Trajectory tracking double two-loop adaptive neural network control for a Quadrotor

Final report

Nguyen Nhu Toan

[1]Hanoi Univesity of Science and Technology
Hanoi, Vietnam

Hanoi, June 2023

Table of Contents

Introduction

- In this paper, the development and experimental validation of a novel double two-loop nonlinear controller based on adaptive neural networks for a quadrotor are presented.
- The proposed controller has a two-loop structure: an outer loop for position control and an inner loop for attitude control
- The functionality of the proposed scheme is demonstrated experimentally, and its performance is compared with another adaptive neural-based scheme and a model-based scheme.

Quadrotor Dynamic Model

The quadrotor dynamic model in the inertial reference frame is given by:

$$\begin{aligned} m\ddot{p} + mg_z + D_p(\eta)\dot{p} &= r_3(\eta)F + \delta_p(t) \\ M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} + D_\eta(\eta)\dot{\eta} &= \Phi(\eta)^{-T}\tau + \delta_\eta(t) \end{aligned} \quad (1)$$

- m: Quadrotor mass
- $g_z = \begin{bmatrix} 0 & 0 & g \end{bmatrix} \in \mathbb{R}^3$: Acceleration due to gravity
- $p = \begin{bmatrix} x & y & z \end{bmatrix} \in \mathbb{R}^3$: Quadrotor position
- $\eta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix} \in \mathbb{R}^3$: Quadrotor attitude
- $D_p(\eta) \in \mathbb{R}^{3\times 3}$: Aerodynamic drag matrix
- $D_\eta(\eta) \in \mathbb{R}^{3\times 3}$: Aerodynamic damping matrix
- $r_3(\eta) \in \mathbb{R}^3$: Third column of rotation matrix $R(\eta) \in SO(3)$
- $\delta_p(t),\ \delta_\eta(t) \in \mathbb{R}^3$: Unknown external disturbances and unmodeled dynamics $||\delta_p(t)||, ||\delta_\eta(t)|| \le \delta$ for $t \ge 0$.

Quadrotor Dynamic Model

The quadrotor has four control inputs related to the four actuators of the quadrotor $F \in \mathbb{R}$ and $\tau = \begin{bmatrix} \tau_\phi & \tau_\theta & \tau_\psi \end{bmatrix} \in \mathbb{R}^3$, which are expressed as:

$$F = 4\sum_{i=1}^{4} T_i,$$

$$\tau = \begin{bmatrix} l(T_2 - T_4) \\ l(T_3 - T_1) \\ k_\tau \sum_{i=1}^{4}(-1)^{i+1} T_i \end{bmatrix}$$

where $T = \begin{bmatrix} T_1 & T_2 & T_3 & T_4 \end{bmatrix} \in \mathbb{R}^4$ and $T_i \in \mathbb{R}$ is the thrust of the i-th actuator. The above relationship can be expressed in matrix form as $[F \ \tau]^T = BT$, where

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -l & 0 & l \\ -l & 0 & l & 0 \\ -k_\tau & k_\tau & -k_\tau & k_\tau \end{bmatrix}.$$

The quadrotor control goal is to design control inputs $F(t)$ and $\tau(t)$ such that the position and attitude errors converge to zero as t approaches infinity.

- Desired position: $p_d(t) = [x_d(t)\ y_d(t)\ z_d(t)]^T$
- Desired attitude: $\eta_d(t) = [\phi_d(t)\ \theta_d(t)\ \psi_d(t)]^T$
- $\phi_d(t)$ and $\theta_d(t)$ are computed by the position controller
- Assumed to be at least twice time-differentiable and bounded until its second time-derivative

Position and attitude error vectors:

$$e_p(t) = p_d(t) - p(t) \tag{2}$$
$$e_\eta(t) = \eta_d(t) - \eta(t) \tag{3}$$

Control goal:

$$\lim_{t\to\infty} \begin{bmatrix} \dot{e}_p(t) & e_p(t) & \dot{e}_\eta(t) & e_\eta(t) \end{bmatrix} = 0$$

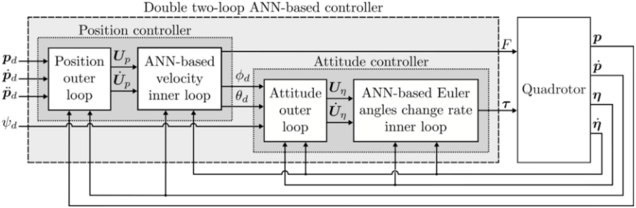

Figure: Control diagram

- The position and attitude controllers have each a two-loop structure and include an adaptive neural network to compensate for disturbances.
- In both cases, the outer loops generate commands to feed the inner loops where the ANNs are employed.

The position outer loop defines a velocity reference signal $U_p \in \mathbb{R}^3$ given by

$$U_p = \dot{p}_d + K_p e_p \tag{4}$$

where $K_p \in \mathbb{R}^{3\times 3}$ is a positive definite diagonal gain matrix.
Taking the time derivative of (3) and substituting the velocity reference signal in (4), the following is obtained:

$$\dot{e}_p = -K_p e_p + U_p - \dot{p} \tag{5}$$

Defining the velocity error as

$$\tilde{U}_p = U_p - \dot{p} \tag{6}$$

the position error dynamics result in:

$$\dot{e}_p = -K_p e_p + \tilde{U}_p \tag{7}$$

It can be observed that as long as $\tilde{U}_p \to 0$ is guaranteed, then $e_p \to 0$ as $t \to \infty$.

Control Method
Position Control, Inner-loop controller

Taking the time derivative of the velocity error (4), substituting the position dynamics in (1) on it, one gets:

$$m\dot{\tilde{U}}_p = m\dot{U}_p + mgz + D_p(\eta)\dot{p} - \delta_p(t) - r_3(\eta)F. \tag{8}$$

Let the mass estimation error as $\tilde{m} = m - \hat{m}$, with $\hat{m}_d$ denoting the mass estimation.
Assuming that the attitude dynamics are faster than the position dynamics, the attitude error converges to zero first, and the term $[r_3(e_\eta + \eta) - r_3(\eta)]F = 0$, leading to:

$$m\dot{\tilde{U}}_p = \hat{m}(\ddot{p}_d + gz) + \hat{m}K_p\dot{e}_p + \tilde{m}(\dot{U}_p + gz) + D_p(\eta)\dot{p} - \delta_p(t) - r_3(\eta_d)F. \tag{9}$$

Considering the time derivative of (4), it can be expressed as:

$$m\dot{\tilde{U}}_p = \hat{m}(\ddot{p}_d + g_z) + f_{W_p}(\dot{p}, \eta, \dot{U}_p) - r_3(\eta_d)F, \tag{10}$$

where $f_{W_p}(\dot{p}, \eta, \dot{U}_p) = \hat{m}K_p\dot{e}_p + \tilde{m}(\dot{U}_p + gz) + D_p(\eta)\dot{p}$.

Control Method

Position Control, Inner-loop controller

The neural network in Figure 2 is used to estimate $f_{W_p}(\dot{p}, \eta, \dot{U}_p)$ as follows:

$$f_{W_p}(\gamma_p) = W_p^T \sigma(V_p^T \gamma_p) + \varepsilon_p, \tag{11}$$

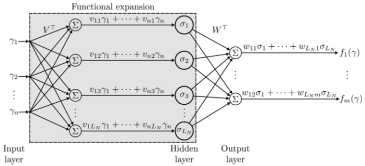

Figure: FLNN structure

- $\gamma_p = [1 \ \dot{p}^T \ \eta^T \ \dot{U}_p^T]^T \in \mathbb{R}^{10}$: neural network extended input vector.
- $V_p^T \in \mathbb{R}^{10\times L_p}$ and $W_p^T \in \mathbb{R}^{L_p \times 3}$: ideal weights input and output matrices, $\|V_p^T\|_F \leq V_P$ and $\|W_p^T\|_F \leq W_P$.
- $\varepsilon_p \in \mathbb{R}^3$: neural network approximation error, $0 < \|\varepsilon_p\| \leq \varepsilon_P$.
- $\sigma(x)$: the neural network activation function (tanh)

Therefore, Eq 9 can be written as:

$$m\dot{\tilde{U}}_p = \hat{m}(\ddot{p}_d + g_z) + W_p^T\sigma_p + \epsilon_p - r_3(\eta_d)F. \tag{12}$$

where $\epsilon_p = \varepsilon_p - \delta_p(t)$ is bounded as $0 < \|\varepsilon_p\| \leq \varepsilon_P$. Thus, the following definition for $r_3(\eta_d)F = [f_x \ f_y \ f_z]^T$ is proposed:

$$r_3(\eta_d)F = K_{Up}\tilde{U}_p + K_{ip}\xi_p + \hat{m}(\ddot{p}_d + g_z) + \hat{W}_p^T\sigma_p + \alpha_p \mathrm{sign}(\tilde{U}_p), \tag{13}$$

$$\dot{\xi}_p = \tilde{U}_p. \tag{14}$$

The output weights adaptation law is given by:

$$\dot{\hat{W}}_p = -N_p(\sigma_p\tilde{U}_p^T - \kappa_p\|\tilde{U}_p\|\hat{W}_p), \tag{15}$$

where $\tilde{W}_p = W_p - \hat{W}_p$ is the output weight matrix estimation error, and $\mathrm{sign}(\tilde{U}_p) = [\mathrm{sign}(\tilde{U}_p^1) \ \ \mathrm{sign}(\tilde{U}_p^2) \ \ \mathrm{sign}(\tilde{U}p^3)]^T \in \mathbb{R}^3$. $\hat{W}_p$ is an estimation of the output weight matrix W_p, and $\alpha_p, \kappa_p, K_{Up}, K_{ip}, N_p > 0$

The total thrust and desired roll and pitch angles can be obtained as follows:

$$F(t) = \frac{f_z}{\cos(\phi)\cos(\theta)} \tag{16}$$

$$\phi_d(t) = \tan^{-1}\left[\frac{1}{f_z}\left(\cos(\theta_d)(\sin(\psi_d)\cos(\phi_d) - \cos(\psi_d)\sin(\phi_d)\right)\right)\right] \tag{17}$$

$$\theta_d(t) = \tan^{-1}\left[\frac{1}{f_z}\left(\cos(\psi_d)\cos(\phi_d) + f_z\sin(\psi_d)\sin(\phi_d)\right)\right] \tag{18}$$

The velocity error dynamics 10 with the controller 13–16 results in the following closed-loop system:

$$\dot{\tilde{U}}_p = -K_{U_p}\tilde{U}_p - K_{ip}\xi_p + \tilde{W}_p^T\sigma_p + \varepsilon_p - \alpha_p \text{sign}(\tilde{U}_p). \tag{19}$$

Then, the following Lyapunov function is defined:

$$V_p = \frac{\beta_p}{2} e_p^T e_p + \frac{m}{2} \tilde{U}_p^T \tilde{U}_p + \frac{1}{2} \xi_p^T K_{ip} \xi_p + \frac{1}{2} \mathrm{Tr}\{ \tilde{W}_p^T N_p^{-1} \tilde{W}_p \} \tag{20}$$

Proposition 1

Assume gain matrices K_p, K_{Up}, and N_p to be positive definite diagonal matrices and positive constants α_p and κ_p satisfying the condition $\alpha_p \geq \kappa_p W_P^2 + \varepsilon_P$. Then, for all initial conditions starting as some compact set, the solutions $e_p(t)$, $\dot{e}_p(t)$, and $\tilde{U}_p(t)$ of the overall closed-loop system in (7), (15), and (19) converge to zero as time goes to infinity, i.e., $e_p(t)$, $\dot{e}_p(t)$, $\tilde{U}_p(t) \to 0$ while $t \to \infty$. Furthermore, the state variable $\xi_p(t)$ and the output weight matrix estimation error $\tilde{W}_p(t)$ remain bounded for all time $t \geq 0$.

The attitude outer-loop defines an Euler angles change rate reference signal $U_\eta \in \mathbb{R}^3$ given by

$$U_\eta = \dot{\eta}_d + K_\eta e_\eta \tag{21}$$

where $K_\eta \in \mathbb{R}^{3\times 3}$ is a positive definite diagonal gain matrix. Considering the attitude error in (3), taking its time derivative and substituting the Euler angles change rate reference signal in (21), the following is obtained:

$$\dot{e}_\eta = -K_\eta e_\eta + U_\eta - \dot{\eta} \tag{22}$$

Defining the Euler angles change rate error as $\tilde{U}_\eta = U_\eta - \dot{\eta}$, the attitude error dynamics (22) results in

$$\dot{e}_\eta = -K_\eta e_\eta + \tilde{U}_\eta \tag{23}$$

where it can be observed that as long as $\tilde{U}_\eta \to 0$ is guaranteed, then $e_\eta \to 0$ while $t \to \infty$.

Taking the time derivative of (21) and substituting the attitude dynamics (1) on it yields:

$$\dot{\tilde{U}}_\eta = \dot{U}_\eta - M(\eta)^{-1}[\Phi(\eta)^T + \delta_\eta(t) - (C(\eta, \dot{\eta}) + D_\eta(\eta))\dot{\eta}] \qquad (24)$$

Based on (24), assume the existence of $\tau_{MB}(t)$, an external disturbances and exact model compensator given by

$$\tau_{MB} = \Phi(\eta)^T [f_\eta(\eta, \dot{\eta}, \dot{U}_\eta) - \delta_\eta(t)] \qquad (25)$$

where $f_\eta(\eta, \dot{\eta}, \dot{U}_\eta) = M(\eta)\dot{U}_\eta + (C(\eta, \dot{\eta}) + D_\eta(\eta))\dot{\eta}$. Finally, the model compensator can be approximated with a neural network as follows:

$$\tau_{MB} = \Phi(\eta)^T [W_\eta^T \sigma_\eta(V_\eta^T \gamma_\eta) + \epsilon_\eta - \delta_\eta(t)] \qquad (26)$$

where $\gamma_\eta = [1, \eta^T, \dot{\eta}^T, \dot{U}_\eta^T]^T \in \mathbb{R}^{10}$, $V_\eta \in \mathbb{R}^{10\times L_\eta}$ and $W_\eta \in \mathbb{R}^{L_\eta \times 3}$, $\|V_\eta\|_F \leq V$ and $\|W_\eta\|_F \leq W$, $\epsilon_\eta \in \mathbb{R}^3$, $0 < \|\epsilon_\eta\| \leq \epsilon$, $L_\eta \in \mathbb{N}$, and $\sigma(x)\forall x \in \mathbb{R}^n$.

Based on Eqs. (25), (27), the following neural network-based controller is proposed:

$$\tau = \Phi(\eta)^T [K_{U_\eta} \tilde{U}_\eta + K_{i\eta}\xi_\eta + \hat{W}_\eta^T \sigma_\eta + \alpha_\eta \mathrm{sign}(\tilde{U}_\eta)] \tag{27}$$

$$\dot{\xi}_\eta = \tilde{U}_\eta \tag{28}$$

where $K_{U_\eta}, K_{i\eta} \in \mathbb{R}^{3\times3}, \alpha_\eta \in \mathbb{R} > 0$, $\hat{W}_\eta$ is an estimation of the output weight matrix W_η with

$$\dot{\tilde{W}}_\eta = -N_\eta(\sigma_\eta \tilde{U}_\eta^T - \kappa_\eta \|\tilde{U}_\eta\| \hat{W}_\eta) \tag{29}$$

where the output weight matrix estimation error $\tilde{W}_\eta = W_\eta - \hat{W}_\eta$. $N_\eta \in \mathbb{R}^{3\times3} > 0$ is the neural network adaptation gains, and $\kappa_\eta \in \mathbb{R} > 0$.

Based on Eq (25), the controller (27) can be rewritten as

$$\tau = \Phi(\eta)^T \Big[K_{U_\eta} \tilde{U}_\eta + K_{i\eta}\xi_\eta + M(\eta)\dot{U}_\eta + (C(\eta, \dot{\eta}) \\ + D_\eta(\eta))\dot{\eta} - \delta_\eta(t) - \epsilon_\eta - \tilde{W}_\eta^T \sigma_\eta + \alpha_\eta \mathrm{sign}(\tilde{U}_\eta) \Big], \tag{30}$$

which in closed-loop with the angular velocity error dynamics (24), results in the following:

$$M(\eta)\dot{\tilde{U}}_\eta = -K_{U_\eta}\tilde{U}_\eta - K_{i\eta}\xi_\eta + \tilde{W}_\eta^T \sigma_\eta + \epsilon_\eta - \alpha_\eta \mathrm{sign}(\tilde{U}_\eta), \tag{31}$$

where $\epsilon_\eta = \epsilon_\eta - \delta_\eta(t)$ is bounded as $0 < \|\epsilon_\eta\| \leq \epsilon$.

Then, define the following Lyapunov function:

$$V_\eta = \frac{\beta_\eta}{2} e_\eta^T e_\eta + \frac{1}{2}\tilde{U}_\eta^T M(\eta)\tilde{U}_\eta + \frac{1}{2}\xi_\eta^T K_{i\eta}\xi_\eta + \frac{1}{2}\mathrm{Tr}\{\tilde{W}_\eta^T N_\eta^{-1}\tilde{W}_\eta\} \quad (32)$$

Proposition 2

Assume gain matrices K_η, K_{U_η}, and N_η to be positive definite diagonal matrices and positive constants α_η and κ_η satisfying the condition $\alpha_\eta \geq \kappa_\eta \|W\|^2 \frac{\epsilon}{4}$. Then, for all initial conditions starting as some compact set, the solutions $e_\eta(t)$, $\dot{e}_\eta(t)$, and $\tilde{U}_\eta(t)$ of the overall closed-loop system in (23), (28), (29), and (31) converge to zero as time goes to infinity, i.e., $e_\eta(t)$, $\dot{e}_\eta(t)$, $\tilde{U}_\eta(t) \rightarrow 0$ while $t \rightarrow \infty$. Furthermore, the state variable $\xi_\eta(t)$ and the output weight matrix estimation error $\tilde{W}_\eta(t)$ remain bounded for all time $t \geq 0$.

Experimental results

The parameters of the QBall2 quadrotor used in the experimental tests are listed in Fig 3. During the implementation, the mass of the quadrotor was estimated to be 70% of its nominal value.

QBall 2 quadrotor parameters.

Parameter	Description	Value	Units
g	Gravitational acceleration constant	9.81	m/s^2
m	Quadrotor mass	1.79	kg
I_{xx}	Inertia moment with respect to the axis x	0.03	kg m^2
I_{yy}	Inertia moment with respect to the axis y	0.03	kg m^2
I_{zz}	Inertia moment with respect to the axis z	0.04	kg m^2
$D_p(\boldsymbol{\eta})$	Aerodynamic drag coefficient matrix	diag{0.002, 0.002, 0.004}	kg/s
$D_\eta(\boldsymbol{\eta})$	Aerodynamic damping coefficient matrix	diag{0.002, 0.002, 0.004}	kg m^2/s

Figure: Qball2 quarotor paramters

The proposed controller will be referred to as DTLANNC (Double Two-Loop Adaptive Neural Network-Based Controller). It is compared with model base controller (TLMBC), and adaptive neural network controller (NNSMC). The DTLANNC gains were determined through a trial and error process.

Experimental results

The trajectory tracking task was evaluated using a tracking Lemniscate path. The desired signals for the path were defined as follows:

$$x_d(t) = 0.5\sin(2\pi \cdot 4t)\,[\mathrm{m}],$$

$$y_d(t) = \cos(2\pi \cdot 8t)\,[\mathrm{m}],$$

$$z_d(t) = \begin{cases} 1 - 0.7e^{-0.1t^3}\,[\mathrm{m}], & \text{if } t \leq 5 \\ 1\,[\mathrm{m}], & \text{if } t > 5 \end{cases}$$

$$\psi_d(t) = 0.0\,[^\circ]$$

It is important to note that the TLMBC, NNSMC, and DTLANNC controllers were implemented with an estimated mass value of $\hat{m} = 1.253\,[\mathrm{kg}]$, which corresponds to 70% of the nominal mass of the quadrotor mentioned in 3. The mass value is critical for takeoff and hovering, as it directly affects the thrust generated by the quadrotor's actuators.

Experimental Results

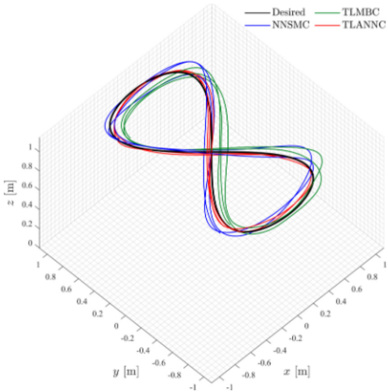

Figure: The paths traced by the quadrotor during the experimental tests, implementing the TLMBC, NNSMC, and DTLANNC schemes while tracking the trajectory

Experimental Results

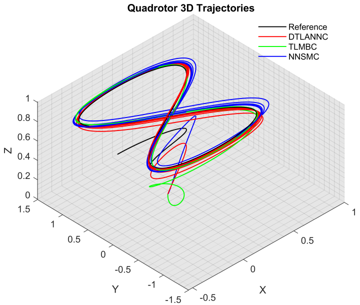

Figure: The paths traced by the quadrotor during the experimental tests, implementing the TLMBC, NNSMC, and DTLANNC schemes while tracking the trajectory

Experimental Results

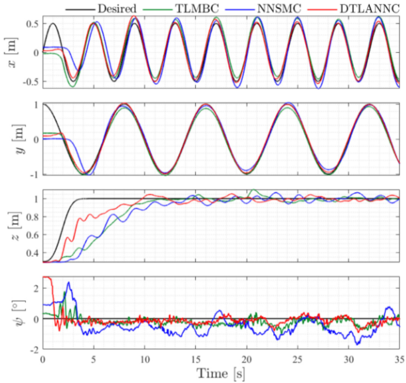

Figure: The time evolution of the position vector $p(t) = [x(t), y(t), z(t)]^T$ and the yaw angle $\psi(t)$ by implementing the TLMBC, NNSMC, and DTLANNC schemes in real-time experiments for performing the trajectory tracking task

Experimental Results

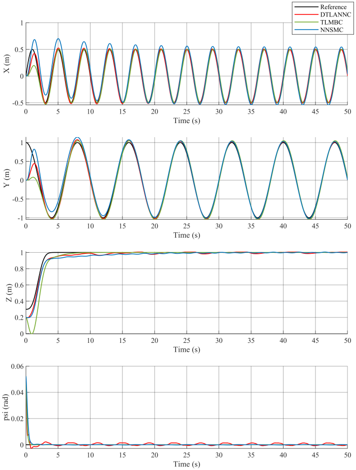

Figure: The time evolution of the position vector $p(t) = [x(t), y(t), z(t)]^T$ and the yaw angle $\psi(t)$ by implementing the TLMBC, NNSMC, and DTLANNC

Experimental Results

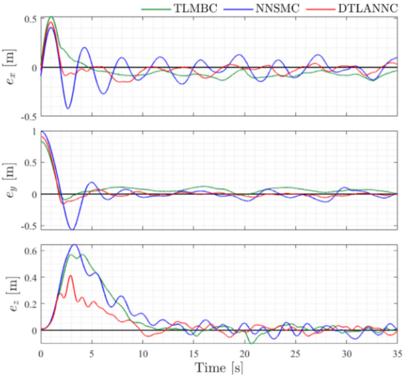

Figure: The time evolution of the position error $e_p(t) = [e_x(t)\ e_y(t)\ e_z(t)]^T$ by implementing the TLMBC, NNSMC, and DTLANNC schemes in real-time experiments for performing the trajectory tracking task

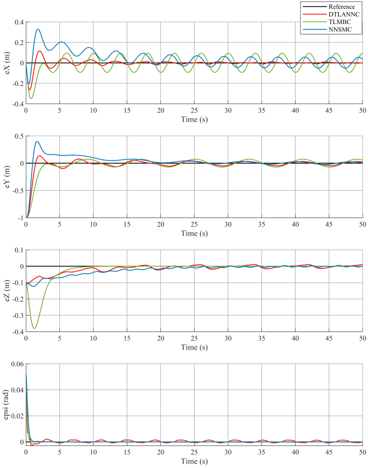

Figure: The time evolution of the position error $e_p(t) = [e_x(t)\ e_y(t)\ e_z(t)]^T$ by implementing the TLMBC, NNSMC, and DTLANNC schemes in real-time

Experimental Results

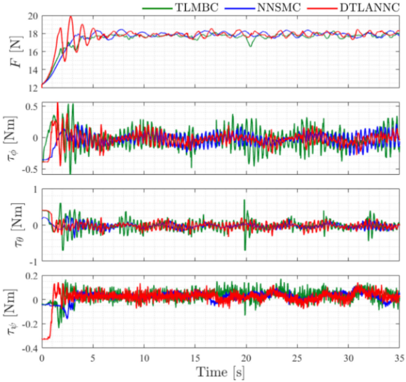

Figure: The time evolution of the control actions $F(t)$ and $\tau(t) = [\tau_\phi(t)\ \tau_\theta(t)\ \tau_\psi(t)]^T$ computed by the TLMBC, NNSMC, and DTLANNC schemes in real-time experiments performing the trajectory tracking task

Experimental Results

RMS values of position and attitude errors $\boldsymbol{e}_p(t)$ and $\boldsymbol{e}_\eta(t)$ obtained by implementing the TLMBC, NNSMC and DTLANNC schemes during the trajectory tracking task in the experimental tests.

Signal	TLMBC	NNSMC	P_{imp}%	DTLANNC	P_{imp}%
e_x [m]	0.0842	0.0731	13.18	0.0448	46.79
e_y [m]	0.0687	0.0580	15.57	0.0216	68.56
e_z [m]	0.0282	0.0342	-21.28	0.0238	15.60
e_ϕ [°]	1.4972	1.2835	14.27	1.1505	23.16
e_θ [°]	1.2705	1.3375	-5.27	1.1638	8.40
e_ψ [°]	0.3655	0.7915	-116.55	0.2834	22.46

Figure: The Root Mean Square (RMS) values of the position error $e_p(t)$ and attitude error $e_\eta(t) = [e_\phi(t)\ e_\theta(t)\ e_\psi(t)]^T$ obtained by implementing the TLMBC, NNSMC, and DTLANNC schemes during the trajectory tracking task.

- The relative percentage improvement is computed as $P_{\text{imp}}(\zeta) = \frac{\text{RMS(TLMBC)}-\text{RMS}(\zeta)}{\text{RMS(TLMBC)}} \times 100\%$, where ζ represents the RMS values obtained with the NNSMC or the DTLANNC schemes as corresponds.

Conclusion
Advantages

- Adaptability: The controller incorporates adaptive neural networks, allowing it to adapt to uncertain and varying quadrotor dynamics and disturbances.
- Trajectory tracking accuracy: The experimental results demonstrate that the proposed controller achieves accurate trajectory tracking in comparision with Model based control method and Adaptive neural network controller.
- Robustness to parameter uncertainties: The controller shows robustness to parametric uncertainties, as it performs well even when the mass estimate of the quadrotor is set to 70% of the nominal value.
- Fast convergence: The proposed scheme exhibits fast convergence of position and attitude errors, leading to quick stabilization of the quadrotor.

Conclusion
Disadvantages

- Computational complexity: The utilization of adaptive neural networks in the controller adds computational complexity to the control system.
- Training and tuning requirements: The design of the adaptive neural networks involves a trial-and-error process to determine the appropriate network gains and parameters.
- Lack of interpretability: Neural networks are often considered as black-box models, meaning that the inner workings of the network are not easily interpretable.
- Requirement of smooth reference path: The controller's performance relies on having a smooth reference trajectory for the quadrotor to track (first and second derivatives of reference path). Sudden or highly dynamic changes in the reference path may deteriorate the controller's performance.