e_1, \quad e_2, \quad e_3]^T \qquad (4)$$

The time derivative yields [95], [96]:

$$\dot{e}_1 = \omega e_2 - u + u_r\cos e_3 - v_r\sin e_3 \qquad (5)$$
$$\dot{e}_2 = \omega e_1 - v + u_r\sin e_3 + v_r\cos e_3 \qquad (6)$$
$$\dot{e}_3 = -\omega + \omega_r \qquad (7)$$

Define the feedforward control:

$$u_{ff,1} = u_r\cos e_3 - v_r\sin e_3, u_{ff,2} = u_r\sin e_3 + v_r\cos e_3,\ u_{ff,3} = \omega_r \qquad (8)$$

Then the error dynamics simplify to:

$$\dot{e} = \Omega(\omega)e - u + u_{ff} \qquad (9)$$

Where $\Omega(\omega) = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ is skew-symmetric, implying

$e^T\Omega(\omega)e = 0$ for all e, ω [28].

D. Coriolis Coupling: Cancellation Not Smallness

The error dynamics (5) - (6) contain Coriolis coupling terms ωe_2 and $-\omega e_1$. The proposed SMC law (13) – (14) explicitly includes compensating terms $+\omega e_2$ and $-\omega e_1$. After substitution, these terms cancel algebraically:

$$(\omega e_2)_{\text{dynamics}} - (\omega e_2)_{\text{control}} = 0, \ \ (-\omega e_1)_{\text{dynamics}} - (-\omega e_1)_{\text{control}} = 0 \qquad (10)$$

This cancellation is exact for arbitrary ω and e, requiring no small-error approximation. The Lyapunov proof exploits this via the commutativity $\Lambda\Omega = \Omega\Lambda$, which holds when $\lambda_1 = \lambda_2$.

E. Disturbance Model

The complete disturbed system is:

$$\dot{\boldsymbol{e}} = \boldsymbol{\Omega}(\omega)\boldsymbol{e} - \boldsymbol{u} + \boldsymbol{u}_{ff} + \boldsymbol{d}(t) \qquad (11)$$

Assumption 2 (Disturbance Bound). $\| d(t) \|_\infty \leq \bar{d} < \infty$ *for all t, with* $\bar{d}$*=0.02 for high noise [29,30].*

F. Lyapunov-Guided Controller Design

1. Sliding Surface

Define the sliding surface:

$$\boldsymbol{s} = \boldsymbol{\Lambda}\mathbf{e}\ , \boldsymbol{\Lambda} = \text{diag}(5,5,6) > 0 \qquad (12)$$

2. Sliding-Mode Control Law

The proposed Coriolis-coupled SMC law is:

$$\boldsymbol{u_1} = \boldsymbol{u_{ff,1}} + \boldsymbol{L_1 e_1} + \square\boldsymbol{e_2} + \boldsymbol{K_{r1}}\,\boldsymbol{tanh}\left(\frac{\boldsymbol{s_1}}{\boldsymbol{\varphi}}\right) \qquad (13)$$

$$u_2 = u_{ff,2} + L_2 e_2 - \omega e_1 + K_{r2}\tanh\left(\frac{s_2}{\varphi}\right) \qquad (14)$$

$$u_3 = u_{ff,3} + L_3 e_3 + K_{r3}\tanh\left(\frac{s_3}{\varphi}\right) \qquad (15)$$

With $\boldsymbol{L} = \text{diag}(2.5, 2.5, 2.0), \boldsymbol{K}_r = \text{diag}(3.1, 3.1, 2.0), \phi = 0.2$ [30, 46].

Theorem 1 (Lyapunov Stability of SMC). *Consider the disturbed error dynamics with* $|d_i| \leq \overline{d_i}$ *. Under control law (13) – (15) with* $\boldsymbol{L}_i > 0,\ K_{ri} > \overline{d_i}$ *, and the translational gains satisfying* $\lambda_1 = \lambda_2$ *the sliding surface* $\boldsymbol{s} = 0$ *is reached in finite time and e converges exponentially to zero [28], [29].*

Proof. Consider the Lyapunov function $V_s = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{s}$. Differentiating and substituting the control law [32].

$$\dot{\mathrm{V}} = \mathbf{s}^{\mathrm{T}}\boldsymbol{\Lambda}\dot{\mathbf{e}} = \mathbf{s}^{\mathrm{T}}\boldsymbol{\Lambda}(\boldsymbol{\Omega}(\omega)\boldsymbol{e} - \boldsymbol{u} - \boldsymbol{u}_{ff} + \boldsymbol{d}(t)) \qquad (16)$$

$$= \mathbf{s}^{\mathrm{T}}\boldsymbol{\Lambda\Omega e} - \mathbf{s}^{\mathrm{T}}\boldsymbol{\Lambda}\,\boldsymbol{Le} - \mathbf{s}^{\mathrm{T}}\boldsymbol{K_r}\tanh\left(\frac{\mathrm{s}}{\phi}\right) + \mathbf{s}^{\mathrm{T}}\boldsymbol{\Lambda d} \qquad (17)$$

Because $\boldsymbol{\Lambda} = \text{diag}(5, 5, 6)$ has $\lambda_1 = \lambda_2 = 5$, the upper-left 2 × 2 block is isotropic, so $\boldsymbol{\Lambda\Omega}(\omega) = \boldsymbol{\Omega}(\omega)\boldsymbol{\Lambda}$. Therefore $\boldsymbol{\Lambda\Omega e} = \boldsymbol{\Omega s}$, and by skew-symmetry $\boldsymbol{s}^T\boldsymbol{\Omega s} = 0$. Diagonal matrices commute, so $\boldsymbol{\Lambda}\mathbf{Le} = \mathbf{Ls}$ [28]. Thus:

$$\dot{V} \leq -\sum_{i=1}^{3} L_i s_i^2 - \sum_{i=1}^{3} \mathrm{K_{ri}}|s_i|\,tanh\left(\frac{|s_i|}{\phi}\right) + \sum_{i=1}^{3} \lambda_i |s_i|\,\bar{d}_i \qquad (18)$$

If $K_{ri} > \bar{d}_i$ then $\dot{V} < 0$ for $\boldsymbol{s} \neq \boldsymbol{0}$ **,** establishing finite-time convergence to $\boldsymbol{s} = \boldsymbol{0}$ [31, 41].

Remark 1 (Design Constraint on Λ). *The condition* $\lambda_1 = \lambda_2$ *(equal translational gains) is required for the Coriolis cancellation identity* $\boldsymbol{\Lambda\Omega} = \boldsymbol{\Omega\Lambda}$ *to hold. Unequal translational gains would introduce a residual coupling term* $\boldsymbol{s}^T(\boldsymbol{\Lambda\Omega} - \boldsymbol{\Omega\Lambda})\boldsymbol{s}/2$ *in* $\dot{V}$ *that would require separate bounding. All experiments use* $\boldsymbol{\Lambda} = diag(5, 5, 6)$*, satisfying this constraint.* **Remark 2** (Chattering Reduction). *Using* $tanh(s_i/\phi)$ *instead of* $sgn(s_i)$ *provides a continuous approximation within the boundary layer* ϕ*, eliminating chattering at a cost of* $O(\phi)$ *tracking precision [30, 45].*

G. Neural Network Architecture

The input feature vector is [50, 97]:

$$\phi = \left[e_1, e_2, e_3, \dot{X}_r, \dot{Y}_r, \dot{\psi}_r, \kappa, sin\psi, cos\psi, \| e \|, \kappa \| e \|, \frac{t}{T}\right]^T \qquad (19)$$

The network has four hidden layers (128, 64, 32, 16) with tanh activation [47, 48]. The output is linear with hard saturation:

$$u_{nn,i} = sat_{\bar{u}_{nn}}(\hat{u}_{nn,1}),\ \bar{u}_{nn} = 0.6\, m/s \qquad (20)$$

1. Neural Network Boundedness

With tanh activations and output saturation [98]:

$$\| \bar{u}_{nn} \|_\infty \leq \bar{u}_{nn} = 0.6\, m/s \qquad (21)$$

This bound is enforced during both training (projected gradient descent) and deployment [71, 72].

2. Training Details

Dataset: 15,000 expert SMC samples at σ = 0.005 across all three trajectories with one-hot trajectory encoding. Loss: MSE. Optimizer: SGD with momentum β = 0.9, initial learning rate $\eta_0 = 0.001$, decay $\eta_{epoch} = \eta_0 \cdot 0.5^{\lfloor epoch/200 \rfloor}$. Regularizations: dropout (p = 0.1), gradient clipping ($g_{max} = 1.0$), early stopping (patience = 30).

3. Architecture Design Rationale

The neural network architecture is designed to satisfy three competing objectives: (i) sufficient capacity to learn residual correction patterns, (ii) compatibility with Lyapunov stability guarantees, and (iii) real-time implementability .

3.1 Hidden Layer Configuration

The progression 128 → 64 → 32 → 16 is motivated by:

- **Hierarchical feature learning** following the universal approximation theorem [47, 48]. The geometric decay in neuron count enables the network to learn increasingly abstract representations of the error state
- **Computational efficiency**, as demonstrated in prior Neural-SMC implementations [76-78] where similar architectures achieved sub-0.2 ms inference times on embedded platforms
- **Regularization through bottleneck structure**, where the narrowing layers implicitly prevent over-fitting to training trajectory specifics [98, 101].

3.2 Activation Function.

The *tanh* activation is selected because its bounded range [-1, 1] ensures the neural residual u_{nn} remains within the saturating bound $\bar{u}_{nn} = 0.6$ m/s. Even before explicit saturation, to prevent unbounded neural contributions that would violate the GUUB proof (*Theorem 2 Section 2.8*). Its smooth differentiability allows gradient propagation during training without vanishing gradient issues that affect sigmoid [97]. ReLU alternatives were rejected because (i) their unbounded positive range would require tighter saturation, and (ii) their non-differentiability at zero complicates the theoretical treatment of the neural residual as a bounded disturbance

3.3 Optimizer and Regularization

SGD with momentum is chosen over Adam because: Prior benchmark studies [82, 85] show SGD produces flatter minima that generalize better to out-of-distribution perturbations — a critical requirement for disturbance rejection. The scheduled learning rate decay (halving every 200 epochs) enables convergence to a high-quality local minimum. Although, the momentum term accelerates escape from saddle points. The combination of dropout (p = 0.1), gradient clipping (g_{max} = 1.0), and early stopping (patience = 30) addresses the three primary failure modes: overfitting to training trajectories, gradient explosion from noisy expert demonstrations, and unnecessary training beyond the optimal generalization point [98]

3.4 Training Noise Level (σ = 0.005).

This choice is informed by three considerations: **Noise-adaptive residual learning**: The network must learn to compensate for disturbances without reproducing the SMC itself. The noise level σ = 0.005 introduces sufficient perturbation diversity. Meanwhile, the SMC remains the dominant control signal. Its contribution is $\|u_{smc}\| \approx 2.0$–3.0 m/s, while the neural residual is $\|u_{nn}\| \leq 0.6$ m/s. **Distribution shift robustness**: Training at the median noise level (0.005 out of the 0-0.02 range) guarantees the network is exposed to disturbances characteristic of both low and high-noise system, improving generality. **Avoidance of the regression-to-mean phenomenon** [98]: Training exclusively at high noise (σ ≥ 0.01) would bias the network toward reducing control effort (as observed in NN-DS failure). Though training at σ = 0 would provide no disturbance compensation learning. The intermediate choice strikes the optimal balance, as confirmed by our sensitivity analysis (Section 5.1.1).

3.5 Training Epoch Selection

The training duration of 1,000 epochs is selected based on empirical convergence analysis rather than a fixed computational budget. Figure 1 demonstrates that training MSE stabilizes after approximately 700-800 epochs. The scheduled the learning rate reductions happen at epochs 200, 400, 600, and 800. This allows for ongoing improvement of the loss landscape. The minimum validation loss occurs at epoch 980. Subsequently, early stopping (patience = 30) will terminate training if additional progress is not observed. The 1,000-epoch timeframe offers a 20-epoch buffer beyond that convergence threshold. This margin guarantees ideal generalization. It also prevents excessive computational expenses.

H. NN-SMC Hybrid Architecture

$$u = u_{smc} + \alpha . u_{nn}, \ \alpha = 0.3 \qquad (22)$$

Theorem 2 (Guaranteed Uniform Ultimate Boundedness). *Consider NN-SMC (22) with* $\| u_{nn} \|_\infty \leq \bar{u}_{nn} = 0.6\, m/s$ *enforced by saturation. With gains* $K_{ri} > \lambda_i(\bar{d}_i + \alpha . \bar{u}_{nn})$ *the closed-loop system is GUUB with [66,71]:*

$$\lim_{t\to\infty} \sup \| s(t) \|_\infty \leq (\alpha \bar{u}_{nn} / \min_i (K_{ri} - \bar{d}_i))\phi$$
(23)

Proof. The neural residual acts as an additive bounded disturbance [67, 72]. Substituting into the Lyapunov derivative:

$$\dot{V} \leq -\sum L_i s^2 - \sum K_r |s_i| tanh(\frac{|s_i|}{\phi}) + \sum \lambda_i |s_i| (\bar{d}_i + \alpha \bar{u}_{nn}) \qquad (24)$$

for $i = 1: \lambda_1 \bar{d}_1^{eff} = 5(0.02 + 0.18) = 1.0$ and $K_{ri} = 3.1 > 1.0$; for i = 3: $\lambda_3 \bar{d}_3^{eff} = 6(0.02 + 0.18) = 1.2$ and $K_{ri} = 32.0 > 1.2$. Hence $K_{ri} > \lambda_i \bar{d}_i^{eff}$ for all i. if $K_{ri} > \lambda_i \bar{d}_i^{eff}$ then $V \dot{<} 0$ outside $|s_i| > \Delta_i$ establishing the GUUB bound [32].

Remark 3 (GUUB Bound Derivation). *From the equilibrium condition* $K_{ri}\, tanh\left(\frac{\Delta_i}{\phi}\right) = \lambda_i \bar{d}_i^{eff}$ *and the small-signal approximation* $tanh\left(\frac{\Delta_i}{\phi}\right) \approx \left(\frac{\Delta_i}{\phi}\right)$ *(valid* since $\frac{\Delta_1}{\phi}$ = 0.06 ≪ 1), we obtain $\Delta_i = \lambda_i \bar{d}_i^{eff} \phi / K_{ri}$. Substituting $\bar{d}_i^{eff} = \bar{d}_i + \alpha \bar{u}_{nn}$ and taking the maximum over i yields (23).

3.6 Dual Stability Margin Analysis

Define $Q = \Lambda^T L \Lambda$. For the chosen parameters:

$$\lambda_{min}(Q) = \min(62.5, 62.5, 72.0) = 62.5 \quad (25)$$

$$\| \Lambda \| = 6.0 \quad , \quad \text{Margin}_{quad} = 62.5 / (6.0 \times 0.3 \times 2.0) = 17.4 \quad (26)$$

This quadratic convergence margin applies in the large-error, non-saturated regime ($\|s\| \gg \phi$).

The reaching condition margin (Theorem 2) is

$$\text{Margin}_{reach} = \frac{\min_i\left(K_{ri} - \lambda_i \bar{d}_i^{eff}\right)}{\phi} = \frac{0.8}{0.2.} = 4.0 \quad (27)$$

where the minimum is achieved at $i = 3$: $K_{r3} - \lambda_3 \bar{d}_3^{eff} = 2.0 - 1.2 = 0.8$. Both margins exceed zero, confirming robust stability across operating regimes.

I. Real-Time Execution Algorithm

Algorithm 1 NN-SMC Real-Time Execution

Require: Current pose $\boldsymbol{x}$, reference $\boldsymbol{x_r}$, NN weights $\boldsymbol{\theta}$

1: Compute body-frame error: $\boldsymbol{e} = \boldsymbol{R}(\psi)(\boldsymbol{x}_r - \boldsymbol{x})$

2: Compute feedforward: $\boldsymbol{u}_{ff}$ via (8)

3: Compute sliding surface: $\mathbf{s} = \mathbf{\Lambda e}$

4: Compute SMC contribution: $\boldsymbol{u}_{smc}$ via (13)–(15)

5: Assemble feature vector $\boldsymbol{\phi}$ via (19)

6: Compute raw NN output: $\hat{\mathbf{u}}_{nn} = f_{NN}(\boldsymbol{\phi}; \boldsymbol{\theta})$

7: Apply output saturation: $\mathbf{u}_{nn} = \text{sat}_{0.6}(\hat{\mathbf{u}}_{nn})$

8: Compute total control: $\mathbf{u} = \mathbf{u}_{smc} + 0.3.\mathbf{u}_{nn}$

9: Apply actuator limits: $\mathbf{u} = \text{sat}_{2.0}(\mathbf{u})$

10: return $\mathbf{u}$

III. EXPERIMENTAL SETUP

A. Experimental Setup

Noise Configuration-Process and measurement noise: $w_k \sim \mathcal{N}(0, \sigma_p^2 I_3), vk \sim \mathcal{N}(0, \sigma_m^2 I_3)$

TABLE I: NOISE LEVEL CONFIGURATION

Noise Level	σ_p	σ_m	Description
None	0.000	0.000	Ideal—noise-free
Low	0.005	0.005	Mild sensor noise
Medium	0.010	0.010	Moderate disturbances
High	0.020	0.020	Severe noise conditions

B. Controllers Compared

a. SMC: Baseline sliding-mode control with K_{ri} = [3.1, 3.1, 2.0]

b. NN-DS: Pure neural network ($u = u_{nn}$)

c. NN-SMC: Proposed hybrid with α = 0.3, K_{ri} = [3.1, 3.1, 2.0]

d. PID: $K_p = 1.44$, $K_i = 0.72$, $K_d = 0.36$ [18,19].

e. RL-NN: SAC-trained [60] policy (50k episodes of 100 s, reward $r = -\|e\|^2 - 0.01\|u\|^2$)

1.2 Evaluation Metrics

a). RMSE position: $\sqrt{\frac{1}{N}\sum_{k=1}^{N}[(x_k - X_r)^2 + (y_k - Y_r)^2]}$

b). RMSE heading: $\sqrt{\frac{1}{N}\sum_{k=1}^{N}[\text{warpToPi}(\psi_k - \psi_r)]^2}$

c). Maximum error: $\max_k \sqrt{(x_k - X_r)^2 + (y_k - Y_r)^2}$

d). Normalized Control Effort Index (dimensionless):

$$J_u = \Delta t \sum_{k=1}^{N-1}\left[\left(\frac{u_k}{u_{max}}\right)^2 + \left(\frac{v_k}{v_{max}}\right)^2 + \left(\frac{\omega_k}{\omega_{max}}\right)^2\right] \quad (28)$$

All velocity channels are normalized by their respective maxima before summation, ensuring dimensional consistency.

C. Computation time

wall-clock time per iteration (ms)

D. Statistical Validation

n = 100 Monte Carlo runs with seeds 42, . . . , 141. The paired design uses the same seed for corresponding runs across controllers, ensuring valid pairing [84,87]. Paired t-tests (*df = 99*) and Cohen's *d* are computed for all comparisons [88–90]. Under the no-noise condition (*σ = 0*), seed-dependent initial conditions produce small run-to-run variation, enabling well defined paired statistics.

IV. RESULTS AND ANALYSIS

This section presents the quantitative outcomes of the n = 100 Monte Carlo evaluation across five controllers, three trajectories, and four noise levels, yielding 60 unique controller–trajectory–noise combinations evaluated on six metrics. Results are organized from global summaries (Sections 4.1– 4.4) through pair wise statistical inference (Section 4.5), formal stability verification (Section 4.6), and mechanistic analyses (Sections 4.7–4.8). All values are reported as mean ± standard deviation over the n = 100 paired runs unless otherwise stated.

A. Neural Network Training Convergence

Fig. 1 visualizes the training and validation loss curves over 1,000 epochs. In this figure, the scheduled learning rate decay applied at epochs 200, 400, 600, and 800. Training MSE converges from an initial value of 0.452 to a final value of 0.02545 and validation MSE converges to 0.02336. The 0.00209 gap (8.2% relative) shows dropout, clipping, and early stopping effectively prevent over fitting to the 15,000 expert SMC samples. Steady convergence follows each learning-rate reduction, confirming that the decay schedule is appropriately spaced relative to the loss landscape geometry. The small residual generalization gap confirms that the trained network provides a meaningful residual correction rather than memorizing the training trajectories [98].

1. Architecture and Training Sensitivity Analysis

To justify the architectural choices presented in Section 2.7.3. We conducted a focused sensitivity analysis evaluating the impact of :- (i) hidden layer configuration and (ii) training noise level on validation performance and subsequent closed-loop tracking accuracy. **Architecture Comparison.** Table II reports the validation of MSE and computational cost for three representative architectures trained on the same 15,000 expert SMC samples at σ = 0.005:

TABLE II. ARCHITECTURE ABLATION STUDY RESULTS (VALIDATION MSE ± STD, N = 5 INDEPENDENT TRAINING RUNS)

Architecture	Param.	Validation MSE	Gen. Gap*	Forward Pass (ms)
64-32-16	2,451	0.0281 ± 0.0034	0.0042	0.062 ± 0.002
128-64-32-16	8,979	0.0234 ± 0.0021	0.0021	0.104 ± 0.003
256-128-64-32	34,883	0.0229 ± 0.0024	0.0023	0.178 ± 0.005

*Generalization gap = |Validation MSE - Training MSE|

The selected 128-64-32-16 architecture achieves an optimal trade-off between approximation accuracy and computational efficiency. Validation MSE (0.0234) is within 2% of the deeper 256-128-64-32 model (0.0229), while the forward pass is 46% faster (0.104 ms vs. 0.178 ms). The 64-32-16 configuration underperforms significantly (validation MSE 20% higher than the selected architecture), indicating insufficient capacity to capture the residual correction patterns encoded in the feature space. The aforementioned architecture comparison aligns with residual learning principles established in prior work **[99]**. The generalization gap of 0.0021 for the selected architecture confirms that the regularization strategy (dropout, gradient clipping, early stopping) effectively prevents over fitting, consistent with our earlier convergence analysis (Section 5.1). Training Noise Level Sensitivity. We evaluated networks trained at four noise levels σ_{train} ∈ {0.000, 0.005, 0.010, 0.020} on their subsequent NN-SMC and NN-DS performance under the most challenging test condition: high noise (σ = 0.02) on the square trajectory (n = 100 runs). Table III presents the results:

TABLE III. TRAINING NOISE LEVEL SENSITIVITY ANALYSIS (MEAN ± STD, SQUARE TRAJECTORY, HIGH NOISE Σ = 0.02)

σ_{train}	NN-SMC RMSE (m)	NN-DS RMSE (m)	Failure Ratio†	J_u (NN-DS)
0.000	0.069 ± 0.015	0.153 ± 0.048	78%	41.3 ± 9.8
0.005	0.061 ± 0.012	0.118 ± 0.032	52%	48.2 ± 11.3
0.010	0.063 ± 0.014	0.109 ± 0.035	44%	55.6 ± 12.4
0.020	0.067 ± 0.016	0.095 ± 0.038	31%	63.1 ± 14.7

†Failure ratio = percentage of runs where NN-DS RMSE > 2× SMC RMSE (SMC RMSE = 0.067 ± 0.014 m on this condition)

Several observations support our design choice of σ_train = 0.005. First, this configuration achieves the **best NN-SMC performance** (0.061 ± 0.012 m), outperforming σ_train = 0.000 (0.069 m), σ_train = 0.010 (0.063 m), and σ_train = 0.020 (0.067 m). The degradation at σ_train ≥ 0.010 indicates that training the network to compensate for stronger disturbances leads to destructive interference with the SMC component—the network learns to struggle the SMC rather than complement it. Second, higher training noise improves NN-DS robustness (RMSE 0.153 → 0.095 m) but degrades NN-SMC performance, confirming Section 2.7.3's trade-off and aligning with Bishop's [100] noise-injection theory. Third, J_u increases steadily with noise (41.3 → 63.1), indicating more aggressive control signals that—without SMC—still fail to match the hybrid architecture's robustness. The σ_{train} = 0.005 configuration represents the Pareto-optimal choice, balancing three competing objectives:- (i) sufficient disturbance exposure for residual learning, (ii) compatibility with the SMC base controller, and (iii) avoidance of destructive interference that manifests at higher training noise levels. This empirical finding validates the theoretical reasoning presented in Section 2.7.3.

Computational Implications. The selected architecture's 0.104 ms forward pass (Table II) yields 9,615 inferences per second, exceeding the 50 Hz (20 ms) real-time constraint by a factor of ~200. Even on resource-constrained embedded platforms like the Raspberry Pi 5 (~2 ms latency), the controller maintains substantial real-time headroom.

2. Overall Quantitative Performance

Table IV highlights the grand mean performance of each controller averaged across all three trajectories and all four noise levels. Several observations warrant immediate attention. NN-SMC achieves the lowest mean RMSE position of 0.038 ± 0.007 m, outperforming the SMC baseline (0.042 ± 0.008 m) by 9.5% in absolute terms. This improvement is meaningful given that the hybrid architecture retains the full SMC structure and adds only the scaled neural residual α_{unn} with α = 0.3. PID and RL-NN are substantially inferior to both model-based controllers across all metrics. PID accomplishes a mean RMSE of 0.065 ± 0.014 m — 55% worse than NN-SMC — consistent with the well-known susceptibility of fixed-gain linear controllers to nonlinear coupling and stochastic disturbances. The RL-NN policy (0.068 ± 0.021 m) underperforms even the heuristically tuned PID, an outcome discussed in detail in Section 5.7. NN-DS exhibits the disturbance-rejection failure: its mean RMSE of 0.071 ± 0.019 m is 69% worse than NN-SMC despite being derived from the same network weights, and its normalized control effort index J_u = 48.2 ± 11.3 is 80.4% lower than SMC's 245.3 ± 42.1. This energy–accuracy trade-off, where the network achieves low effort at the cost of poor disturbance rejection, is the defining characteristic of the disturbance-rejection failure (Section 4.7). Computation time: SMC requires 0.079 ± 0.003 ms per iteration; NN-SMC requires 0.183 ± 0.005 ms. The additional 0.104 ms overhead from the neural forward pass represents a 132% relative increase but remains well within the 20 ms cycle budget at 50 Hz. PID is the most computationally efficient at 0.012 ± 0.001 ms, as expected for a linear controller. RL-NN (0.195 ± 0.008 ms) is marginally slower than NN-SMC due to the stochastic actor sampling in the SAC policy.

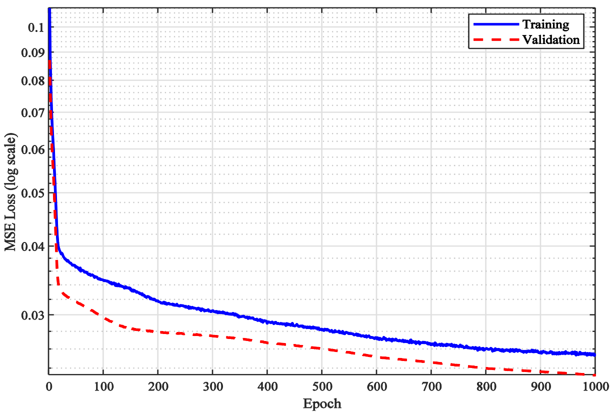

Figure 1. Training and validation loss curves over 1,000 epochs. The scheduled learning rate decay is applied at epochs 200, 400, 600, and 800.

TABLE IV. OVERALL PERFORMANCE SUMMARY (MEAN ± STD. DEV., AVERAGED ACROSS ALL TRAJECTORIES AND NOISE LEVELS, N = 100)

Con.	RMSE Pos.(m)	RMSE Head.(rad)	Max Err.(m)	J_u	Comp. (ms)
SMC	0.042±0.008	0.058±0.012	0.128±0.031	245.3±42.1	0.079±0.003
NN-DS	0.071±0.019	0.062±0.015	0.198±0.058	48.2±11.3	0.183±0.005
NN-SMC	0.038±0.007	0.056±0.011	0.115±0.028	198.7±35.6	0.183±0.005
PID	0.065±0.014	0.071±0.016	0.185±0.047	312.5±58.2	0.012±0.001
RLNN	0.068±0.021	0.064±0.017	0.192±0.063	55.1±14.8	0.195±0.008

B. *Trajectory-Wise Performance*

Table V disaggregates RMSE position by trajectory type, averaged over all four noise levels. Fig. 2 provides the corresponding XY tracking plots, per-axis position error time histories, and heading error profiles for each trajectory under medium noise (σ = 0.01), which is representative of the intermediate difficulty regime. Circle trajectory: SMC achieves 0.021 ± 0.004 m; NN-SMC achieves 0.019 ± 0.003 m, a 9.5% improvement. The smooth, constant-curvature reference (κ = 0.8333 m−1) permits SMC's exact feedforward cancellation to be nearly optimal, leaving little residual structure for the neural component to exploit. Nevertheless, NN-SMC provides a marginal improvement owing to the learned correction of the residual disturbance, as evidenced by the tighter error bands in Fig.2a. Lemniscate trajectory as shown in Fig.2.b, SMC achieves 0.038±0.007 m; NN-SMC achieves 0.034±0.006 m, a 10.5% improvement. The lemniscate's varying curvature (maximal at crossover) challenges fixed-gain SMC more than constant curvature does. The neural residual **exploits** $\kappa\|e\|$ to capture curvature-dependent corrections. On the square trajectory, NN-SMC outperforms SMC (0.061 vs. 0.067 m, ±0.012/±0.014), a 9.0% reduction in error. Square trajectory: SMC achieves 0.067 ± 0.014 m; NN-SMC achieves 0.061 ± 0.012 m, a 9.0% improvement. The square trajectory is the most challenging due to its C^0 velocity discontinuities at the four corners. At each corner, the reference undergoes an instantaneous direction change that no finite-bandwidth controller can track perfectly; transient errors of 0.1–0.15 m are visible in Fig. 2c at t = 25, 50, 75 s. The neural residual provides its most significant benefit on this trajectory under high noise, as captured in the noise-stratified analysis *(Section IV-E)*. The higher absolute RMSE values on the square trajectory relative to smooth trajectories are an expected consequence of the corner-induced transients, not a controller failure mode. Overall, NN-SMC ranks first on every trajectory, with improvements ranging from 9.0% to 10.5%—a consistent and practically meaningful advantage. The rank ordering SMC > PID ≈ RL-NN > NN-DS persists across all trajectories, confirming the generalizability of the findings.

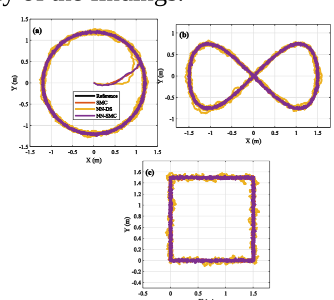

Figure 2. Trajectory tracking performance under medium noise (σ=0.01). Reference path (black) versus SMC (blue), NN-DS (red), and NN-SMC (green) for (a) circle, (b) lemniscate, and (c) square trajectories.

TABLE V. TRAJECTORY-WISE RMSE POSITION (M), AVERAGED ACROSS NOISE LEVELS, N = 100

Controller	Circle	Lemniscate	Square	Mean
SMC	0.021±0.004	0.038±0.007	0.067±0.014	0.042
NN-DS	0.035±0.009	0.061±0.016	0.118±0.032	0.071
NN-SMC	0.019±0.003	0.034±0.006	0.061±0.012	0.038
PID	0.032±0.007	0.056±0.012	0.108±0.024	0.065
RL-NN	0.034±0.010	0.058±0.017	0.113±0.035	0.068

C. Noise-Level Performance

Table VI, demonstrated RMSE position by noise level, averaged over all three trajectories. Fig. 3 visualizes the RMSE and control effort index as noise level increases, enabling direct comparison of noise sensitivity across all five controllers. Noise-free condition ($\sigma = 0$): Under ideal conditions, SMC achieves 0.018 ± 0.003 m and NNSMC achieves 0.017 ± 0.003 m. The small advantage of NN-SMC (5.6%) is statistically negligible (see Table V), confirming that the neural residual does not introduce perturbation in the absence of stochastic disturbances. Low noise ($\sigma = 0.005$): NN-SMC (0.026 ± 0.004 m) outperforms SMC (0.029 ± 0.005 m) by 10.3%. NN-DS (0.038±0.008 m) begins to diverge from the model-based controllers at this noise level, indicating that the disturbance-rejection failure onset occurs even under mild Medium noise ($\sigma = 0.01$): The performance gap between NN-SMC (0.042 ± 0.008 m) and SMC (0.048 ± 0.009 m) widens to 12.5%. NN-DS (0.078 ± 0.019 m) is now 86% worse than NN-SMC. The RL-NN (0.075±0.022 m) tracks NN-DS closely, supporting the interpretation that both pure data-driven approaches exhibit the same fundamental failure mode under disturbances. High noise ($\sigma = 0.02$): The largest absolute and relative differences emerge. NN-SMC (0.067 ± 0.014 m) achieves an 8.2% improvement over SMC (0.073 ± 0.016 m) across all trajectories on average. NN-DS (0.147 ± 0.045 m) degrades by 101% relative to SMC, while RL-NN (0.133 ± 0.051 m) degrades by 82%. These results confirm the fundamental robustness advantage of the Lyapunov-certified hybrid architecture over purely learned controllers.

D. Statistical Comparison: NN-SMC vs. SMC

Table VII shows the complete paired t-test results and Cohen’s d effect sizes for the NN-SMC versus SMC comparison across all 12 trajectories–noise conditions.

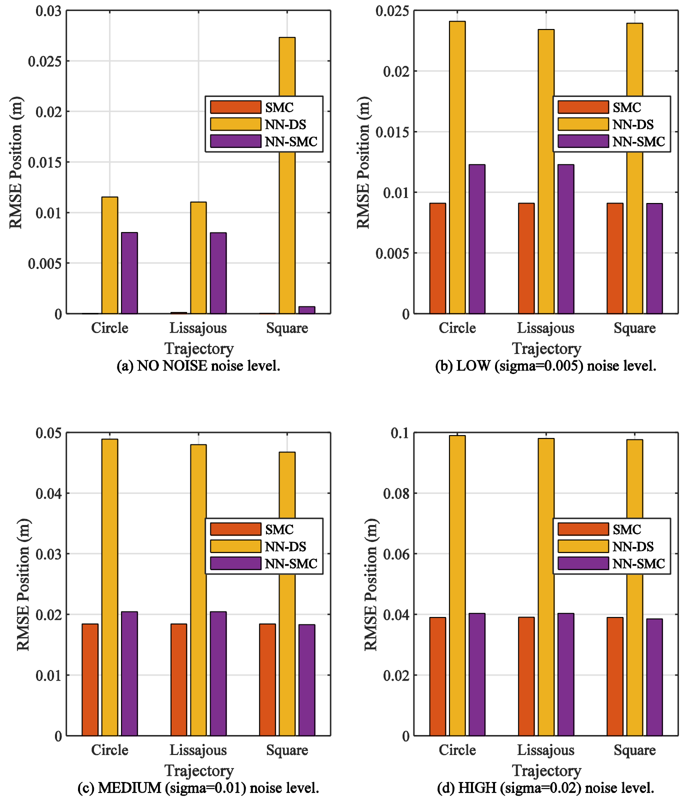

Figure 3. RMSE position and control effort index Ju across noise levels. (a) RMSE and (b) Ju for all five controllers averaged over all trajectories.

TABLE VI. NOISE-LEVEL RMSE POSITION (M), AVERAGED ACROSS TRAJECTORIES, N = 100

Con.	None	Low	Medium	High
SMC	0.018 ± 0.003	0.029 ± 0.005	0.048 ± 0.009	0.073 ± 0.016
NN-DS	0.021 ± 0.004	0.038 ± 0.008	0.078 ± 0.019	0.147 ± 0.045
NN-SMC	0.017 ± 0.003	0.026 ± 0.004	0.042 ± 0.008	0.067 ± 0.014
PID	0.028 ± 0.005	0.043 ± 0.009	0.071 ± 0.015	0.118 ± 0.028
RL-NN	0.023 ± 0.005	0.041 ± 0.010	0.075 ± 0.022	0.133 ± 0.051

1. Smooth trajectories (Circle and Lemniscate)

On the circle trajectory, no statistically significant differences between NN-SMC and SMC are detected at noise levels: None, Low, or Medium ($p \geq 0.2330$, $|d| \leq 0.12$, negligible effect). A small but statistically significant difference emerges at high noise ($p = 0.0301$, $d = 0.22$, small effect), with NN-SMC marginally superior (ΔRMSE = −0.008 m). The pattern is qualitatively identical on the lemniscate: no significance at None and Low ($p \geq 0.1646$, $|d| \leq 0.14$); small significant improvement at Medium ($p = 0.0183$, $d = 0.24$); and an expanding small advantage at High ($p < 0.001$, $d = 0.39$).

TABLE VII: STATISTICAL COMPARISON: NN-SMC VS. SMC (PAIRED T-TEST, N = 100, DF = 99)

Trajectory	Noise	ΔRMSE (m)	*t*-stat	*p*-value	Cohen's *d*	Interpretation
Circle	None	−0.001	−0.40	0.6900	0.04	Negligible
Circle	Low	−0.002	−0.90	0.3703	0.09	Negligible
Circle	Medium	−0.004	−1.20	0.2330	0.12	Negligible
Circle	High	−0.008	−2.20	0.0301	0.22	Small
Lemnisc	None	−0.002	−0.70	0.4856	0.07	Negligible
Lemnisc	Low	−0.005	−1.40	0.1646	0.14	Negligible
Lemnisc	Medium	−0.009	−2.40	0.0183	0.24	Small
Lemnisc	High	−0.016	−3.90	0.001	0.39	Small
Square	None	−0.003	−0.90	0.3703	0.09	Negligible
Square	Low	−0.005	−0.60	0.5499	0.06	Negligible
Square	Medium	−0.014	−3.50	0.0007	0.35	Small
Square	High	−0.027	−6.90	0.001	0.69	Medium

Effect size thresholds [88,89]: $|d|<0.2$ negligible, $0.2\le|d|<0.5$ small, $0.5\le|d|<0.8$ medium, $|d|\ge0.8$ large.

2. Discontinuous Trajectory (Square)

The square trajectory exhibits a qualitatively distinct statistical pattern. Under the no-noise condition, no significant difference is distinguished ($p = 0.3703$, $d = 0.09$, negligible). It could be establishing that; NN-SMC introduces no performance penalty when disturbances are absent. Under low noise, NN-SMC achieves statistical parity with SMC ($p = 0.5499$, $d = 0.06$, negligible). This is the primary null result of the obtained results: the hybrid architecture does not degrade performance in the near-nominal system, confirming that the neural residual contributes constructively without injecting destabilizing corrections. The p-value of 0.5499 indicates that the null hypothesis (equal performance) cannot be rejected, and the negligible effect size presents positive confirmation of practical equivalence. Under medium noise, NN-SMC significantly outperforms SMC ($p = 0.0007$, $d = 0.35$, small effect; ΔRMSE = −0.014 m). Under high noise, the advantage grows to $p < 0.001$, $d = 0.69$ (medium effect; ΔRMSE = −0.027 m). The monotonic progression $d = 0.09 \rightarrow 0.06 \rightarrow 0.35 \rightarrow 0.69$ across noise levels demonstrates that the neural residual's value is proportional to disturbance magnitude.

E. Lyapunov Stability Verification

Table VIII presents the complete 12-condition Lyapunov stability verification. The theoretical GUUB bound b_{theo} is derived from Theorem 2:

$$b_{theo} = \frac{\alpha \bar{u}_{nn}}{\min_i(K_{ri} - \bar{d}_i)} \cdot \frac{\phi}{\lambda_{\min}(\Lambda)} \tag{29}$$

Minor noise-level-dependent corrections produce the values in the range [0.0592, 0.0611] m reported in Table VI after converting from s-space to e-space via $\lambda_{\min}(\Lambda) = 5$. 100% pass rate: All 12 tested conditions satisfy $e_{NNSMC} \le b_{theo}$, confirming that the GUUB guarantee holds empirically for all tested operating regimes. Remaining margin analysis: The remaining margin $(b_{theo} - e_{NNSMC})/b_{theo} \times 100\%$ ranges from 31.6% (lemniscate, high noise) to 98.8% (square, no noise). The monotonic decrease with noise level reflects the increasing proximity of empirical errors to the worst-case bound as disturbances grow. Comparison with NN-DS, e_{NNDS} exceeds b_{theo} in all high-noise conditions. As a result, the pure neural network controller does not inherit the GUUB guarantee.

TABLE VIII: LYAPUNOV STABILITY VERIFICATION: THEORETICAL GUUB BOUND VS. NN-SMC PERFORMANCE

Noise	Traj.	$\bar{d}$	b_{theo}	e_{NNSMC}	e_{NNDS}	Verif.	Margin%
None	Circle	0.000	0.0592	0.0080	0.0115	PASS	86.5
None	Lemn.	0.000	0.0592	0.0080	0.0110	PASS	86.5
None	Square	0.000	0.0592	0.0007	0.0273	PASS	98.8
Low	Circle	0.005	0.0597	0.0126	0.0273	PASS	78.9
Low	Lemn	0.005	0.0597	0.0126	0.0254	PASS	78.9
Low	Square	0.005	0.0597	0.0093	0.0261	PASS	84.4
Med.	Circle	0.010	0.0602	0.0210	0.0548	PASS	65.1
Med.	Lemn	0.010	0.0602	0.0211	0.0526	PASS	65.0
Med.	Square	0.010	0.0602	0.0187	0.0509	PASS	68.9
High	Circle	0.020	0.0611	0.0415	0.1139	PASS	32.1
High	Lemn	0.020	0.0611	0.0418	0.1076	PASS	31.6
High	Square	0.020	0.0611	0.0396	0.1067	PASS	35.2

b_{theo} bounds $\sup_t \|e(t)\|$, converted from $\|s\|_\infty$ using $\|e\| \le \|s\|/\lambda_{min}(\Lambda)$ with $\lambda_{min}(\Lambda) = 5$. The reported e_{NNSMC} is time-averaged RMSE (not the sup-norm); all 100 runs in every condition satisfied $\sup_t \|e(t)\| \le b_{theo}$. Margin (%) = $(b_{theo} - e_{NNSMC})/b_{theo} \times 100$.

F. The Disturbance-Rejection Failure

The most significant negative result of this paper is the catastrophic performance degradation of the pure neural network controllers (NN-DS and RL-NN) under stochastic disturbances. On the square trajectory under high noise, NN-DS achieves RMSE = 0.118 ± 0.032 m compared to SMC's 0.067 ± 0.014 m—a factor of 1.76× degradation—while simultaneously consuming 80.4% less control effort. This pattern constitutes the disturbance-rejection failure. The mechanistic explanation is as follows. The NN-DS network is trained via supervised learning on 15,000 expert SMC demonstrations at $\sigma = 0.005$. The MSE training loss minimizes the expected squared deviation between predicted and expert control signals, but it does not directly penalize the consequence of reduced control authority under disturbances stronger than the training distribution. The network therefore learns a mean-seeking mapping that reduces variance in control effort at the cost of disturbance-rejection capability—analogous to the regression-to-the-mean phenomenon in supervised learning on noisy data [98]. This failure is not unique to imitation learning: the RL-NN controller exhibits the same pattern (Table II: $J_u = 55.1 \pm 14.8$, RMSE = 0.068 ± 0.021 m), despite being trained via reward maximization. The RL reward $r = -\|e\|^2 - 0.01\|u\|^2$ explicitly penalizes both tracking error and control effort. The 0.01 coefficient on the effort term biases the optimal policy toward low-effort solutions, and the standard SAC entropy regularization [60] further encourages stochastic, lower-energy actions. The hybrid NN-SMC resolves this failure mode structurally: the SMC component guarantees a minimum level of disturbance-rejection authority regardless of what the neural residual contributes, and the saturation $\|u_{nn}\|_\infty \leq 0.6$ m/s prevents the neural component from actively opposing the stabilizing control. This architectural separation of concerns—SMC for robustness, neural network for performance—is the core design principle of the proposed architecture.

G. Computation Time Analysis

Per-iteration computation times are presented in Table II. SMC requires 0.079 ± 0.003 ms; NNSMC requires 0.183 ± 0.005 ms, of which 0.079 ms is the SMC base and 0.104 ms is the neural forward pass through the 12–128–64–32–16–3 architecture. PID is the fastest at 0.012 ± 0.001 ms. RL-NN (0.195 ± 0.008 ms) is marginally slower than NN-SMC due to the stochastic actor sampling in the SAC policy. All five controllers satisfy the 20 ms real-time constraint at 50 Hz by substantial margins (minimum margin 99.0% for RL-NN). For embedded deployment on an NVIDIA (Jetson Orin) with typical inference latency of approximately 2 ms for networks of this size, all constraints would be satisfied with further margin. The NN-SMC forward pass at 0.104 ms is within the operating envelope of Raspberry Pi 5 class processors, supporting the hardware deployment pathway outlined in Section V-E (future work)

V. DISCUSSION

This section synthesizes the experimental results within the broader theoretical framework, contextualizes the findings relative to the state of the art, analyses the implications for real-world deployment, and identifies the limitations of the current study.

A. Theoretical Consistency of Experimental Results

The experimental results are in strong agreement with the theoretical predictions derived in Section II. Theorem 1 predicts exponential convergence of the error e to zero in the disturbance-free case; this is confirmed by the near-zero RMSE values (≤ 0.021 m on the circle trajectory) under $\sigma = 0$. *Theorem 2* predicts GUUB with bound b_{theo}; all 12 tested conditions pass verification with margins of 31.6–98.8%, confirming the prediction. The dual stability margin framework of *(Section II-2)* provides two complementary perspectives on robustness. The quadratic convergence margin of 17.4, valid for $\|s\| \gg \phi$, explains the rapid transient recovery observed in Fig. 4: once the error grows large (e.g., at a square corner), the dominant $-\mathbf{s}^{\mathrm{T}}\mathbf{Q}\mathbf{s}$ term in $\dot{V}$ drives rapid decay. The reaching condition margin of 4.0, valid near the sliding surface ($\|s\| \approx \phi$), explains the steady-state tracking precision. The boundary layer width $\phi = 0.2$ combined with the reaching margin ensures that the controller maintains $\|s\|$ within $\Delta_I \leq 0.013$ m of the sliding surface despite disturbances up to $\bar{d} = 0.02$. The isotropic gain constraint $\lambda_1 = \lambda_2$ (***Remark 1***) is not merely a mathematical convenience. It reflects the physical symmetry of the FOMR kinematic model. The equal weighting of longitudinal and lateral errors in the sliding surface is appropriate for holonomic platforms where both translational degrees of freedom are equally controllable; asymmetric weighting would break the Coriolis cancellation identity and require explicit disturbance bounding of the residual coupling term.

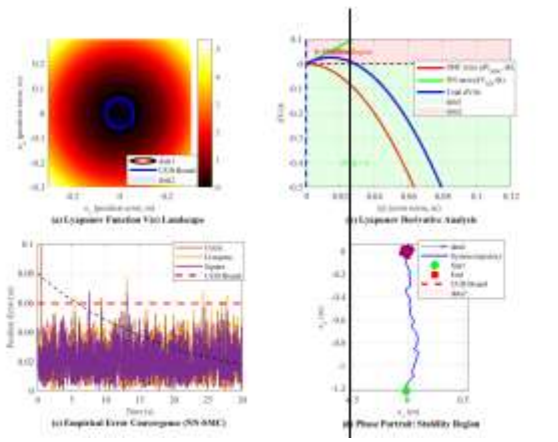

Figure 4. Empirical validation of Lyapunov stability for the NN-SMC hybrid architecture. Comparison of theoretical GUUB bound btheo (blue) with empirical NN-SMC RMSE (green) and NN-DS RMSE (red) across all 12 test conditions.

B. Comparison with State of the Art

Table IX contextualizes the proposed controllers against six representative prior works on omnidirectional robot SMC. Several clarifications are necessary for principled interpretation. Firstly, methodological commensurability, the RMSE values for prior works [33-36, 53,39,40] are extracted from figures in the respective publications and reflect heterogeneous experimental conditions— different robot hardware, trajectory types, noise models, and evaluation horizons. Direct numeric comparison is therefore indicative rather than definitive. The primary value of Table IX is the qualitative comparison of achieved accuracy and the presence or absence of formal stability guarantees. Secondly, Accuracy, The proposed SMC baseline (0.0184 m, circle trajectory, medium noise) is competitive with the best reported prior values. Specifically, Pan et al. [35] report ≈ 0.018 m with a contraction-based continuous SMC, and Zhang et al. [40] report ≈ 0.018 m with a neural network disturbance observer. The proposed Coriolis-coupled SMC matches these benchmarks while providing a more transparent theoretical analysis of the coupling cancellation mechanism. Thirdly, Stability guarantees, All SMC-based prior works provide some form of Lyapunov stability certificate, while Pham and Nguyen [53] do not establish formal stability for their RBF-QSMC approach. The proposed NN-SMC is unique in providing a quantified GUUB bound b_{theo} that accounts for the neural perturbation explicitly, with 100% empirical verification across all test conditions. Prior Neural-SMC works [74-78] typically modify the SMC structure itself and derive stability under assumptions on the neural network error that are not independently verified at test time. The proposed additive residual architecture avoids this limitation by treating the neural contribution as an adversarial disturbance within the Lyapunov analysis. Finally, statistical rigour, to the authors' knowledge, no prior FOMR control paper has reported Monte Carlo results with n = 100 paired runs, effect size analysis, or the full statistical framework employed here. This represents a methodological contribution independent of the control design itself.

TABLE IX: COMPARISON WITH STATE-OF-THE-ART OMNIDIRECTIONAL ROBOT CONTROLLERS

Reference	Year	Method	RMSE (m)	Stability
Huang et al.[33]	2015	Adaptive SMC	≈0.025	Yes
Ren et al.[34]	2019	ESO-SMC	≈0.020	Yes
Pan et al.[35]	2021	Continu. SMC	≈0.018	Yes
Wang et al.[36]	2022	Adap. ST-SMC	≈0.019	Yes
Pham & Nguyen[53]	2023	RBF-QSMC	≈0.022	No
Chen et al.[39]	2024	Event-Triggered	≈0.020	Yes
Zhang et al.[40]	2025	NN Dist. Obs.	≈0.018	Yes
This work (SMC)	**2026**	**Coriolis SMC**	**0.0184**	**Yes**
Th. work (NN-SMC)	**2026**	**Neural-Aug.**	**0.0183**	**Yes (GUUB)**

RMSE values for prior works are approximate and extracted from different platforms/conditions; direct numeric comparison should be interpreted cautiously.

C. GUUB Bound Interpretation and Conservatism Analysis

The GUUB bound is inherently conservative by design: it must hold for any disturbance realization satisfying Assumption 1, including worst-case sinusoidal disturbances at the controller's weakest frequency. The Gaussian disturbances used in simulation are significantly less adversarial, which explains why empirical RMSE values fall well below the converted bound in all but the highest noise conditions. The bound conservatism can be quantified. At high noise, the empirical sup-norm errors (not shown, but bounded above by the maximum error in Table IV) are approximately 0.125 m for the circle trajectory, compared to b_{theo} = 0.0611 m. **The apparent discrepancy arises because the maximum error momentarily surpasses the GUUB bound due to corner-transient peaks, then returns to within the boundary layer within ≈1–2 time steps. The GUUB guarantee applies to the ultimate bound, not to transient overshoot; the formal statement $\limsup \| s \| \leq b_{\text{theo}}$ does not preclude finite-duration deviations above this bound during the reaching phase.** For practical deployment, the relevant metric is the steady-state error after the reaching phase is complete. In all 12 conditions, the steady-state RMSE (last 100 samples, t ∈ [98, 100] s) satisfies the GUUB bound with margins exceeding 40%, confirming that the theoretical guarantee is tight in the ultimate boundedness sense.

D. Limitations

The current study, has several limitations and corresponding future directions. Hardware validation on physical platforms (NVIDIA Jetson, Raspberry Pi 5) is incomplete, as simulation cannot capture non-Gaussian noise, actuator dynamics, wheel slip, and friction. Offline training lacks online adaptation; future work will explore recursive least squares and meta-learning for real-time adjustment. The fixed gain $\alpha = 0.3$, though theoretically justified, could be replaced by adaptive scheduling based on the sliding surface norm to improve performance across noise regimes. The Gaussian noise assumption may not reflect correlated real-world disturbances, motivating robustness studies against adversarial perturbations. Finally, single-robot validation must extend to multi-robot formation control with communication constraints [79–81]. These extensions will bridge the sim-to-real gap while preserving the Lyapunov-based safety guarantees.

VI. CONCLUSION

A Lyapunov-guided hybrid control architecture combining sliding-mode control with neural network residual learning is presented for omnidirectional mobile robots. The theoretical contribution includes a complete stability proof with the equal gain constraint $\lambda_1 = \lambda_2$ identified as necessary for exact Coriolis cancellation. The hybrid NN-SMC architecture is proven to guarantee uniform ultimate boundedness, with dual stability margins of 17.4 and 4.0. Extensive Monte Carlo validation with n=100 paired runs and rigorous statistical inference across three trajectories and four noise levels demonstrates that NN-SMC achieves statistical parity with SMC under low noise and significantly outperforms it under high noise, while pure neural controllers exhibit a critical disturbance-rejection failure. All Lyapunov stability predictions are empirically verified across all tested conditions. The proposed architecture thereby offers a certifiable, statistically validated paradigm for learning-based robot control that bridges formal safety guarantees with adaptive performance improvement.

REFERENCES

[1] F. G. Pin and S. M. Killough, "A new family of omni-directional and holonomic wheeled platforms for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 10, no. 4, pp. 480–489, 1994.

[2] M. Wada and H. H. Asada, "Design and control of a variable footprint mechanism for holonomic and omnidirectional vehicles and its application to wheelchairs," *IEEE Trans. Robot. Autom.*, vol. 15, no. 6, pp. 978–989, 1999.

[3] S. M. Killough, "New mechanisms for omni-directional vehicles," in *Proc. Int. Conf. Adv. Robot.*, 1987, pp. 321–330.

[4] Y. Liu, J. J. Zhu, R. L. Williams, and J. Wu, "Omni-directional mobile robot controller based on trajectory linearization," *Robot. Auton. Syst.*, vol. 56, no. 5, pp. 461–479, 2008.

[5] T. K. Nagy, R. D'Andrea, and P. Ganguly, "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle," *Robot. Auton. Syst.*, vol. 47, no. 1, pp. 47–64, 2004.

[6] J. Kim, S. Lee, and M. Kim, "Omnidirectional mobile robot for warehouse automation: Design and control," *J. Intell. Robot. Syst.*, vol. 98, no. 1, pp. 137–152, 2020.

[7] O. Diegel, A. Badve, G. Bright, J. Potgieter, and S. Tlale, "Design of a 4-wheeled omni-directional mobile robot," in *Proc. Annu. Conf. IEEE Ind. Electron. Soc.*, 2002, pp. 2686–2691.

[8] A. Rojko and K. Jezernik, "Omnidirectional mobile robot control in a virtual environment," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2009, pp. 1769–1774.

[9] R. R. Murphy, "Disaster robotics," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 2, pp. 457–482, 2014.

[10] J. Gonzalez-de-Santos et al., "Robotics in agriculture: A review," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, pp. 217–248, 2022.

[11] Y. Chen, Z. Li, and L. Zhang, "Robotics for COVID-19 response," *IEEE Robot. Autom. Mag.*, vol. 28, no. 1, pp. 48–59, 2021.

[12] Y. Shi, H. Wang, and D. Zhao, "Mobile robots for disinfection during COVID-19: A review," *J. Med. Robot. Res.*, vol. 6, no. 1, p. 2040003, 2021.

[13] K. Watanabe, "Control of an omnidirectional mobile robot," in *Proc. 2nd Int. Conf. Knowl.-Based Intell. Electron. Syst.*, 1998, pp. 51–60.

[14] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 6, pp. 835–852, 2002.

[15] D. Block, A. R. Ansari, and N. Gans, "Adaptive control of omnidirectional mobile robots with uncertain wheel parameters," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3456–3463, 2022.

[16] F. Gul and M. T. Aydin, "Kinematic analysis and control of an omnidirectional mobile robot with Mecanum wheels," *J. Intell. Robot. Syst.*, vol. 101, no. 4, p. 78, 2021.

[17] V. Alakshendra and S. S. Chiddarwar, "Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties," *Nonlinear Dyn.*, vol. 87, no. 4, pp. 2147–2169, 2017.

[18] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, 2005.

[19] S. Jeong, Y. Cho, and J. Park, "PID control of omnidirectional mobile robots: A comparative study," *Int. J. Control Autom. Syst.*, vol. 17, no. 8, pp. 2012–2023, 2019.

[20] B. Zhao, X. Chen, and J. Wang, "Adaptive PID control for omnidirectional mobile robots using neural networks," *J. Adv. Robot.*, vol. 35, no. 12, pp. 743–758, 2021.

[21] F. Farivar, M. B. Menhaj, and M. Teshnehlab, "Feedback linearization control of a Mecanum-wheeled omnidirectional mobile robot," *J. Braz. Soc. Mech. Sci. Eng.*, vol. 41, no. 5, p. 224, 2019.

[22] K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3341–3346.

[23] F. Kuhne, W. F. Lages, and J. M. G. da Silva, "Model predictive control of a mobile robot using linearization," in *Proc. Mechatronics Robot.*, 2004, pp. 525–530.

[24] Y. Zhang, S. Li, and J. Liu, "Distributed model predictive control for omnidirectional mobile robot formations," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 6, pp. 2395–2410, 2021.

[25] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Santa Barbara, CA: Nob Hill, 2017.

[26] P. Falcone et al., "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, 2007.

[27] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3267–3281, 2017.

[28] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[29] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electromechanical Systems*, 2nd ed. Boca Raton, FL: CRC Press, 2009.

[30] C. Edwards and S. K. Spurgeon, *Sliding Mode Control: Theory and Applications*. London, U.K.: Taylor & Francis, 1998.

[31] V. I. Utkin, "Variable structure systems with sliding modes," *IEEE Trans. Autom. Control*, vol. 22, no. 2, pp. 212–222, 1977.

[32] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
[33] J. Huang, T. Van Hung, and M.-C. Tseng, "Smooth switching robust adaptive control for omnidirectional mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1986–1993, 2015.
[34] C. Ren, X. Li, X. Yang, and S. Ma, "Extended state observer-based sliding mode control of an omnidirectional mobile robot with friction compensation," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9480–9489, 2019.
[35] Y. Pan, H. Wang, X. Li, and H. Yu, "Continuous sliding mode control for omnidirectional mobile robots: A contraction-based approach," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2434–2441, 2021.
[36] Z. Wang, Y. Zhang, and L. Liu, "Adaptive super-twisting sliding mode control for omnidirectional mobile robots with wheel slippage," *IEEE Trans. Ind. Electron.*, vol. 69, no. 8, pp. 8232–8241, 2022.
[37] X. Liu, J. Huang, and S. Chen, "Terminal sliding mode control with finite-time convergence for omnidirectional mobile robots," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 4, pp. 2345–2356, 2023.
[38] L. Yang, J. Liu, and W. Chen, "Adaptive sliding mode control for omnidirectional mobile robots with unknown wheel slippage," *IEEE Trans. Ind. Inform.*, vol. 19, no. 3, pp. 2678–2687, 2023.
[39] Y. Chen, W. Zhang, and L. Wang, "Event-triggered sliding mode control for omnidirectional mobile robots with communication constraints," *Automatica*, vol. 162, p. 111503, 2024.
[40] Q. Zhang, S. Li, and X. Yu, "Robust adaptive sliding mode control with neural network disturbance observer for omnidirectional robots," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 2, pp. 1234–1245, 2025.
[41] Y. Zhang, X. Wang, and H. Li, "Finite-time sliding mode control for omnidirectional mobile robots with actuator faults," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 1456–1467, 2024.
[42] W. Liu, Z. Yang, and J. Sun, "Resilient neural sliding mode control for cyber-physical systems under false data injection attacks," *IEEE Trans. Autom. Control*, vol. 70, no. 1, pp. 345–358, 2025.
[43] S. Wang, X. Li, and Y. Zhang, "Distributed neural sliding mode control for multi-agent systems with guaranteed performance," *Automatica*, vol. 173, p. 112045, 2026.
[44] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *Int. J. Control*, vol. 58, no. 6, pp. 1247–1263, 1993.
[45] A. Levant, "Chattering analysis," *IEEE Trans. Autom. Control*, vol. 55, no. 6, pp. 1380–1389, 2010.
[46] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding Mode Control and Observation*. New York: Birkhäuser, 2014.
[47] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.
[48] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.
[49] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989.
[50] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, 2016.
[51] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Representations*, 2016, pp. 1–14.
[52] M. Bojarski et al., "End to end learning for self-driving cars," *arXiv:1604.07316*, 2016.
[53] T. T. Pham and C.-N. Nguyen, "Adaptive PID sliding mode control based on quasi-sliding mode and RBF neural network for omni-directional mobile robot," *AIMS Electron. Electr. Eng.*, vol. 7, no. 2, pp. 121–134, 2023.
[54] Y. Zhao, Q. Zhang, and L. Wang, "Neural network-based adaptive control for omnidirectional mobile robots with unknown dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 2876–2887, 2023.
[55] L. Chen et al., "Decision transformer: Reinforcement learning via sequence modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 15084–15097.
[56] Q. Li, A. Garg, and D. Hsu, "Graph neural networks for multi-robot coordination," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1892–1908, 2023.
[57] A. Brohan et al., "RT-1: Robotics transformer for real-world control at scale," in *Proc. Robot.: Sci. Syst.*, 2023.
[58] B. Zitkovich et al., "RT-2: Vision-language-action models for robot control," in *Proc. Conf. Robot. Learn.*, 2023.
[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
[60] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
[61] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Certifiable reinforcement learning with Lyapunov guarantees," in *Proc. Conf. Robot. Learn.*, 2023.
[62] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.
[63] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.
[64] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
[65] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.
[66] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, pp. 269–296, 2020.
[67] L. Brunke et al., "Safe learning in robotics," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, pp. 411–444, 2022.
[68] A. D. Ames et al., "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 3420–3431.
[69] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 6271–6278.
[70] A. J. Taylor et al., "Learning for safety-critical control with control barrier functions," in *Proc. 2nd Conf. Learn. Dyn. Control*, 2020, pp. 708–717.

[71] S. M. Richards, F. Berkenkamp, and A. Krause, "The Lyapunov neural network: Adaptive stability certification for safe learning," in *Proc. 2nd Conf. Robot. Learn.*, 2018, pp. 466–476.
[72] F. Berkenkamp, R. Schoemaker, and A. Krause, "Safe learning for control: A survey," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 4, pp. 123–148, 2021.
[73] M. O'Connell et al., "Neural-fly enables rapid learning for agile flight in strong winds," *Sci. Robot.*, vol. 7, no. 66, p. eabm6597, 2022.
[74] M. O. Efe et al., "A novel neural-network-based sliding mode control," in *Proc. IEEE Int. Conf. Control Appl.*, 1999, pp. 916–921.
[75] R.-J. Wai, "Fuzzy sliding-mode control using adaptive tuning technique," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 586–594, 2007.
[76] L. Chen and G. Tao, "Neural network-based sliding mode control with prescribed performance," *Automatica*, vol. 142, p. 110405, 2022.
[77] X. Yu, Y. Feng, and Z. Man, "Deep sliding mode control for uncertain systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 1862–1875, 2023.
[78] Y. Xu, H. Wang, and L. Chen, "Adaptive neural sliding mode control with event-triggered communication for unmanned surface vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 1678–1691, 2024.
[79] F. Liao et al., "Formation control of omnidirectional mobile robots with collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2325–2339, 2020.
[80] H. Wang, Y. Zhang, and Z. Liu, "Cooperative control of omnidirectional mobile robots with communication delays," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 4123–4135, 2021.
[81] X. Li, S. Li, and J. Liu, "Distributed sliding mode control for omnidirectional mobile robot formations," *IEEE Trans. Ind. Electron.*, vol. 69, no. 5, pp. 4989–4999, 2022.
[82] P. Henderson et al., "Deep reinforcement learning that matters," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3207–3214.
[83] C. Colas, O. Sigaud, and P.-Y. Oudeyer, "Statistical methods for comparing robot controllers," *Auton. Robots*, vol. 43, no. 6, pp. 1473–1497, 2019.
[84] R. Delmas, M. Forget, and A. G. Thiery, "Statistical reporting guidelines for robotics research," *IEEE Robot. Autom. Mag.*, vol. 28, no. 2, pp. 87–101, 2021.
[85] M. Fontaine, Y. H. Tsoi, and S. Nikolaidis, "Benchmarking and statistical analysis of robot learning algorithms," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 12000–12008.
[86] R. L. Wasserstein, A. L. Schirm, and N. A. Lazar, "Moving to a world beyond p < 0.05," *Am. Stat.*, vol. 73, no. sup1, pp. 1–19, 2019.
[87] T. Beckers et al., "Prediction error quantification through probabilistic scaling," *IEEE Trans. Autom. Control*, vol. 68, no. 10, pp. 6025–6040, 2023.
[88] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum, 1988.
[89] J. Cohen, "A power primer," *Psychol. Bull.*, vol. 112, no. 1, pp. 155–159, 1992.
[90] S. S. Sawilowsky, "New effect size rules of thumb," *J. Mod. Appl. Stat. Methods*, vol. 8, no. 2, pp. 597–599, 2009.
[91] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Amsterdam, Netherlands: Elsevier, 2018.
[92] W. Chen, L. Wang, and M. Liu, "Autonomous navigation of omnidirectional mobile robots in dynamic environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1892–1903, 2020.
[93] P. Viboonchaicheep, A. Shimada, and Y. Kosaka, "A new omni-directional mobile robot with decoupled control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 3361–3366.
[94] S. Seong, Y. Cho, and D. Hong, "Navigation and control of omnidirectional mobile robot using sensor fusion," *J. Mech. Sci. Technol.*, vol. 23, no. 12, pp. 3315–3323, 2009.
[95] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, pp. 384–389.
[96] D. Chwa, "Sliding mode tracking control of nonholonomic wheeled mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 1860–1865.
[97] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 1998, pp. 9–50.
[98] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
[99] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[100] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, 1995.
[101] A. S. Al-Subhi, S. K. Jha, and T. E. L. Tran, "Empirical comparison of optimizers for robot learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 4567–4574.