Two-Layer Neuro-Adaptive Compensation Control Applied to a 4-Wheeled Omnidirectional Mobile Robot

Sergio López, *Member, IEEE,* and Miguel A. Llama, *Senior Member, IEEE*

***Abstract*—Thanks to recent advances in artificial intelligence, interest in autonomous mobile systems has increased, and consequently, the development and validation of advanced control schemes for them has also seen a rise. This work introduces a two-layer neuro-adaptive compensation control scheme designed to address the trajectory tracking problem for an omnidirectional wheeled mobile robot equipped with four independent Mecanum wheels. The two-layer artificial neural network is used to compensate for the unknown dynamics of the mobile robot; the filtered error technique is used to obtain the weights of the artificial neural network. This approach does not require offline training. A key contribution of this approach is the integration of a novel auxiliary signal to provide robustness, particularly in non-ideal scenarios. This robust term effectively bounds the disturbance commonly encountered in such control approaches. A significant advantage of this approach is its independence from precise knowledge of plant parameters or the overall plant dynamics. Experimental results demonstrate the effectiveness of the proposed controller in achieving desired performance for the 4-wheeled omnidirectional mobile robot.**

Link to graphical and video abstracts, and to code:
https://latamt.ieeer9.org/index.php/transactions/article/view/10018

***Index Terms*—Neuro-adaptive control, tracking control, online weight update, omnidirectional mobile robot, Mecanum wheels.**

I. Introduction

FEEDFORWARD dynamic neural networks with online weight updates are an upgraded version of traditional feedforward static neural networks. They can generate better-quality results and can learn from their surroundings as well [1]. One of the main application areas where the effectiveness of these architectures can be observed is the control of nonlinear systems. Additionally, there is significant interest in model-free controllers, which operate without requiring a plant model. Several techniques can achieve this, notably artificial neural networks (ANNs) [2], [3]. On the other hand, the maneuverability of a wheeled mobile robot depends on its configuration and wheel type. An interesting proposal for ground mobile robots is the 4-Wheeled Omnidirectional Mobile Robot (4-WOMR) that uses Mecanum wheels [4], which is used in this work.

The following works explore neural feedback control methods that eliminate the need for offline learning. Notably, [5] proposes a control design approach using ANNs and fuzzy logic, detailing feedback control architectures and weight-tuning algorithms to guarantee closed-loop stability and weight boundedness. Wang et al. [6] proposed an adaptive neural compensation strategy to control a micro-electro-mechanical system gyroscope. Simulations were used to confirm the effectiveness of the designed controller. [7] developed an adaptive neural control strategy for nonlinear robot manipulators. Their approach incorporates sliding mode techniques within the control law. [8] proposed an adaptive ANN control scheme using backstepping for uncertain robotic manipulators with external disturbances and time-varying output constraints. A direct adaptive robust emotional neuro-control approach was implemented on a real-world 3-spherical-prismatic-spherical parallel robot, as proposed in [9]. [10] developed an adaptive dynamic surface control strategy based on ANNs for a class of systems characterized by unknown time delays and nonlinearities in the input due to hysteresis. [11] presents a robust adaptive control method for industrial robotic manipulators, utilizing recurrent fuzzy wavelet ANNs. In [12], the control of an underactuated CMG is studied, focusing on trajectory tracking. A feedback linearization controller and an adaptive ANN are designed for greater robustness. Real-time experiments compare their performance with linear and PID-PID controllers, achieving good accuracy. [13] suggest a robust Neural Network-based Sliding Mode Controller to ensure trajectory tracking for an omnidirectional wheeled mobile manipulator system. [14] design a Neural Network Adaptive Sliding Mode Controller to handle the tracking problem for a 4-WOMR prototype. [15] presents a controller for a 4-WOMR with Mecanum wheels. A self-organizing fuzzy neural network is used to estimate the robot's uncertainties. Furthermore, a preview strategy based on Bézier curve trajectory replanning is proposed to optimize the curve following initiation. This scheme is validated in simulation. In [16], a predefined convergence-time trajectory control method for omnidirectional mobile robots with uncertainties is proposed using a fuzzy neural network. A predefined-time stability-based position controller is designed, allowing the upper bound on the convergence time to be explicitly set. For more accurate angular tracking, a Type 1 fuzzy neural network is used to estimate the uncertainty. The effectiveness of the method is verified by simulations.

The associate editor coordinating the review of this manuscript and approving it for publication was Roberto S. Murphy (*Corresponding author: Sergio López*).

Sergio Lopez, and M. A. Llama are with the Division of Graduate Studies and Research, at Tecnológico Nacional de México/Instituto Tecnológico de La Laguna, Torreón, Coah., México (e-mails: slopezh@lalaguna.tecnm.mx, and mllama@lalaguna.tecnm.mx).

This work was supported by Secihti (Secretaría de Ciencia, Humanidades, Tecnología e Innovación) Projects and TecNM (Tecnológico Nacional de México) Projects, México.

This work addresses the control problem of a 4-WOMR, focusing on the trajectory tracking problem under conditions of parametric uncertainty and a lack of knowledge of mobile robot dynamics. Motivated by recent advances in artificial intelligence, in this work, a two-layer neuro-adaptive compensation control (TLNAC) scheme is proposed to solve the trajectory-tracking problem of a 4-WOMR with Mecanum wheels. To compensate for the unknown dynamics of the 4-WOMR, the two-layer ANN is used, and the weights are updated online using filtered error and suitable adaptive laws. Additionally, real-time experiments are presented to control the speed of the wheels of the Nexus 4-WOMR. One of the main contributions of this work is the use of a novel auxiliary signal to provide robustness relevant to the non-ideal case presented in this scheme. This robust term is necessary to bound the disturbance terms inherent to such control approaches. The dynamic model of the 4-WOMR with Mecanum wheels is provided in Section II. Section III shows the proposed TLNAC scheme design. Experimental results are presented in Section IV. Finally, conclusions and observations are formulated in the Section V, while future work is found in the Section V-A.

II. Nexus 4-WOMR Dynamic Model

This section describes the dynamic model of the Nexus 4-WOMR. The dynamic position and orientation model of a 4-WOMR with Mecanum wheels is given by [17]:

$$\boldsymbol{R}^T(\theta)\boldsymbol{M_R}\ddot{\boldsymbol{\xi}} + \boldsymbol{E}^T\boldsymbol{I}\ddot{\boldsymbol{\varphi}} = \boldsymbol{E}^T\boldsymbol{\tau_\varphi}, \tag{1}$$

where $\boldsymbol{\xi} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ represents the configuration variables in the task space, $\boldsymbol{\varphi} = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 & \varphi_4 \end{bmatrix}^T$ represents the angular position of each wheel,

$$\boldsymbol{\tau_\varphi} = \begin{bmatrix} \tau_{\varphi_1} & \tau_{\varphi_2} & \tau_{\varphi_3} & \tau_{\varphi_4} \end{bmatrix}^T$$

is the vector of torques applied to each wheel,

$$\boldsymbol{M_R} = \text{Diag}\{m_1 + 4m_2, m_1 + 4m_2, 4m_2(l_1^2 + l_2^2) + J_1 + 4J_3\}$$

and $\boldsymbol{I} = \text{Diag}\{J_2, J_2, J_2, J_2\}$ are the inertia matrices of the robot and the wheels, respectively,

$$\boldsymbol{E} = \frac{1}{r}\begin{bmatrix} 1 & 1 & l_1 + l_2 \\ 1 & -1 & -(l_1 + l_2) \\ 1 & 1 & -(l_1 + l_2) \\ 1 & -1 & l_1 + l_2 \end{bmatrix}$$

represents the Jacobian matrix of the system and

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is the rotation matrix for planar motion. Table I shows the kinematic and dynamic parameters of the Nexus 4-WOMR.

The dynamics of the four DC motors with permanent magnets, neglecting the armature inductance L_a and considering a linear friction, is given by [18]:

$$J_m\,\ddot{\boldsymbol{\varphi}} + k_v\dot{\boldsymbol{\varphi}} + \frac{1}{r_e^2}\boldsymbol{\tau_\varphi} + \frac{K_a\,K_b}{R_a}\dot{\boldsymbol{\varphi}} = \frac{K_a}{R_a\,r_e}\boldsymbol{u}, \tag{2}$$

where J_m [kg-m^2] is the inertia of the rotor, k_v [N-m] is the viscous friction, r_e [m] is the transmission ratio, K_a [N-m/A] is a motor-torque constant, K_b [V-s/rad] is the back electromotive force constant, R_a [Ω] is the armature resistance and $\boldsymbol{u} \in \mathbb{R}^4$ [V] is the armature voltage vector.

After performing several algebraic manipulations and taking into account the relationship $\dot{\boldsymbol{\varphi}} = \boldsymbol{E}\boldsymbol{R}(\theta)^T\dot{\boldsymbol{\xi}}$, it can be shown that the dynamics (1), when considering the actuators (2), is expressed as:

$$\boldsymbol{M}\ddot{\boldsymbol{\xi}} + \boldsymbol{C}(\dot{\theta})\dot{\boldsymbol{\xi}} + \boldsymbol{D}\dot{\boldsymbol{\xi}} + \boldsymbol{\delta} = \boldsymbol{\tau}, \tag{3}$$

where $\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T$ is the generalized force vector, $\boldsymbol{\delta} = \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 \end{bmatrix}^T$ is a vector that includes bounded perturbation forces for each configuration variable, $\boldsymbol{M} \in \mathbb{R}^{3\times3}$ is the mass and inertia matrix, $\boldsymbol{C}(\dot{\theta}) \in \mathbb{R}^{3\times3}$ is the centrifugal and Coriolis forces matrix, and $\boldsymbol{D} \in \mathbb{R}^{3\times3}$ is a matrix that includes the friction and parameters of the motors; the matrices $\boldsymbol{M}$, $\boldsymbol{C}(\dot{\theta})$, $\boldsymbol{D}$ and $\boldsymbol{B}$ are given as follows

$$\boldsymbol{M} = \boldsymbol{M_R} + (J_2 + J_m r_e^2)\boldsymbol{E^T E},$$

$$\boldsymbol{C}(\dot{\theta}) = \frac{4}{r^2}(J_2 + J_m r_e^2)\dot{\theta}\boldsymbol{B},$$

$$\boldsymbol{D} = r_e^2\left(\frac{K_a\,K_b}{R_a} + K_v\right)\boldsymbol{E^T E},$$

and

$$\boldsymbol{B} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

It should be noted that $\boldsymbol{M}$ is a constant diagonal matrix. The equation

$$\boldsymbol{\tau} = \frac{K_a\,r_e}{R_a}\boldsymbol{R}(\theta)\boldsymbol{E}^T\boldsymbol{u}$$

relates the generalized forces to the armature voltages of the motors. The 4-WOMR has brushed motors, gear reduction and operates in voltage mode. The angular positions of the wheels are obtained using incremental encoders. This robot is shown in Fig. 1.

III. Control Design using Two-Layer Neuro-Adaptive Compensation Control

In this section, two error-filtered, two-layer neuro-adaptive controllers are designed for position tracking of a 4-WOMR.

TABLE I
KINEMATIC AND DYNAMIC PARAMETERS OF THE NEXUS 4-WORM

Description	Parameter
Radius of the wheels	r [m]
Mass of the body	m_1 [kg]
Mass of each wheel	m_2 [kg]
Inertia of the body	J_1 [kg-m^2]
Inertia of each wheel over the motor shaft	J_2 [kg-m^2]
Inertia of each wheel perpendicular to the motor shaft	J_3 [kg-m^2]
Distance from the geometric center to the shaft of the wheels over R_1	l_1 [m]
Distance from the geometric center to the shaft of the wheels over R_2	l_2 [m]

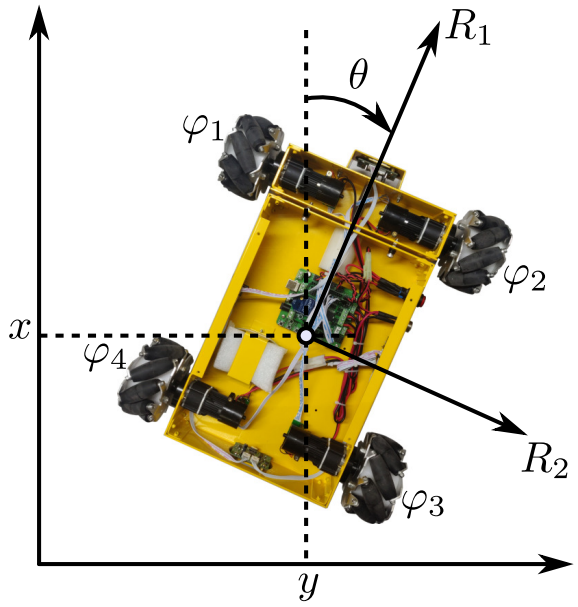

Fig. 1. Diagram of the Nexus 4-Wheeled Omnidirectional Mobile Robot.

An ANN is used to approximate the unknown dynamics of the mobile robot, allowing its weights to be estimated online using appropriate adaptation laws.

A. Problem Statement

The following errors are defined to ensure trajectory tracking

$$\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi_d} - \boldsymbol{\xi}, \quad \dot{\tilde{\boldsymbol{\xi}}} = \dot{\boldsymbol{\xi}}_{\boldsymbol{d}} - \dot{\boldsymbol{\xi}},$$

where $\boldsymbol{\xi_d}, \dot{\boldsymbol{\xi}}_{\boldsymbol{d}} \in \mathbb{R}^n$ are the positions and desired configurations velocities. Besides, filtered tracking error also defined as

$$\boldsymbol{r} = \dot{\tilde{\boldsymbol{\xi}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\xi}}, \tag{4}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ is a diagonal matrix of design parameters, positive definite. Expressing the 4-WOMR dynamic model (3) as a function of $\boldsymbol{r} \in \mathbb{R}^n$ considering that

$$\dot{\boldsymbol{\xi}} = -\boldsymbol{r} + \dot{\boldsymbol{\xi}}_{\boldsymbol{d}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\xi}}, \quad \ddot{\boldsymbol{\xi}} = -\dot{\boldsymbol{r}} + \ddot{\boldsymbol{\xi}}_{\boldsymbol{d}} + \boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{\xi}}},$$

then,

$$\boldsymbol{M}\dot{\boldsymbol{r}} = \boldsymbol{M}(\ddot{\boldsymbol{\xi}}_{\boldsymbol{d}} + \boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{\xi}}}) + \boldsymbol{C}(\dot{\theta})\dot{\boldsymbol{\xi}} + \boldsymbol{D}\dot{\boldsymbol{\xi}} + \boldsymbol{\delta} - \boldsymbol{\tau}. \tag{5}$$

Let us define

$$\boldsymbol{f} \coloneqq \boldsymbol{M}(\ddot{\boldsymbol{\xi}}_{\boldsymbol{d}} + \boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{\xi}}}) + \boldsymbol{C}(\dot{\theta})\dot{\boldsymbol{\xi}} + \boldsymbol{D}\dot{\boldsymbol{\xi}} + \boldsymbol{\varepsilon}, \tag{6}$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the estimation error, which is bounded such that $\| \boldsymbol{\varepsilon} \| < \varepsilon_N$, where the operator $\| \cdot \|$ is the standard Euclidian norm. Then (5) can be expressed as

$$\boldsymbol{M}\dot{\boldsymbol{r}} = \boldsymbol{W^T}\boldsymbol{\Phi}(\boldsymbol{V}^T\boldsymbol{x}) + \boldsymbol{\delta} + \boldsymbol{\varepsilon} - \boldsymbol{\tau},$$

where $\boldsymbol{W} \in \mathbb{R}^{L\times n}$ and $\boldsymbol{V} \in \mathbb{R}^{m\times L}$ are the unknown ideal weights of the ANN, $\boldsymbol{\Phi}(\cdot) \in \mathbb{R}^L$ is the activation function of the ANN and $\boldsymbol{x} \in \mathbb{R}^m$ is the input of the ANN selected heuristically.

Let us group the unknown ideal weights of the ANN as

$$\boldsymbol{Z} \coloneqq \begin{bmatrix} \boldsymbol{W} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V} \end{bmatrix},$$

bounded so that $\| \boldsymbol{Z} \|_F \leq Z_B$, where the operator $\| \cdot \|_F$ is the Frobenius norm. The perturbation forces are bounded so that $\| \boldsymbol{\delta} \| \leq \delta_B$.

Weight estimation errors are expressed as

$$\tilde{\boldsymbol{Z}} = \boldsymbol{Z} - \hat{\boldsymbol{Z}}, \qquad \tilde{\boldsymbol{W}} = \boldsymbol{W} - \hat{\boldsymbol{W}}, \qquad \tilde{\boldsymbol{V}} = \boldsymbol{V} - \hat{\boldsymbol{V}}.$$

The Taylor series approximation of $\boldsymbol{\Phi}(\boldsymbol{V}^T\boldsymbol{x})$ is defined by

$$\boldsymbol{\Phi}(\boldsymbol{V}^T\boldsymbol{x}) = \boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}) + \boldsymbol{\Phi}'(\hat{\boldsymbol{V}}^T\boldsymbol{x})\tilde{\boldsymbol{V}}^T\boldsymbol{x} + \boldsymbol{O}(\tilde{\boldsymbol{V}}^T\boldsymbol{x})^2,$$

where $\boldsymbol{O}(\tilde{\boldsymbol{V}}^T\boldsymbol{x})^2$ is the sum of the higher-order terms in the Taylor series and $\boldsymbol{\Phi}'(\hat{\boldsymbol{V}}^T\boldsymbol{x}) \coloneqq \boldsymbol{\Phi}'$ is the Jacobian of $\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x})$.

B. Neural Network Architecture

The core of the presented approach is an ANN designed to approximate the target function (6). The ANN architecture consists of a two-layer feedforward network, where the output layer employs a linear activation function. In control applications, the activation function in the hidden layer is typically chosen as a hyperbolic tangent. Its operation is defined by the following expression:

$$\hat{\boldsymbol{f}} = \hat{\boldsymbol{W}}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}).$$

The ANN receives an input $\boldsymbol{x}$, which is first transformed by the weight $\hat{\boldsymbol{V}}$. This linear combination is then passed through an element-wise non-linear activation function $\boldsymbol{\Phi}(\cdot)$, in this case, a hyperbolic tangent function, introducing the non-linearity necessary for the network to learn complex patterns. Finally, the activated hidden layer output is multiplied by the output weight $\hat{\boldsymbol{W}}$, and this nonlinear transformation produces the output of the ANN $\hat{\boldsymbol{f}}$.

The parameters of the network, specifically the weights $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{W}}$, are tuned during the on-line training process using a suitable adaptive laws to minimize a defined loss function. This architecture, despite its simplicity, is a powerful universal function approximator capable of modeling a wide range of relationships between inputs and outputs [19].

C. Two-layer neuro-adaptive compensation control (TLNAC)

Inspired by the basis of the two-layer neuro-adaptive control law presented in [2], we propose the following neuro-adaptive control law (see Fig. 2):

$$\boldsymbol{\tau} = \hat{\boldsymbol{W}}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}) + \boldsymbol{K_v}\boldsymbol{r} + \boldsymbol{v}, \tag{7a}$$

$$\dot{\hat{\boldsymbol{W}}} = \boldsymbol{\Gamma}_W\left(\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}) - \boldsymbol{\Phi}'(\hat{\boldsymbol{V}}^T\boldsymbol{x})\hat{\boldsymbol{V}}^T\boldsymbol{x}\right)\boldsymbol{r}^T, \tag{7b}$$

$$\dot{\hat{\boldsymbol{V}}} = \boldsymbol{\Gamma}_V\boldsymbol{x}\boldsymbol{r}^T\hat{\boldsymbol{W}}^T\boldsymbol{\Phi}'(\hat{\boldsymbol{V}}^T\boldsymbol{x}), \tag{7c}$$

with

$$\boldsymbol{v} = K_x\boldsymbol{\alpha}(\boldsymbol{r}) + (\| \hat{\boldsymbol{Z}} \|_F + Z_B)(K_y\boldsymbol{\alpha}(\boldsymbol{r}) + K_z\boldsymbol{r}), \tag{8}$$

$$\boldsymbol{\alpha}(\boldsymbol{r}) = \begin{cases} \dfrac{\boldsymbol{r}}{\| \boldsymbol{r} \|}, & \text{if } \boldsymbol{r} \neq \boldsymbol{0}, \\ \boldsymbol{0}, & \text{if } \boldsymbol{r} = \boldsymbol{0}, \end{cases}$$

where $\boldsymbol{\Gamma}_W \in \mathbb{R}^{L\times L}$ and $\boldsymbol{\Gamma}_V \in \mathbb{R}^{m\times m}$ are tuning rate matrices, positive definite, typically diagonal. $\boldsymbol{v} \in \mathbb{R}^n$ is a support term that provides robustness, and $\boldsymbol{K_v} \in \mathbb{R}^{n\times n}$ is a diagonal, positive definite gain matrix known as velocity gain. Let us define $\boldsymbol{\Lambda} = \boldsymbol{K_v}^{-1}\boldsymbol{K_p}$, where $\boldsymbol{K_p} \in \mathbb{R}^{n\times n}$ is a diagonal, positive definite gain matrix known as position gain, this leads to the expression $\boldsymbol{K_v r}$ being merely a PD.

Next, it is necessary to show how the adaptive laws (7b-7c) dynamically adjust the ANN weight estimates $\hat{\boldsymbol{Z}}$ to ensure stable tracking. This tuning process refines $\hat{\boldsymbol{Z}}$ to approximate the unknown ideal weights $\boldsymbol{Z}$.

The nonlinear function $\boldsymbol{f} \in \mathbb{R}^n$ in (6) is estimated by the ANN

$$\hat{\boldsymbol{f}} = \hat{\boldsymbol{W}}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}),$$

and assuming $\boldsymbol{Z}$ is constant, and letting

$$\tilde{\boldsymbol{f}} = \boldsymbol{f} - \hat{\boldsymbol{f}} = \boldsymbol{W}^T\boldsymbol{\Phi}(\boldsymbol{V}^T\boldsymbol{x}) + \boldsymbol{\varepsilon} - \hat{\boldsymbol{W}}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}).$$

Applying the proposed control law (7a-8) results the following closed-loop dynamics:

$$\boldsymbol{M}\dot{\boldsymbol{r}} = \boldsymbol{W}^T\boldsymbol{\Phi}(\boldsymbol{V}^T\boldsymbol{x}) - \hat{\boldsymbol{W}}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}) - \boldsymbol{K_v r} + \boldsymbol{\varepsilon} + \boldsymbol{\delta} - \boldsymbol{v}.$$

Adding and subtracting $\boldsymbol{W}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x})$, it yields

$$\boldsymbol{M}\dot{\boldsymbol{r}} = \boldsymbol{W}^T\boldsymbol{\Phi}(\tilde{\boldsymbol{V}}^T\boldsymbol{x}) + \tilde{\boldsymbol{W}}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x}) - \boldsymbol{K_v r} + \boldsymbol{\varepsilon} + \boldsymbol{\delta} - \boldsymbol{v}.$$

Adding and subtracting now $\hat{\boldsymbol{W}}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x})$, it yields

$$\begin{aligned}\boldsymbol{M}\dot{\boldsymbol{r}} = \tilde{\boldsymbol{W}}^T\boldsymbol{\Phi}(\tilde{\boldsymbol{V}}^T\boldsymbol{x}) + \hat{\boldsymbol{W}}^T\boldsymbol{\Phi}(\tilde{\boldsymbol{V}}^T\boldsymbol{x}) + \tilde{\boldsymbol{W}}^T\boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x})\\ -\boldsymbol{K_v r} + \boldsymbol{\varepsilon} + \boldsymbol{\delta} - \boldsymbol{v}. \quad (9)\end{aligned}$$

Defining $\hat{\boldsymbol{\Phi}} \coloneqq \boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x})$, then

$$\begin{aligned}\tilde{\boldsymbol{\Phi}} &\coloneqq \boldsymbol{\Phi}(\boldsymbol{V}^T\boldsymbol{x}) - \boldsymbol{\Phi}(\hat{\boldsymbol{V}}^T\boldsymbol{x})\\ &= \boldsymbol{\Phi}'(\hat{\boldsymbol{V}}^T\boldsymbol{x})\tilde{\boldsymbol{V}}^T\boldsymbol{x} + \boldsymbol{O}(\tilde{\boldsymbol{V}}^T\boldsymbol{x})^2\\ &\coloneqq \boldsymbol{\Phi} - \hat{\boldsymbol{\Phi}}.\end{aligned}$$

And (9) can be written as

$$\begin{aligned}\boldsymbol{M}\dot{\boldsymbol{r}} = \tilde{\boldsymbol{W}}^T(\hat{\boldsymbol{\Phi}} - \hat{\boldsymbol{\Phi}}'\hat{\boldsymbol{V}}^T\boldsymbol{x}) + \hat{\boldsymbol{W}}^T\hat{\boldsymbol{\Phi}}'\tilde{\boldsymbol{V}}^T\boldsymbol{x}\\ -\boldsymbol{K_v r} + \boldsymbol{w} - \boldsymbol{v}, \quad (10)\end{aligned}$$

where the disturbance terms are defined as

$$\boldsymbol{w} = \tilde{\boldsymbol{W}}^T\hat{\boldsymbol{\Phi}}'\boldsymbol{V}^T\boldsymbol{x} + \boldsymbol{W}^T\boldsymbol{O}(\tilde{\boldsymbol{V}}^T\boldsymbol{x})^2 + \boldsymbol{\varepsilon} + \boldsymbol{\delta}. \quad (11)$$

Choosing the Lyapunov candidate function as

$$\begin{aligned}U = \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}\boldsymbol{r} + \tilde{\boldsymbol{\xi}}^T\boldsymbol{K_p}\tilde{\boldsymbol{\xi}} + \frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{W}}^T\boldsymbol{\Gamma}_W^{-1}\tilde{\boldsymbol{W}})\\ +\frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{V}}^T\boldsymbol{\Gamma}_V^{-1}\tilde{\boldsymbol{V}}), \quad (12)\end{aligned}$$

where $\mathrm{tr}(\cdot)$ represents the trace trace operator, the time derivative of U can be expressed as

$$\dot{U} = \boldsymbol{r}^T\boldsymbol{M}\dot{\boldsymbol{r}} + 2\tilde{\boldsymbol{\xi}}^T\boldsymbol{K_p}\dot{\tilde{\boldsymbol{\xi}}} + \mathrm{tr}(\tilde{\boldsymbol{W}}^T\boldsymbol{\Gamma}_W^{-1}\dot{\tilde{\boldsymbol{W}}}) + \mathrm{tr}(\tilde{\boldsymbol{V}}^T\boldsymbol{\Gamma}_V^{-1}\dot{\tilde{\boldsymbol{V}}}).$$

Substituting (10) into the above expression, it yields

$$\begin{aligned}\dot{U} = \boldsymbol{r}^T\left(\tilde{\boldsymbol{W}}^T(\hat{\boldsymbol{\Phi}} - \hat{\boldsymbol{\Phi}}'\hat{\boldsymbol{V}}^T\boldsymbol{x}) + \hat{\boldsymbol{W}}^T\hat{\boldsymbol{\Phi}}'\tilde{\boldsymbol{V}}^T\boldsymbol{x} - \boldsymbol{K_v r} + \boldsymbol{w} - \boldsymbol{v}\right)\\ +2\tilde{\boldsymbol{\xi}}^T\boldsymbol{K_p}\dot{\tilde{\boldsymbol{\xi}}} + \mathrm{tr}(\tilde{\boldsymbol{W}}^T\boldsymbol{\Gamma}_W^{-1}\dot{\tilde{\boldsymbol{W}}}) + \mathrm{tr}(\tilde{\boldsymbol{V}}^T\boldsymbol{\Gamma}_V^{-1}\dot{\tilde{\boldsymbol{V}}}),\\ \dot{U} = \boldsymbol{r}^T\left(\tilde{\boldsymbol{W}}^T(\hat{\boldsymbol{\Phi}} - \hat{\boldsymbol{\Phi}}'\hat{\boldsymbol{V}}^T\boldsymbol{x}) + \hat{\boldsymbol{W}}^T\hat{\boldsymbol{\Phi}}'\tilde{\boldsymbol{V}}^T\boldsymbol{x}\right) - \boldsymbol{r}^T\boldsymbol{K_v r}\\ -\boldsymbol{r}^T(\boldsymbol{v} - \boldsymbol{w}) + 2\tilde{\boldsymbol{\xi}}^T\boldsymbol{K_p}\dot{\tilde{\boldsymbol{\xi}}} + \mathrm{tr}(\tilde{\boldsymbol{W}}^T\boldsymbol{\Gamma}_W^{-1}\dot{\tilde{\boldsymbol{W}}}) + \mathrm{tr}(\tilde{\boldsymbol{V}}^T\boldsymbol{\Gamma}_V^{-1}\dot{\tilde{\boldsymbol{V}}}), \quad (13)\end{aligned}$$

then using the property $\boldsymbol{a}^T\boldsymbol{b} = \mathrm{tr}(\boldsymbol{b}\boldsymbol{a}^T)$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ must have compatible dimensions, thus (13) results in

$$\begin{aligned}\dot{U} = -\boldsymbol{r}^T\boldsymbol{K_v r} + 2\tilde{\boldsymbol{\xi}}^T\boldsymbol{K_p}\dot{\tilde{\boldsymbol{\xi}}} - \boldsymbol{r}^T(\boldsymbol{v} - \boldsymbol{w})\\ +\mathrm{tr}\left(\tilde{\boldsymbol{W}}^T\left[\boldsymbol{\Gamma}_W^{-1}\dot{\tilde{\boldsymbol{W}}} + (\hat{\boldsymbol{\Phi}} - \hat{\boldsymbol{\Phi}}'\hat{\boldsymbol{V}}^T\boldsymbol{x})\boldsymbol{r}^T\right]\right)\\ +\mathrm{tr}\left(\tilde{\boldsymbol{V}}^T\left[\boldsymbol{\Gamma}_V^{-1}\dot{\tilde{\boldsymbol{V}}} + \boldsymbol{x}\boldsymbol{r}^T\hat{\boldsymbol{W}}^T\hat{\boldsymbol{\Phi}}'\right]\right).\end{aligned}$$

Now, from the adaptive laws (7b-7c), and since $\dot{\tilde{W}} = -\dot{\hat{W}}$ and $\dot{\tilde{V}} = -\dot{\hat{V}}$, then the time derivative of the Lyapunov function result in

$$\begin{aligned}\dot{U} &= -\boldsymbol{r}^T\boldsymbol{K_v r} + 2\tilde{\boldsymbol{\xi}}^T\boldsymbol{K_p}\dot{\tilde{\boldsymbol{\xi}}} - \boldsymbol{r}^T(\boldsymbol{v} - \boldsymbol{w}),\\ \dot{U} &= -\left[\dot{\tilde{\boldsymbol{\xi}}}^T + \tilde{\boldsymbol{\xi}}^T\boldsymbol{\Lambda}^T\right]\boldsymbol{K_v}\left[\dot{\tilde{\boldsymbol{\xi}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\xi}}\right] + 2\tilde{\boldsymbol{\xi}}^T\boldsymbol{K_p}\dot{\tilde{\boldsymbol{\xi}}} - \boldsymbol{r}^T(\boldsymbol{v} - \boldsymbol{w}),\\ \dot{U} &= -\dot{\tilde{\boldsymbol{\xi}}}^T\boldsymbol{K_v}\dot{\tilde{\boldsymbol{\xi}}} - \tilde{\boldsymbol{\xi}}^T\boldsymbol{\Lambda}^T\boldsymbol{K_v}\boldsymbol{\Lambda}\tilde{\boldsymbol{\xi}} - \dot{\tilde{\boldsymbol{\xi}}}^T\boldsymbol{K_v}\boldsymbol{\Lambda}\tilde{\boldsymbol{\xi}} - \tilde{\boldsymbol{\xi}}^T\boldsymbol{\Lambda}^T\boldsymbol{K_v}\dot{\tilde{\boldsymbol{\xi}}}\\ &\quad +2\tilde{\boldsymbol{\xi}}^T\boldsymbol{K_p}\dot{\tilde{\boldsymbol{\xi}}} - \boldsymbol{r}^T(\boldsymbol{v} - \boldsymbol{w}),\\ \dot{U} &= -\dot{\tilde{\boldsymbol{\xi}}}^T\boldsymbol{K_v}\dot{\tilde{\boldsymbol{\xi}}} - \tilde{\boldsymbol{\xi}}^T\boldsymbol{\Lambda}^T\boldsymbol{K_p}\tilde{\boldsymbol{\xi}} - \boldsymbol{r}^T(\boldsymbol{v} - \boldsymbol{w}). \quad (14)\end{aligned}$$

According to [5], disturbance terms (11) are bounded, so that

$$\| \boldsymbol{w} \| \leq (\varepsilon_N + \delta_B + c_1 Z_B) + c_2 Z_B \| \tilde{\boldsymbol{Z}} \|_F + c_3 Z_B \| \tilde{\boldsymbol{Z}} \|_F \| \boldsymbol{r} \|$$

or

$$\| \boldsymbol{w} \| \leq C_1 + (\| \hat{\boldsymbol{Z}} \|_F + Z_B)(C_2 + C_3 \| \boldsymbol{r} \|) \quad (15)$$

with C_1, C_2 and C_3 positive constants.

Using the inequality (15), the robust term (8) and satisfying $K_x > C_1$, $K_y > C_2$, $K_z > C_3$ in expression (14), we prove that

$$\dot{U} \leq -\begin{bmatrix}\tilde{\boldsymbol{\xi}}\\ \dot{\tilde{\boldsymbol{\xi}}}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{\Lambda}^T\boldsymbol{K_p} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K_v}\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\xi}}\\ \dot{\tilde{\boldsymbol{\xi}}}\end{bmatrix} \leq 0, \quad (16)$$

$$\dot{U} \leq -\begin{bmatrix}\tilde{\boldsymbol{\xi}}\\ \dot{\tilde{\boldsymbol{\xi}}}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{K_p}\,\boldsymbol{K_v}^{-1}\boldsymbol{K_p} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K_v}\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\xi}}\\ \dot{\tilde{\boldsymbol{\xi}}}\end{bmatrix} \leq 0. \quad (17)$$

Since $\boldsymbol{\Lambda}$ is non-singular and $\boldsymbol{K_v}$ is symmetric positive definite, $\boldsymbol{\Lambda}^T\boldsymbol{K_p} = \boldsymbol{K_p}\,\boldsymbol{K_v}^{-1}\boldsymbol{K_p}$ is also symmetric positive definite. This ensures that U in (16) is negative semi-definite. Given that the Lyapunov candidate U in (12) is positive definite, radially unbounded, and decreasing, the origin $\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}} = \boldsymbol{0}, \tilde{\boldsymbol{Z}} = \boldsymbol{0}$ is uniformly stable, and $\boldsymbol{r}$, $\tilde{\boldsymbol{\xi}}$, $\dot{\tilde{\boldsymbol{\xi}}}$ and $\tilde{\boldsymbol{Z}}$ are bounded for all initial conditions. Integrating both sides of (17), we show that

$$\begin{aligned}\int_0^t \tilde{\boldsymbol{\xi}}(\sigma)^T\tilde{\boldsymbol{\xi}}(\sigma)\,\mathrm{d}\sigma &\leq \frac{U(0, \boldsymbol{r}(0), \tilde{\boldsymbol{\xi}}(0), \tilde{\boldsymbol{Z}}(0))}{\lambda_{\min}\{\boldsymbol{K_p}\,\boldsymbol{K_v}^{-1}\boldsymbol{K_p}\}}, \text{and}\\ \int_0^t \dot{\tilde{\boldsymbol{\xi}}}(\sigma)^T\dot{\tilde{\boldsymbol{\xi}}}(\sigma)\,\mathrm{d}\sigma &\leq \frac{U(0, \boldsymbol{r}(0), \tilde{\boldsymbol{\xi}}(0), \tilde{\boldsymbol{Z}}(0))}{\lambda_{\min}\{\boldsymbol{K_v}\}},\end{aligned}$$

which signifies that $\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}} \in \mathrm{L}_2^n$, consequently, application of Barbalat's Lemma [20] leads to the conclusion that

$$\lim_{t\to\infty}\tilde{\boldsymbol{\xi}}(t) = \boldsymbol{0}, \quad \lim_{t\to\infty}\dot{\tilde{\boldsymbol{\xi}}}(t) = \boldsymbol{0} \quad \text{and} \quad \lim_{t\to\infty}\boldsymbol{r}(t) = \boldsymbol{0}.$$

This stability analysis diverges from that presented in [2] in several key aspects. A different Lyapunov candidate function

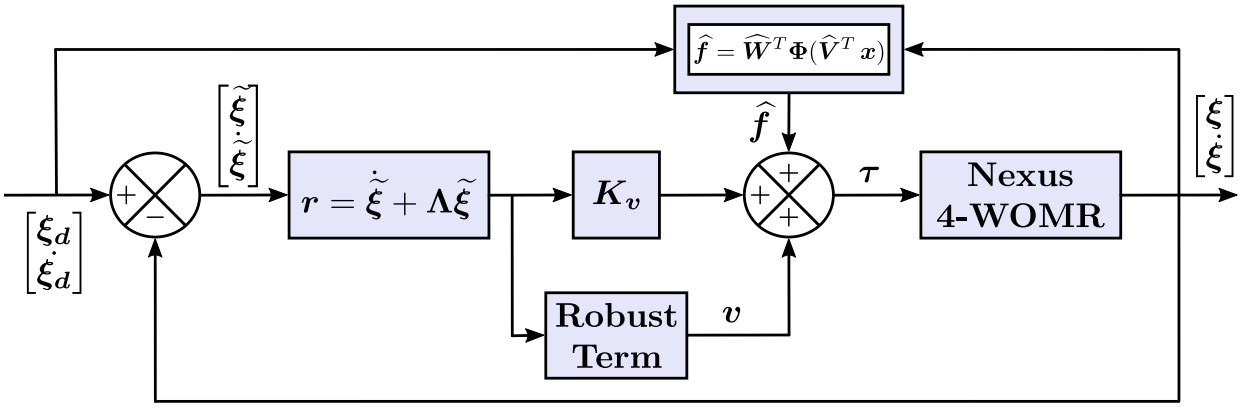

Fig. 2. Block diagram of the TLNAC.

is used, allowing for direct analysis of $\tilde{\boldsymbol{\xi}}$ and $\dot{\tilde{\boldsymbol{\xi}}}$. Moreover, we explicitly consider $\boldsymbol{\varepsilon}$, $\boldsymbol{\delta}$, and higher-order Taylor series terms, and utilize a distinct robust term $\boldsymbol{v}$.

A novel suitable robust term $\boldsymbol{v}$ (8) is proposed to correctly counterbalance the disturbance terms $\boldsymbol{w}$ (15). Note the importance of the robust term $\boldsymbol{v}$ (8) proposed here, since in the state of the art corresponding to TLNAC, it is usually assumed that the disturbance terms $\boldsymbol{w}$ (15) have a constant bound.

IV. TLNAC Real-Time Implementation

In this section, the TLNAC scheme is real-time implemented in the Nexus 4-WOMR. The desired wheel trajectories are determined via inverse robot kinematics, and then the wheel speeds are controlled (see Fig. 3). This controller has a slightly higher computational cost than the scheme proposed in [2], but it is low enough to run on the Nexus 4-WOMR microcontroller, an ATmega328p.

Since the controller is implemented to control wheel speeds, an adaptation to the controller is necessary. In order to achieve speed tracking, the following errors are defined

$$\tilde{\varphi}_i = \varphi_{di} - \varphi_i, \quad \dot{\tilde{\varphi}}_i = \dot{\varphi}_{di} - \dot{\varphi}_i$$

for every i-th wheel, with $i = 1, 2, 3, 4$. In addition, filtered error is also defined as

$$r_i = \dot{\tilde{\varphi}}_i + \Lambda \tilde{\varphi}_i.$$

Thus, the control law results in

$$u_i = \hat{\boldsymbol{W}}_i^T \boldsymbol{\Phi}(\hat{\boldsymbol{V}}_i^T \boldsymbol{x}_i) + K_v r_i,$$
$$\dot{\hat{\boldsymbol{W}}}_i = \Gamma_W \left(\boldsymbol{\Phi}(\hat{\boldsymbol{V}}_i^T \boldsymbol{x}_i) - \boldsymbol{\Phi}'(\hat{\boldsymbol{V}}_i^T \boldsymbol{x}_i) \hat{\boldsymbol{V}}_i^T \boldsymbol{x}_i \right) r_i,$$
$$\dot{\hat{\boldsymbol{V}}}_i = \Gamma_V r_i \boldsymbol{x}_i \hat{\boldsymbol{W}}_i^T \boldsymbol{\Phi}'(\hat{\boldsymbol{V}}_i^T \boldsymbol{x}_i).$$

The following desired trajectories were selected:

$$\boldsymbol{q_d} = \begin{bmatrix} \text{sen}\,(\omega\, t) \\ \cos\,(\omega\, t) \\ -\omega\, t \end{bmatrix} \begin{matrix} \text{[m]} \\ \text{[m]} \\ \text{[rad]} \end{matrix} \tag{18}$$

where $\omega = 2\pi/15$ is the frequency of the circle and rotation of the robot. The initial conditions where $\boldsymbol{\xi}(0) = [0, 1, 0]^T$, $\hat{\boldsymbol{W}}_i(0) = \boldsymbol{0}_{2\times 1}$ and $\hat{\boldsymbol{V}}_i(0)$ small random values.

The controller tuning technique was described in [21]. The gains resulting from the tuning procedure are given as follows:

$$K_v = 25 \text{ [V s/rad]}, \quad K_p = 150 \text{ [V/rad]},$$
$$\Lambda = 6 \text{ [1/s]}, \quad \Gamma_W = 1500 \quad \text{and} \quad \Gamma_V = 1500.$$

The input vector $\boldsymbol{x}_i$, which is fed into the ANN, is given by

$$\boldsymbol{x}_i \coloneqq \begin{bmatrix} \tilde{\varphi}_i & \dot{\tilde{\varphi}}_i & 1 \end{bmatrix}^T,$$

and the activation function is given by

$$\boldsymbol{\Phi}(\boldsymbol{x_i}) = \begin{bmatrix} \tanh(x_{i1}) & \tanh(x_{i2}) & \tanh(x_{i3}) \end{bmatrix}^T,$$

where $\tanh(\cdot)$ is the hyperbolic tangent function. To measure the performance of the controller the RMS of the configuration position errors $\tilde{\boldsymbol{\xi}}$ is considered, this value is computed according to

$$\tilde{\xi}_{iRMS} = \sqrt{\frac{1}{\Delta T} \int_t^{t+\Delta T} \tilde{\xi}_i(\sigma)^2 d\sigma},$$

with $i = x,\, y,\, \theta$.

The following figures show the experimental results of the wheel speeds control. Figs. 4-5 show the time evolution of positions $\boldsymbol{\xi}$ and the reference trajectories $\boldsymbol{\xi_d}$ for the TLNAC scheme. Figs. 6-7 show the speed errors $\tilde{\boldsymbol{\varphi}}$ and the voltages $\boldsymbol{u}$ for the TLNAC scheme obtained experimentally. The Table II shows the RMS values of the configuration position errors $\tilde{\boldsymbol{\xi}}$. The values are given in centimeters (cm) and radians (rad), respectively.

Note that when implementing wheel speed control, no direct measurement of the configuration variables $\boldsymbol{\xi}$ is made; therefore, the Figs. 4-5 are obtained through the robot kinematics.

TABLE II
RMS Values of the Position Errors. RA-C4

Variable	RMS value	Units
$\tilde{\xi}_{xRMS}$	0.0177	cm
$\tilde{\xi}_{yRMS}$	0.0015	cm
$\tilde{\xi}_{\theta RMS}$	0.0774	rad

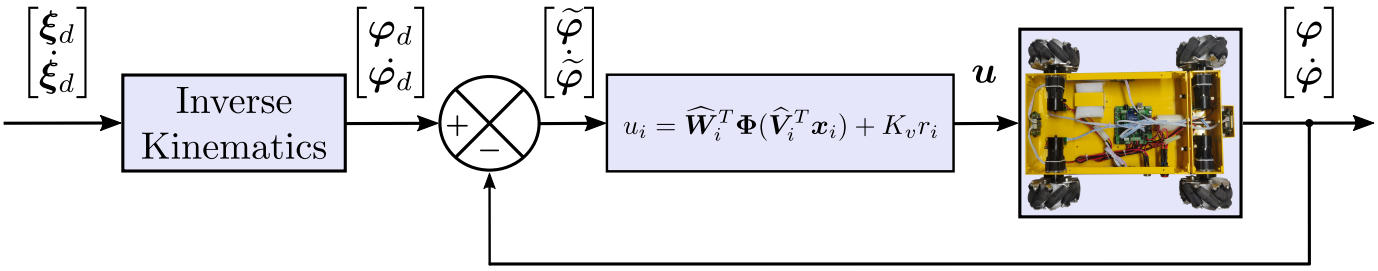

Fig. 3. Block diagram of real-time speed control of the 4-WOMR using the TLNAC scheme.

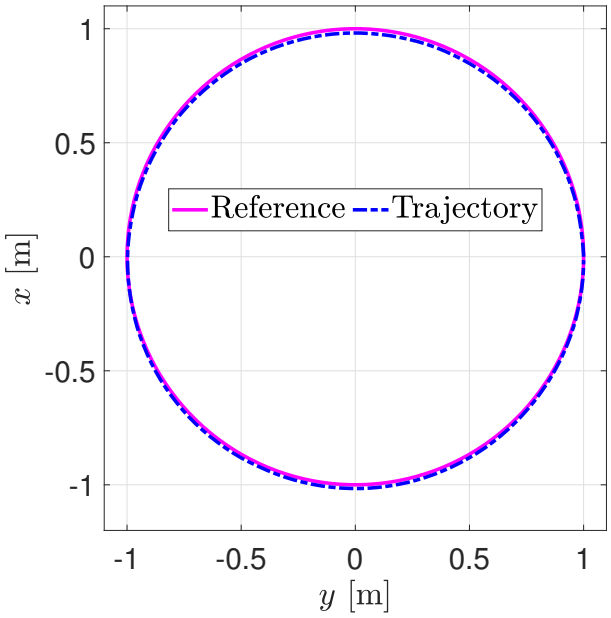

Fig. 4. Position system response for the TLNAC scheme in real-time.

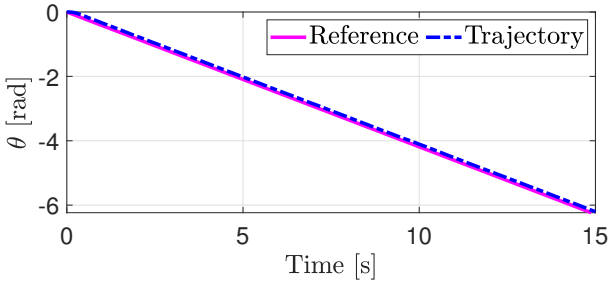

Fig. 5. Orientation system response for the TLNAC scheme in real-time.

V. Conclusions

In this work, a neuro-adaptive control scheme with a novel auxiliary signal to provide robustness is proposed to control the Nexus 4-WOMR. The two-layer ANN is used to compensate for the 4-WOMR unknown dynamics, and the filtered error technique is used to update the weights of the ANN. Experiments on the Nexus 4-WOMR showed that the TLNAC scheme is good enough to compensate for the different friction coefficients of each wheel without saturating the motor voltages. In addition, it should be mentioned that to increase wheel traction, sheets of neoprene are used between the robot and the floor.

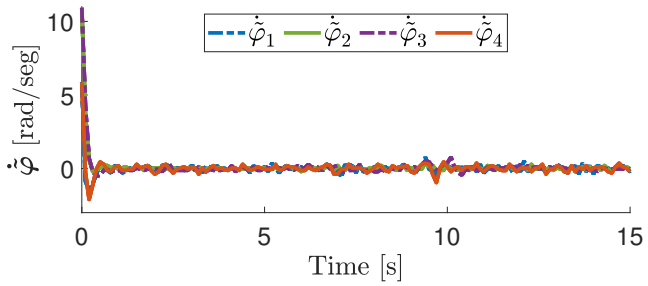

Fig. 6. Motor speed errors for the TLNAC scheme in real-time.

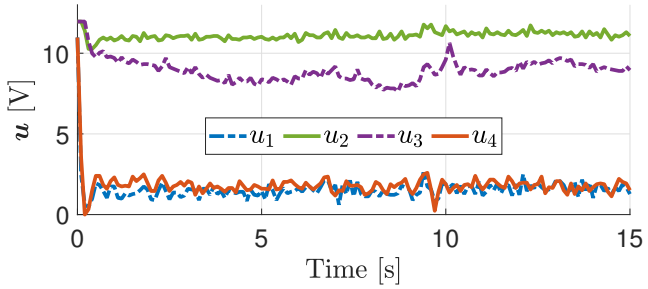

Fig. 7. Motor voltage signals recorded in real-time under the TLNAC scheme.

One of the main contributions of this work is the use of a novel auxiliary signal to provide robustness v (8) relevant to the non-ideal case presented scheme, this robust term is necessary to bound the disturbance terms w (15). Note the importance of the robust term v (8), since in the state of the art corresponding to TLNAC, it is usually assumed that the disturbance terms w (15) have a constant bound.

In summary, it was possible to design a two-layer neural controller incorporating a novel robust term, v (8). This controller successfully achieved good performance without excessive control signal noise and effectively compensated for friction and uncertain system parameters.

A. Future Work

The goal of future work is to test the effectiveness of the TLNAC scheme experimentally, but now with a platform that allows measuring the operational coordinates of the 4-WOMR Nexus.

Disclosure 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] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7436–7456, 2022. DOI: 10.1109/TPAMI.2021.3117837.

[2] F. L. Lewis, A. Yesildirak, and S. Jagannathan, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Bristol, PA, USA: Taylor & Francis, Inc., 1998.

[3] J. Farrell and M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control, Wiley, 2006. DOI: 10.1002/0471781819.

[4] Z. Hendzel *et al.*, "Modelling of dynamics of a wheeled mobile robot with mecanum wheels with the use of Lagrange equations of the second kind," *International Journal of Applied Mechanics and Engineering*, vol. 22, no. 1, pp. 81–99, 2017. DOI: 10.1515/ijame-2017-0005.

[5] F. Lewis, "Nonlinear network structures for feedback control," *Asian Journal of Control*, vol. 1, no. 4, pp. 205–228, 1999. DOI: 10.1111/j.1934-6093.1999.tb00021.x.

[6] H. Wang, Y. Yang, J. Fei, and Y. Fang, "Adaptive control of micro-electro-mechanical system gyroscope using neural network compensator," *Advances in Mechanical Engineering*, vol. 11, no. 12, p. 1687814019898325, 2019. DOI: 10.1177/1687814019898325.

[7] F. Luan, J. Na, Y. Huang, and G. Gao, "Adaptive neural network control for robotic manipulators with guaranteed finite-time convergence," *Neurocomputing*, vol. 337, pp. 153 – 164, 2019. DOI: 10.1016/j.neucom.2019.01.063.

[8] Y. Wu, R. Huang, X. Li, and S. Liu, "Adaptive neural network control of uncertain robotic manipulators with external disturbance and time-varying output constraints," *Neurocomputing*, vol. 323, pp. 108 – 116, 2019. DOI: 10.1016/j.neucom.2018.09.072.

[9] F. Baghbani, M.-R. Akbarzadeh-T, M.-B. Naghibi-Sistani, and A. Akbarzadeh, "Emotional neural networks with universal approximation property for stable direct adaptive nonlinear control systems," *Engineering Applications of Artificial Intelligence*, vol. 89, p. 103447, 2020. DOI: 10.1016/j.engappai.2019.103447.

[10] X. Wang, X. Li, Q. Wu, and X. Yin, "Neural network based adaptive dynamic surface control of nonaffine nonlinear systems with time delay and input hysteresis nonlinearities," *Neurocomputing*, vol. 333, pp. 53 – 63, 2019. DOI: 10.1016/j.neucom.2018.12.058.

[11] V. T. Yen, W. Y. Nan, and P. Van Cuong, "Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators," *Neural Computing and Applications*, vol. 31, no. 11, pp. 6945–6958, 2019. DOI: 10.1007/s00521-018-3520-3.

[12] J. Moreno–Valenzuela, J. Montoya–Cháirez, and V. Santibáñez, "Robust trajectory tracking control of an underactuated control moment gyroscope via neural network–based feedback linearization," *Neurocomputing*, vol. 403, pp. 314 – 324, 2020. DOI: 10.1016/j.neucom.2020.04.019.

[13] D. Xu, D. Zhao, J. Yi, and X. Tan, "Trajectory tracking control of omnidirectional wheeled mobile manipulators: robust neural network-based sliding mode approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 3, pp. 788–799, 2009. DOI: 10.1109/TSMCB.2008.2009464.

[14] X. Lu, X. Zhang, G. Zhang, J. Fan, and S. Jia, "Neural network adaptive sliding mode control for omnidirectional vehicle with uncertainties," *ISA transactions*, vol. 86, pp. 201–214, 2019. DOI: 10.1016/j.isatra.2018.10.043.

[15] T. Zhao, P. Qin, and Y. Zhong, "Trajectory tracking control method for omnidirectional mobile robot based on self-organizing fuzzy neural network and preview strategy," *Entropy*, vol. 25, no. 2, 2023. DOI: 10.3390/e25020248.

[16] P. Qin, T. Zhao, N. Liu, Z. Mei, and W. Yan, "Predefined-time fuzzy neural network control for omnidirectional mobile robot," *Processes*, vol. 11, no. 1, 2023. DOI: 10.3390/pr11010023.

[17] G. Campion, G. Bastin, and B. Dandrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 47–62, Feb 1996. DOI: 10.1109/70.481750.

[18] R. Kelly and V. Santibáñez, *Control de movimiento de robots manipuladores*. Automática y Robótica, Pearson Educación, 2003.

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation*. International edition, Prentice Hall, 1999.

[20] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

[21] S. López, M. Llama, R. García-Hernández, and V. Santibáñez, "Pd with neuro-adaptive compensation control using the signed power function," *International Journal of Control*, vol. 96, no. 6, pp. 1638–1649, 2022. DOI: 10.1080/00207179.2022.2062452.

Sergio López Hernández received his B.Sc. degree in Electronic Engineering with a specialization in Instrumentation and Control in 2014, and the M.Sc. and Ph.D. degrees in Electric Engineering with a specialization in Mechatronics and Control in 2017 and 2024, respectively, all from the Tecnológico Nacional de México / Instituto Tecnológico de La Laguna. He is affiliated with the Instituto Tecnológico de La Laguna. His research interests include intelligent control, deep learning, neural networks, adaptive control, autonomous driving, and fuzzy control. He is a member of the IEEE and the National System of Researchers (SNII) in Mexico.

Miguel A. Llama received his B.S. degree in electronics & communications engineering from ITESM Campus Monterrey, México, in 1977; the M.Sc. degree in electrical engineering from The University of Texas at Austin, in 1981 and, the Ph.D. degree in electrical engineering from Tecnológico Nacional de México/I. T. La Laguna (ITL) at Torreón, Coahuila, México, in 2001. He has been with ITL as a research professor since 1982. His research interest is in the area of intelligent systems and fuzzy logic applied to robot control. He is member of the Research National System (SNII) of Mexico and IEEE Senior Member.