



Accurate trajectory tracking control with adaptive neural networks for omnidirectional mobile robots subject to unmodeled dynamics

Gabriel da Silva Lima¹ · Victor Ramon Firmo Moreira¹ · Wallace Moreira Bessa² 

Received: 22 June 2022 / Accepted: 12 December 2022
© The Author(s) 2022

Abstract

Omnidirectional mobile robots have gained a lot of attention in recent years due to their maneuverability capabilities. However, ensuring accurate trajectory tracking with this class of robots is still challenging control system designers. In this work, a novel intelligent controller is introduced for accurate trajectory tracking of omnidirectional robots subject to unstructured uncertainties. An adaptive neural network is adopted within a Lyapunov-based nonlinear control scheme to deal with frictional forces and other unmodeled dynamics or external disturbances that may occur. Online learning, rather than supervised offline training, is employed to allow the robot to learn on its own how to compensate for uncertainties and disturbances by interacting with the environment. The adoption of a combined error signal as the single input in the neural network significantly reduces the computational complexity of the disturbance compensation scheme and enables the resulting intelligent controller to be implemented in the embedded hardware of mobile robots. The boundedness and convergence properties of the proposed control scheme are proved by means of a Lyapunov-like stability analysis. The effectiveness of the proposed intelligent controller is numerically evaluated and experimentally validated using an omnidirectional mobile robot. The comparative analyses of the obtained results show that the adoption of an intelligent compensation scheme based on adaptive neural networks allows reductions of more than 95% in the tracking error, thus guaranteeing an accurate tracking and confirming the great superiority of the proposed control strategy.

Keywords Adaptive algorithms · Intelligent control · Mobile robots · Radial basis function networks

1 Introduction

Mobile robots are autonomous devices that are able to move themselves and interact with the environment [1, 2]. It is noteworthy that mobile robotics is a cutting-edge research field and one of the most rapidly growing technologies at the

moment, with applications ranging from warehouse logistics and surveillance operations to healthcare services and planetary exploration [3–5]. In addition, a special class of mobile robots, namely mecanum-wheeled omnidirectional robots, has gained a lot of attention lately due to their great maneuverability [6]. Independently driven mecanum wheels allow robots to move in an arbitrary direction with simultaneous translation and rotation, which grants them the ability to perform complex maneuvers [7] and navigate in narrow spaces [8, 9]. However, despite all these advantages, mecanum wheels can also lead to imprecise motion caused by wobbly contact between the free rollers with the ground [3, 8], which makes the design of accurate controllers for trajectory tracking with omnidirectional mobile robots (OMR) an urgent and imperative requirement [9, 10].

Although conventional controllers such as computed torque and other feedback linearization-based approaches have been successfully employed in industrial robotics [11], this is usually not the most suitable choice for trajectory tracking problems in mobile robotics. Mobile robots

Technical Editor: Rogério Sales Gonçalves.

✉ Wallace Moreira Bessa
wmobes@utu.fi

Gabriel da Silva Lima
gabriel.lima.095@ufrn.edu.br

Victor Ramon Firmo Moreira
victor.moreira@ufrn.br

¹ Universidade Federal do Rio Grande do Norte, Campus Universitário Lagoa Nova, Natal, RN 59078-970, Brazil

² Department of Mechanical and Materials Engineering, University of Turku, Joulukahaiskatu 3, 20520 Turku, Finland

are often subject to unstructured environments, modeling inaccuracies and external disturbances, which in fact pose major challenges for the design of classical control laws [12, 13]. Furthermore, special attention should be also paid to the effects of friction on trajectory tracking with omnidirectional robots [14–16]. The contact of a mecanum wheel with the floor is theoretically limited to only one point, which causes the robot to slide and change its position/orientation abruptly even when subjected to only small changes in the friction force [16]. Thus, it is essential that the adopted control approach is also capable of dealing with the uncertainties arising from friction [9, 14].

In order to address the issues of modeling inaccuracies and imperfect information, computational intelligence offers a very attractive option for the control of mobile robots. On this basis, both fuzzy logic [8–10, 16–19] and artificial neural networks [20–25] based schemes have been adopted in the control of omnidirectional mobile robots. Cao et al. [8], for example, present a fuzzy adaptive PID controller for an OMR with eight mecanum wheels. In [9], backstepping is combined with fuzzy approximators to compensate for friction effects as well as other modeling inaccuracies, and the effectiveness of the proposed scheme is evaluated through numerical simulations considering a four-wheeled OMR. Zou et al. [10] propose a type 2 adaptive range fuzzy controller for a four-wheeled OMR, and its performance is compared with a PID scheme by means of computer simulations. A fuzzy proportional-integral (PI) controller is presented in [16] to improve the heading accuracy of a four-wheeled OMR. In [17], Mamdani hierarchical fuzzy systems are employed to allow a three-wheeled OMR to navigate in an unstructured environment. Trajectory tracking with three-wheeled OMR is also addressed in [18], but using an adaptive fuzzy sliding mode controller. Abiyev et al. [19] introduce a Z-number-based fuzzy inference system for the control of omnidirectional soccer robots.

Due to their universal approximation capability [26], radial basis function (RBF) networks are also widely employed for trajectory control with OMR. Zijie et al. [20], for instance, combine RBF networks with PID to enhance the heading precision of a four-wheeled OMR. In [21], RBF networks are used to estimate uncertain nonlinear functions in a sliding mode scheme, with the control performance being evaluated through numerical simulations. RBF networks have been also combined with Dynamic surface control to handle the trajectory tracking problem with four-wheeled [22, 23] and three-wheeled [24] OMR.

Despite the ability of control schemes based on computational intelligence to compensate for uncertainties and disturbances, there are some fundamental requirements that are often overlooked and not fulfilled:

1. The control scheme must be able to handle the robot dynamics, rather than just taking its kinematics into account, if accurate trajectory tracking is desired. Dynamic effects are the main source of uncertainty in mobile robotics and, if neglected, usually lead to degraded control performance.
2. Online learning, rather than offline training, should be used to allow the mobile robot to continuously learn by interacting with its surroundings. This is actually a very important feature because the control scheme must be able to continuously improve its overall performance even when subjected to a dynamically changing environment.
3. The intelligent algorithm should learn to deal with uncertainties and disturbances indirectly, as they are usually not available to be measured. The compensation scheme, in this case, should not rely on direct measurements of the signal to be estimated.
4. The resulting intelligent controller must be simple enough to avoid computational complexity, so that it can be implemented on the low-power embedded hardware of mobile robots. The computational complexity of neural networks and fuzzy systems is directly related to the number of neurons in the input layer or fuzzy variables in the premise of the rules, respectively.

As a matter of fact, none of the aforementioned control schemes satisfies all these four essential conditions.

In this work, a novel multivariable intelligent controller is introduced for three-wheeled omnidirectional mobile robots that is able to successfully handle these four requirements. The proposed scheme employs an adaptive radial basis function neural network to represent the robot uncertain dynamics within a Lyapunov-based nonlinear control law. In order to keep computational complexity low, the architecture adopted for the network requires only a combined error measure as the single input neuron. Moreover, the weight vector of the neural net can be updated online by minimizing the combined error measure, which allows the controller to be continuously improved as the robot moves around and does not need direct measurements of the disturbance that must be compensated. In fact, the proposed approach can be understood as an extension for a multiple-input and multiple-output system of the control strategy presented by Dos Santos and Bessa [27] for electrohydraulic actuators and by Bessa et al. [28] for a micro-diving cell. The boundedness and convergence properties of the tracking error are rigorously proved by means of a Lyapunov-like stability analysis. In addition, the effectiveness of the intelligent controller is evaluated not only via computer simulations, but also through experiments performed with an omnidirectional mobile robot. The superiority of the proposed control strategy can be clearly seen through the obtained results: the adoption of an intelligent compensation scheme guarantees a

reduction of tracking error by more than 95%. It should also be highlighted that the proposed control approach presents a good trade-off between simplicity and efficacy, which allows its implementation in the low-power embedded hardware normally used in mobile robotics.

2 Omnidirectional mobile robot

An omnidirectional mobile robot like Robotino®, Fig. 1, has its individually controllable wheels arranged in such a manner that it can perform both translation and rotation motions independently.

In order to start designing the control law, a mathematical model representing the omnidirectional robot is quite useful. Dynamic models for such systems are well documented in the literature [24, 29–31]. In this section, we first present the kinematic relationship between the angular velocities of the wheels and the velocities described in the inertial frame. Then, the equation of motion of the system is obtained by means of Lagrangian mechanics.

2.1 Robot kinematics

A schematic representation of the mobile robot with the adopted coordinate systems is shown in Fig. 2.

Considering that each wheel touches the ground at a single point and rolls without slipping, the velocity of the center point can be computed by:

$$B_i \dot{r}_{w,i} = B_i \dot{r}_{p,i} + B_i \dot{\theta}_i \times B_i r_w \tag{1}$$

where $B_i \dot{\theta}_i = [\omega_i \ 0 \ 0]^T$, with ω_i being the angular velocity of the wheel, $B_i r_w = [0 \ 0 \ r]^T$ the vector with wheel radius

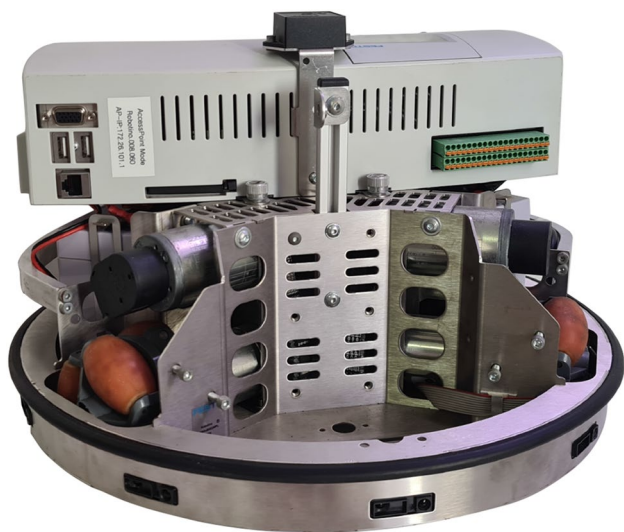


Fig. 1 Three-wheeled omnidirectional mobile robot

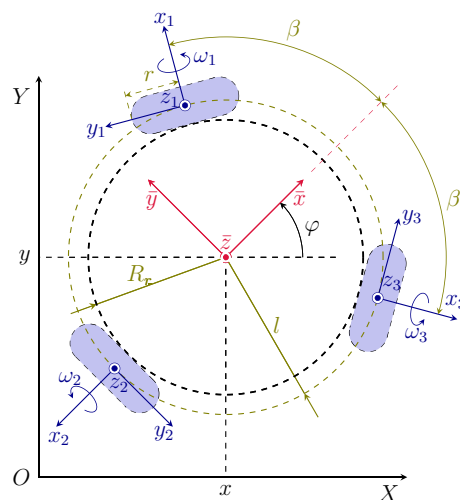


Fig. 2 Schematic top view of the omnidirectional robot

r , and $B_i \dot{r}_{p,i} = [\dot{r}_i \ 0 \ 0]^T$ standing for the velocity due to the other two wheels.

Equation (1) can be also represented in the inertial frame using the appropriate transformation matrices:

$$I \dot{r}_{w,i} = T_\varphi T_i B_i \dot{r}_{w,i} \tag{2}$$

where T_φ is the transformation matrix between the inertial and the robot's fixed reference frames, $I r = T_\varphi \bar{B} r$:

$$T_\varphi = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

and T_i is the transformation matrix relating the center of the i th wheel to the robot's center of mass $\bar{x}\bar{y}$, $\bar{B} r = T_{iB_i} r$:

$$T_1 = T_3^T = \begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4a}$$

$$T_2 = \text{diag} \{-1, -1, 1\} \tag{4b}$$

with β being the angle between the wheel axes, as depicted in Fig. 2.

Now, the velocity of each wheel is obtained by combining the translational velocity of the robot $I \dot{r}_t$ with its rotational velocity $I \dot{r}_r$:

$$I \dot{r}_{w,i} = I \dot{r}_t + I \dot{r}_r \tag{5}$$

where

$$I \dot{r}_t = [\dot{x} \ \dot{y} \ 0]^T \tag{6}$$

and

$${}_I\dot{\mathbf{r}}_r = {}_I\dot{\boldsymbol{\varphi}} \times (T_{\varphi\bar{B}}\mathbf{r}_i) \tag{7}$$

with ${}_I\dot{\boldsymbol{\varphi}}$ being the resulting angular velocity of the robot:

$${}_I\dot{\boldsymbol{\varphi}} = \begin{bmatrix} 0 \\ 0 \\ \dot{\varphi} \end{bmatrix} + T_{\varphi} \sum_{i=1}^3 T_{iB_i} \dot{\boldsymbol{\theta}}_i \tag{8}$$

and $\bar{B}\mathbf{r}_i$ being the center of rotation of each wheel with respect to the center of the robot:

$$\bar{B}\mathbf{r}_i = T_i \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix} \tag{9}$$

By combining Eqs. (2) and (5), the angular velocities of the wheels and the speed of the robot, $\dot{\mathbf{q}} = [\dot{x} \ \dot{y} \ \dot{\varphi}]^T$, can be related as follows:

$$\dot{\mathbf{q}} = T_r \boldsymbol{\omega} \tag{10}$$

with $T_r \in \mathbb{R}^{3 \times 3}$ being the corresponding transformation matrix [30], as detailed in Appendix A, and $\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \omega_3]^T$.

2.2 Robot dynamics

In order to derive the equations of motion for the omnidirectional robot using the Lagrangian approach, the generalized coordinates of the system need to be defined first:

$$\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3]^T \tag{11}$$

with $\dot{\boldsymbol{\theta}} = \boldsymbol{\omega}$.

In addition to uniquely describing the robot’s position, the chosen coordinates are directly related to the input torques $\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \tau_3]^T$.

In this work it is considered that the robot moves freely on a flat surface. Thus, the equations of motion can be obtained from the Euler–Lagrange equation:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{K}(\boldsymbol{\theta}, \boldsymbol{\omega})}{\partial \boldsymbol{\omega}} \right) - \frac{\partial \mathcal{K}(\boldsymbol{\theta}, \boldsymbol{\omega})}{\partial \boldsymbol{\theta}} = \boldsymbol{\tau} \tag{12}$$

with $\mathcal{K}(\boldsymbol{\theta}, \boldsymbol{\omega})$ being the kinetic energy.

Basically, a three-wheeled mobile robot is composed by four rigid bodies: a main body and the three driving blocks. However, as the controller to be proposed in the next section is capable of handling unmodeled dynamics, only the main body of the robot will be considered in kinetic energy:

$$\mathcal{K}(\boldsymbol{\theta}, \boldsymbol{\omega}) = \frac{1}{2} m_r \dot{\mathbf{r}}_t^T \dot{\mathbf{r}}_t + \frac{1}{2} {}_I\dot{\boldsymbol{\varphi}}^T {}_I\mathbf{I}_r {}_I\dot{\boldsymbol{\varphi}} \tag{13}$$

where ${}_I\dot{\boldsymbol{\varphi}} = [0 \ 0 \ \dot{\varphi}]^T$ and ${}_I\mathbf{I}_r$ is the inertia matrix of the main body:

$${}_I\mathbf{I}_r = m_r \text{diag} \left\{ \frac{h_r^2 + 3R_r^2}{12}, \frac{h_r^2 + 3R_r^2}{12}, \frac{R_r^2}{2} \right\} \tag{14}$$

with m_r , h_r , and R_r being the mass, height, and radius of the robot, respectively.

So, applying (6), (7), and (14) into (13), it follows that:

$$\mathcal{K}(\boldsymbol{\theta}, \boldsymbol{\omega}) = \frac{1}{2} m_r \left(\dot{x}^2 + \dot{y}^2 + \frac{1}{2} R_r^2 \dot{\varphi}^2 \right) \tag{15}$$

Therefore, using the transformation matrix (10) to replace the variables \dot{x} , \dot{y} , $\dot{\varphi}$, the equation of motion for a omnidirectional mobile robot becomes [30, 31]:

$$\mathbf{M}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} \tag{16}$$

where $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ is the inertia matrix, which is detailed in Appendix B.

Finally, by combining the time derivative of (10) with (16), the robot dynamics can be expressed in the following vector form:

$$\ddot{\mathbf{q}} = \mathbf{f} + T_r \mathbf{M}^{-1} \boldsymbol{\tau} + \mathbf{d} \tag{17}$$

where $\mathbf{f} = \dot{T}_r T_r^{-1} \dot{\mathbf{q}}$, and \mathbf{d} is introduced to represent not only external disturbances but also modeling inaccuracies.

3 Intelligent controller

Intelligent controllers are able to adapt themselves, learn by interacting with the environment and can fairly predict the outcome of their own actions. They have been successfully used to control robotic [28, 32], underactuated [33–35], chaotic [36, 37] and other uncertain nonlinear systems [27, 38].

First, following the feedback linearization method [39], the control law for the omnidirectional mobile robot is defined as follows:

$$\boldsymbol{\tau} = \mathbf{M} T_r^{-1} (-\mathbf{f} - \hat{\mathbf{d}} + \ddot{\mathbf{q}}_d - 2\boldsymbol{\Lambda}\dot{\tilde{\mathbf{q}}} - \boldsymbol{\Lambda}^2\tilde{\mathbf{q}}) \tag{18}$$

with $\hat{\mathbf{d}}$ being the estimate of \mathbf{d} , $\boldsymbol{\Lambda} \in \mathbb{R}^{3 \times 3}$ a diagonal matrix with strictly positive entries λ_i , and $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$ the tracking error vector.

By applying the control law (18) to the robot dynamics (17), the error dynamics becomes

$$\ddot{\tilde{\mathbf{q}}} + 2\boldsymbol{\Lambda}\dot{\tilde{\mathbf{q}}} + \boldsymbol{\Lambda}^2\tilde{\mathbf{q}} = \mathbf{d} - \hat{\mathbf{d}} \tag{19}$$

Inspired by the sliding mode method, a combined error vector is adopted:

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}} + \boldsymbol{\Lambda}\tilde{\mathbf{q}} \tag{20}$$

Thus, if (20) is considered in the description of the closed-loop system, Eq. (19) becomes:

$$\dot{s} + \Lambda s = d - \hat{d} \tag{21}$$

From (19) to (21), it can be verified that in the ideal case, i.e., when $\hat{d} = d$ (perfect estimation), both tracking error \tilde{q} and combined error s exponentially converge to zero. If it is not the case, closed-loop dynamics is driven by the approximation error. Furthermore, it also suggests that the combined error s represents a reasonable metric for tracking success and can be used to help calculate the estimate \hat{d} .

Then, assuming that artificial neural networks can ensure universal approximation [26] with an arbitrary precision ϵ_i , it follows that $d_i = \hat{d}_i^* + \epsilon_i$, with \hat{d}_i^* being the output related to the optimal weight vector w_i^* .

Let us now turn the proposed control law into an intelligent scheme, allowing the estimate \hat{d} to be calculated by RBF neural networks:

$$\hat{d}_i = w_i^T \psi_i(s_i), \quad i = 1, 2, 3 \tag{22}$$

where \hat{d}_i are the components of \hat{d} , $w_i = [w_{i,1} \dots w_{i,n_i}]^T$ stand for the weight vectors, and $\psi_i(s_i) = [\psi_{i,1} \dots \psi_{i,n_i}]^T$ represent the vectors with the activation functions $\psi_{i,j}$, with $i = 1, 2, 3$, $j = 1, \dots, n_i$, and n_i being the number of neurons in the hidden layer of the corresponding i th component of \hat{d} . Figure 3 presents the architecture of the radial basis function network.

The boundedness and convergence properties of closed-loop signals in the presence of modeling inaccuracies can now be proved by means of a Lyapunov-type stability analysis. Thus, let a positive-definite function V_i be defined for each component of s :

$$V_i(t) = \frac{1}{2} s_i^2 + \frac{1}{2\eta_i} \delta_i^T \delta_i \tag{23}$$

with η_i being a strictly positive constant and $\delta_i = w_i - w_i^*$.

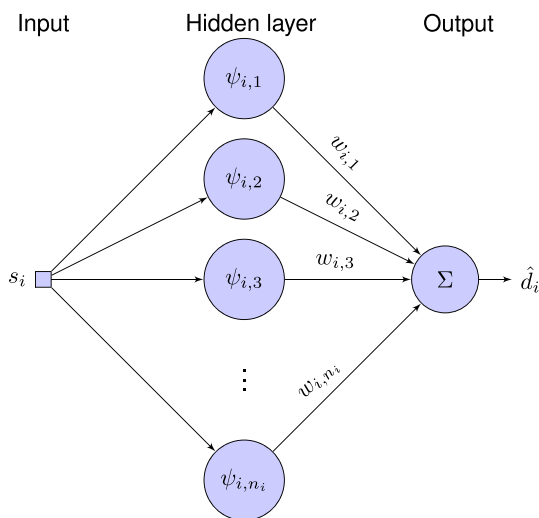


Fig. 3 Architecture of the RBF neural network

Since $\dot{\delta}_i = \dot{w}_i$, the time derivative of V_i is

$$\begin{aligned} \dot{V}_i(t) &= s_i \dot{s}_i + \eta_i^{-1} \delta_i^T \dot{w}_i \\ &= [-\lambda_i s_i + d_i - \hat{d}_i] s_i + \eta_i^{-1} \delta_i^T \dot{w}_i \\ &= -[\lambda_i s_i - (\hat{d}_i^* + \epsilon_i - \hat{d}_i)] s_i + \eta_i^{-1} \delta_i^T \dot{w}_i \\ &= -[\lambda_i s_i - \epsilon_i + \delta_i^T \psi_i] s_i + \eta_i^{-1} \delta_i^T \dot{w}_i \\ &= -[\lambda_i s_i - \epsilon_i] s_i + \eta_i^{-1} \delta_i^T [\dot{w}_i - \eta_i s_i \psi_i] \end{aligned}$$

Hence, by updating w_i according to $\dot{w}_i = \eta_i s_i \psi_i$, the time derivative \dot{V}_i becomes

$$\begin{aligned} \dot{V}_i(t) &= -[\lambda_i s_i - \epsilon_i] s_i \\ &\leq -[\lambda_i |s_i| - \epsilon_i] |s_i| \end{aligned}$$

which means that \dot{V}_i is negative definite when $|s_i| > \epsilon_i / \lambda_i$. Thus, the bounds of w_i cannot be ensured with $\dot{w}_i = \eta_i s_i \psi_i$ when $|s_i| \leq \epsilon_i / \lambda_i$. Fortunately, the projection algorithm [40] can be evoked to guarantee that w_i will remain within a convex region $\mathcal{W}_i = \{w_i \in \mathbb{R}^n : w_i^T w_i \leq \mu_i^2\}$:

$$\dot{w}_i = \begin{cases} \eta_i s_i \psi_i & \text{if } \|w_i\|_2 < \mu_i \text{ or} \\ & \text{if } \|w_i\|_2 = \mu_i \text{ and} \\ & \eta_i s_i w_i^T \psi_i < 0 \\ \left(I - \frac{w_i w_i^T}{w_i^T w_i} \right) \eta_i s_i \psi_i & \text{otherwise} \end{cases} \tag{24}$$

where μ_i is the desired upper bound of $\|w_i\|_2$.

Since $\|w_i(0)\|_2 \leq \mu_i$, it follows that $|s_i| \leq \epsilon_i / \lambda_i$ and $\|w_i(t)\|_2 \leq \mu_i$ as $t \rightarrow \infty$. Then, recalling that $s_i = \dot{q}_i + \lambda_i \tilde{q}_i$, it follows that

$$-\lambda_i^{-1} \epsilon_i \leq \dot{q}_i + \lambda_i \tilde{q}_i \leq \lambda_i^{-1} \epsilon_i \tag{25}$$

Multiplying (26) by $e^{\lambda_i t}$ and integrating the resulting inequalities between 0 and t , yields

$$-\lambda_i^{-1} \epsilon_i e^{\lambda_i t} \leq (\dot{q}_i + \lambda_i \tilde{q}_i) e^{\lambda_i t} \leq \lambda_i^{-1} \epsilon_i e^{\lambda_i t}$$

$$-\lambda_i^{-1} \epsilon_i e^{\lambda_i t} \leq \frac{d}{dt} (\tilde{q}_i e^{\lambda_i t}) \leq \lambda_i^{-1} \epsilon_i e^{\lambda_i t}$$

Integrating between 0 and t , and dividing by $e^{\lambda_i t}$, gives

$$-\frac{\epsilon_i}{\lambda_i^2} - \left[|\tilde{q}_i(0)| + \frac{\epsilon_i}{\lambda_i^2} \right] e^{-\lambda_i t} \leq \tilde{q}_i \leq \frac{\epsilon_i}{\lambda_i^2} + \left[|\tilde{q}_i(0)| + \frac{\epsilon_i}{\lambda_i^2} \right] e^{-\lambda_i t}$$

For $t \rightarrow \infty$, it becomes:

$$-\frac{\epsilon_i}{\lambda_i^2} \leq \tilde{q}_i \leq \frac{\epsilon_i}{\lambda_i^2} \tag{26}$$

Applying (26) to (25), yields

$$-2 \frac{\epsilon_i}{\lambda_i} \leq \dot{q}_i \leq 2 \frac{\epsilon_i}{\lambda_i} \tag{27}$$

This implies that the proposed control scheme ensures the exponential convergence of the tracking error to the closed region $\mathcal{Q} = \{(\tilde{q}, \dot{\tilde{q}}) \in \mathbb{R}^6 : |\tilde{q}_i| \leq \epsilon_i \lambda_i^{-2} \text{ and } |\dot{\tilde{q}}_i| \leq 2\epsilon_i \lambda_i^{-1}\}$.

It is important to emphasize that the proposed control scheme, defined by Eqs. (18), (22) and (24), has all the required features, namely adaptation, learning and prediction, which enables it to be called an intelligent controller according to Bessa et al.[41].

The block diagram of the proposed intelligent control scheme is shown in Fig. 4.

4 Numerical and experimental results

The effectiveness of the proposed control scheme is now evaluated first through numerical studies and then with the help of a three-wheeled OMR.

4.1 Numerical studies

Numerical simulations were performed in C++, by applying the proposed control scheme to the robot dynamics (17). The system of ordinary differential equations was numerically solved using the fourth-order Runge–Kutta method with sampling rates of 10 Hz for the controller and 400 Hz for the dynamic model. The Armadillo C++ library [42] was used to assist in scientific computing tasks. To evaluate control performance in the presence of modeling uncertainties and external disturbances, the term d is calculated as $d = \Delta M\ddot{q} - \xi$, where $\Delta M\ddot{q}$ takes into account the uncertainty in the inertia matrix and $\xi = \Xi\dot{q}$ stands for the unknown friction effects, with $\Xi = \text{diag}\{5;4;12\}$.

Regarding the other model parameters, they are chosen as: $m_r = 11.2$ kg, $l = 0.294$ m, $\beta = 60^\circ$, $r = 0.051$ m, $R_r = 0.27$ m, and $g = 9.81$ m/s². In the control design, an error of 10% is assumed with respect the inertial matrix and the gains are set to $\lambda_i = 1$. Considering the network architecture presented in Fig. 3, three RBF networks are considered, one for each state variable, where each one contains one input (s_i), one output (\hat{d}_i) and seven neurons in the hidden layer. The number of neurons was chosen to ensure a good trade-off between control accuracy and computational

burden. Gaussian functions are adopted for the neurons in the hidden layer: $\psi_i(s_i; c_i, \sigma_i) = \exp\{-0.5[(s_i - c_i)/\sigma_i]^2\}$, where $c_i = [-\gamma_i/2; -\gamma_i/4; -\gamma_i/8; 0.0; \gamma_i/8; \gamma_i/4; \gamma_i/2]$ and $\sigma_i = [\gamma_i/3; \gamma_i/5; \gamma_i/15; \gamma_i/30; \gamma_i/15; \gamma_i/5; \gamma_i/3]$, with $\gamma = [0.5; 0.5; 3.0]$. In order to let the robot learn about its environment from scratch, the weight vector is initialized as $w_i = \mathbf{0}$ and updated according to (24), with learning rates $\eta = [30; 30; 35]$. Figure 5 shows the tracking performance associated with the trajectory $q_d = [r \cos(\alpha t) \quad -r \sin(\alpha t) \quad -(\alpha t + \pi/2)]^T$, with $r = 0.5$ and $\alpha = \pi/15$. The influence of the neural network on the control performance is evaluated by comparing the proposed intelligent approach with a conventional nonlinear controller, namely the computed torque scheme, which is commonly used in robotics. In fact, the computed torque scheme can be easily obtained from the intelligent approach by simply disregarding the contribution of the RBF networks to the control signal, i.e., by setting $\hat{d} = \mathbf{0}$.

By observing the obtained numerical results, it can be seen that the intelligent controller is capable of tracking the desired trajectory, presenting a strongly improved performance when compared to the conventional approach, Fig. 5a. The efficacy of the proposed control scheme can also be verified by noting that the conventional controller presented a significant residual tracking error, while the intelligent approach was able to practically zero it, Fig. 5b, with a comparable control effort, Fig. 5c. This improved tracking response, as a matter of fact, is due to the capacity of the neural network to compensate for modeling uncertainties and external disturbances, Fig. 5d, which quite often degrade the performance of conventional controllers.

4.2 Experimental validation

The performance of the intelligent controller is now assessed by means of experimental results obtained with the omnidirectional mobile robot Robotino®. The robot consists of three controllable mecanum-like wheels equipped with optical encoders, nine infrared distance sensors, a bumper sensor, a digital gyroscope, and an embedded PC 104 with real-time Linux operating system. The proposed control scheme was implemented in C++ using the Robotino® API.

Fig. 4 Block diagram of the intelligent control scheme

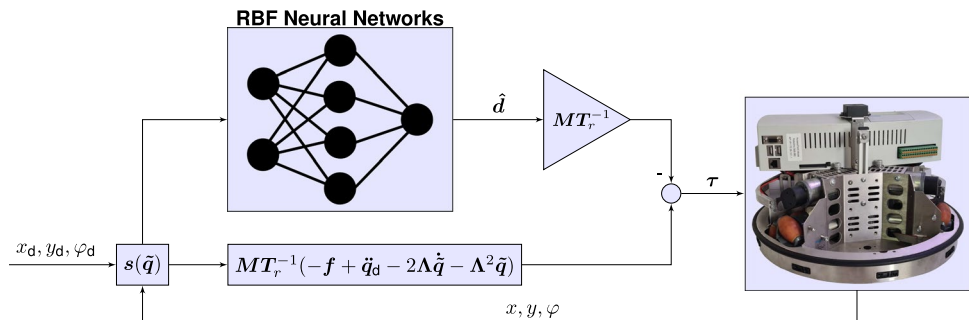
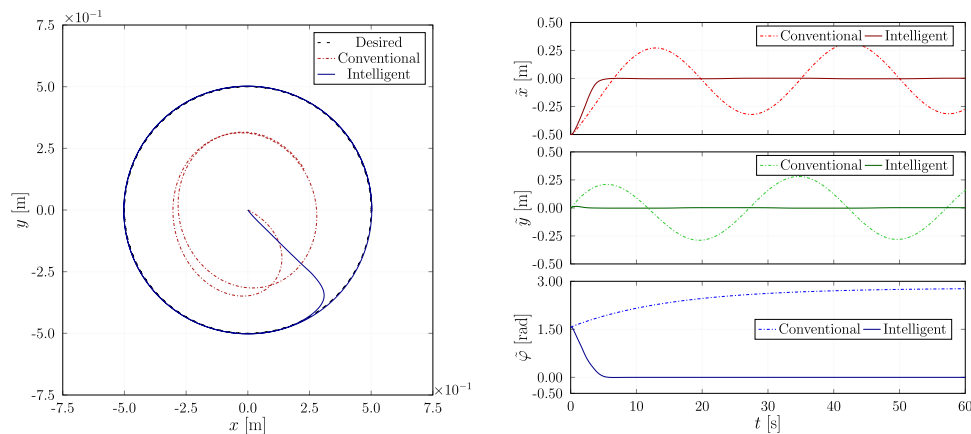
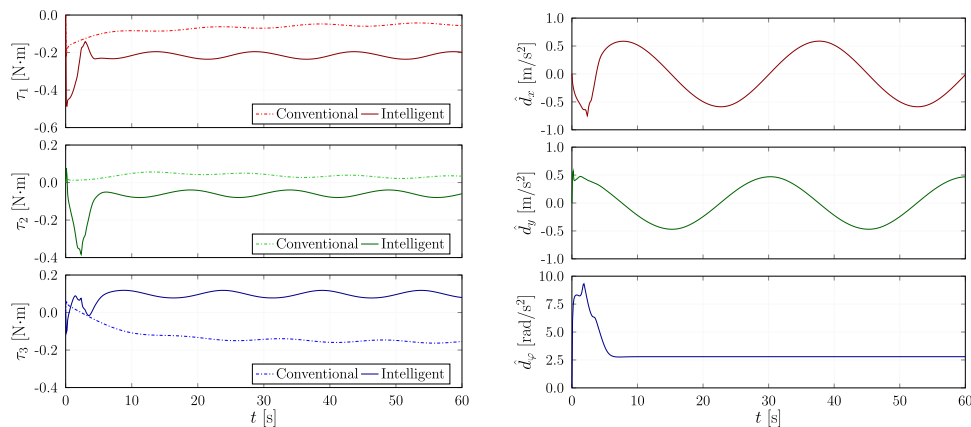


Fig. 5 Numerical results for the tracking of $q_d = [r \cos(\alpha t) - r \sin(\alpha t) - (\alpha t + \pi/2)]^T$



(a) Trajectory tracking.

(b) Tracking error.



(c) Control signals.

(d) Uncertainty estimates.

All control parameters are the same used in numerical studies. Uncertainties regarding the inertia matrix and no prior knowledge of the friction effects are once again assumed. The robot's position and orientation are provided by the odometry system using the gyroscope and motor encoders. It should be noted that, instead of torques, the robot requires the angular speeds of the wheels as control inputs, which is easily obtained with a first order filter. Moreover, as both conventional and intelligent controllers need the robot speed, i.e., $\dot{q} = [\dot{x} \ \dot{y} \ \dot{\varphi}]^T$, sliding mode observers [43] are used for this purpose. The deployed computer code and all obtained experimental data can be accessed at <https://github.com/RoboteamUFRN/Intelligent-Control-Robotino>.

Figures 6 and 7 show the obtained results for the tracking of $q_d = [r \cos(\alpha t) - r \sin(\alpha t) - (\alpha t + \pi/2)]^T$.

As shown in Fig. 6, the intelligent controller allows the robot to track the desired trajectory with a small associated error. Some video frames captured during the experiment are also overlaid to allow a better understanding of the obtained results. Furthermore, the results presented in Fig. 7 confirm the much superior performance of the proposed controller, as well as the effectiveness of the

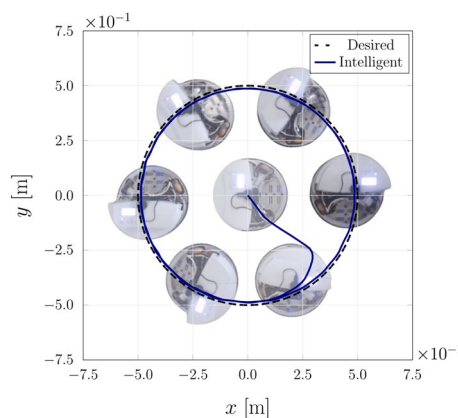


Fig. 6 Experimental results and overlaid video frames for the trajectory tracking with Robotino®

adaptive neural network in compensating for uncertainties and disturbances, Fig. 7d. As can be seen in Fig. 7b, the intelligent control scheme is able to provide a much smaller tracking error when compared to the conventional approach.

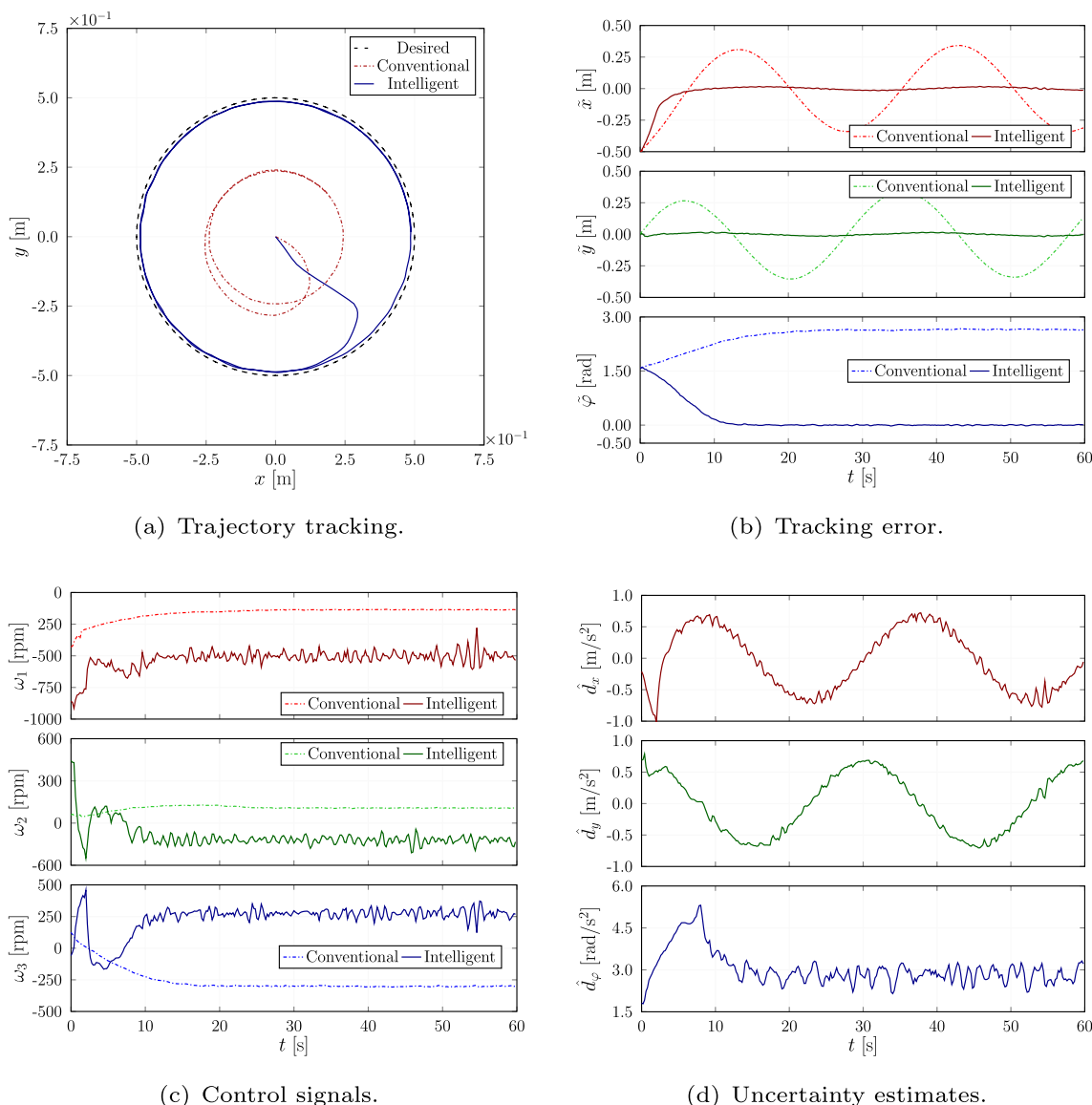


Fig. 7 Experimental results for the tracking of $q_d = [r \cos(at) - r \sin(at) - (at + \pi/2)]^T$

In fact, for a quantitative comparison, the absolute errors obtained in steady-state can be summed according to the equation below:

$$E_{\tilde{z}} = \int_{t_1}^{t_2} |\tilde{z}| dt \tag{28}$$

with \tilde{z} representing the tracking error for one of the three states. Table 1 shows the results for both conventional and intelligent controllers, related to the tracking errors in Fig. 7b obtained from $t_1 = 30$ s to $t_2 = 60$ s.

From Table 1 it can be observed that when incorporating the RBF networks in the control law, the tracking error obtained with the intelligent controller in the steady-state

phase is from 95.83 to 99.73% lower, which confirms its much superior performance when compared to the conventional approach.

Table 1 Comparative analysis of the tracking error obtained with conventional and intelligent controllers: sum of the absolute errors during the steady-state phase

	Conventional	Intelligent	Error reduction (%)
$E_{\tilde{x}}$	32.39	1.35	95.83
$E_{\tilde{y}}$	32.32	1.23	96.19
$E_{\tilde{\varphi}}$	397.45	1.09	99.73

The full video of the trajectory tracking with Robotino®, as well as the animated plots with both control schemes, can be accessed at <https://youtu.be/GT9OINhxZpo>.

5 Concluding remarks

This work introduces a novel intelligent controller for accurate trajectory tracking with omnidirectional mobile robots. The proposed scheme takes advantage of a Lyapunov-based nonlinear approach, but adaptive neural networks are also included in the control law to deal with modeling uncertainties and external disturbances. The boundedness and convergence properties of all closed-loop signals are analytically proved.

Moreover, the proposed intelligent controller is able to fulfill the four essential requirements mentioned at the beginning of this work:

1. The intelligent control scheme can adequately handle the dynamic effects associated with the robot's motion, rather than just kinematic control, which allows accurate trajectory tracking even when subjected to unmodeled dynamics;
2. Online learning, rather than offline training, is adopted to allow the robot to continuously improve its overall performance in a dynamically changing environment;
3. The neural network can learn to compensate for uncertainties and disturbances by minimizing the combined error measurement, making direct measurements of the signal to be estimated (which is not possible in real applications) completely unnecessary;
4. As only s_i is used as input to the neural networks, instead of the states or state errors, the complexity of the intelligent compensation scheme can be considered quite low, making its implementation possible even on the low-power embedded hardware of mobile robots.

Finally, the effectiveness of the proposed control scheme is evaluated both numerically, through simulation studies, and experimentally with a three-wheeled omnidirectional mobile robot. Both numerical and experimental results confirm the greatly improved performance achieved by the intelligent controller. The quantitative comparison of the experimental results obtained with the conventional and intelligent controllers showed that the proposed approach is capable of providing tracking errors from 95.83 to 99.73% lower, thus ensuring accurate trajectory tracking even in the presence of high levels of uncertainty.

From the results presented in this work, it can be verified that intelligent control schemes are essential to deal with the high level of uncertainties to which mobile robots are subject. However, it is worth mentioning that control systems,

whether they are intelligent or not, still depend on a previously established reference. In mobile robotics, this reference is often defined by a path planning algorithm, which along with perception, localization and tracking control is also one of the most researched subjects nowadays. Path planning is, nonetheless, a NP (nondeterministic polynomial-time) hard problem and in some applications, such as self-driving cars, it usually involves many ethical issues. Therefore, future works in mobile robotics should focus on the development of integrated strategies for simultaneous motion planning and control, where a single algorithm is able to plan and control the trajectory to be followed in a smooth and holistic way.

Appendix A: Transformation matrix

$$\mathbf{T}_r = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \quad (29)$$

$$t_{11} = \frac{r[\sin(\beta - \varphi) + \sin \varphi]}{\cos 2\beta - 1} \quad (30)$$

$$t_{12} = \frac{r \sin \varphi}{\cos \beta + 1} \quad (31)$$

$$t_{13} = \frac{r[\sin \varphi - \sin(\beta + \varphi)]}{\cos 2\beta - 1} \quad (32)$$

$$t_{21} = \frac{r[\cos(\beta - \varphi) - \cos \varphi]}{\cos 2\beta - 1} \quad (33)$$

$$t_{22} = -\frac{r \cos \varphi}{\cos \beta + 1} \quad (34)$$

$$t_{23} = \frac{r[\cos(\beta + \varphi) - \cos \varphi]}{\cos 2\beta - 1} \quad (35)$$

$$t_{31} = t_{33} = \frac{r}{2l(\cos \beta + 1)} \quad (36)$$

$$t_{32} = \frac{r \cos \beta}{l(\cos \beta + 1)} \quad (37)$$

Inertia matrix

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (38)$$

$$m_{11} = m_{33} = \frac{m_r r^2 \sec^4(\beta/2)[2l^2 \csc^2(\beta/2) + R_r^2]}{32l^2} \quad (39)$$

$$m_{12} = \frac{m_r r^2 \sec^4(\beta/2)(R_r^2 \cos \beta - 2l^2)}{16l^2} \quad (40)$$

$$m_{13} = m_{31} = \frac{m_r r^2 \sec^4(\beta/2) \left(\frac{4l^2 \cos \beta}{\cos \beta - 1} + R_r^2 \right)}{32l^2} \quad (41)$$

$$m_{22} = \frac{m_r r^2 \sec^4(\beta/2)(2l^2 + R_r^2 \cos^2 \beta)}{8l^2} \quad (42)$$

$$m_{21} = m_{23} = m_{32} = m_{12} \quad (43)$$

Funding Open Access funding provided by University of Turku (UTU) including Turku University Central Hospital. The authors acknowledge the support of the Brazilian research agencies CNPq and CAPES.

Code availability The computer code developed for the experiments with Robotino® and all the experimental data obtained can be accessed at <https://github.com/RoboteamUFRN/Intelligent-Control-Robotino>.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Niloy MAK, Shama A, Chakraborty RK, Ryan MJ, Badal FR, Tasneem Z, Ahamed MH, Moyeen SI, Das SK, Ali MF, Islam MR, Saha DK (2021) Critical design and control issues of indoor autonomous mobile robots: A review. *IEEE Access* 9:35338–35370. <https://doi.org/10.1109/ACCESS.2021.3062557>
2. Tzafestas SG (2018) Mobile robot control and navigation: A global overview. *J Intell Robot Syst* 91(1):35–58. <https://doi.org/10.1007/s10846-018-0805-9>
3. Ding T, Zhang Y, Ma G, Cao Z, Zhao X, Tao B (2022) Trajectory tracking of redundantly actuated mobile robot by MPC velocity control under steering strategy constraint. *Mechatron* 84:102779. <https://doi.org/10.1016/j.mechatronics.2022.102779>
4. Saenz A, Santibañez V, Bugarin E, Dzul A, Ríos H, Villalobos-Chin J (2021) Velocity control of an omnidirectional wheeled mobile robot using computed voltage control with visual feedback: Experimental results. *Int J Control Autom Syst* 19(2):1089–1102. <https://doi.org/10.1007/s12555-019-1057-6>
5. Rubio F, Valero F, Llopis-Albert C (2019) A review of mobile robots: Concepts, methods, theoretical framework, and applications. *Int J Adv Robot Syst* 16(2):1729881419839596. <https://doi.org/10.1177/1729881419839596>
6. Sun Z, Hu S, He D, Zhu W, Xie H, Zheng J (2021) Trajectory-tracking control of mecanum-wheeled omnidirectional mobile robots using adaptive integral terminal sliding mode. *Comput Electr Eng* 96:107500. <https://doi.org/10.1016/j.compeleceng.2021.107500>
7. Jeong S, Chwa D (2021) Sliding-mode-disturbance-observer-based robust tracking control for omnidirectional mobile robots with kinematic and dynamic uncertainties. *IEEE/ASME Trans Mechatron* 26(2):741–752. <https://doi.org/10.1109/TMECH.2020.2998506>
8. Cao G, Zhao X, Ye C, Yu S, Li B, Jiang C (2022) Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot. *J Mech Sci Technol* 36(4):2019–2029. <https://doi.org/10.1007/s12206-022-0337-x>
9. Zhao T, Zou X, Dian S (2022) Fixed-time observer-based adaptive fuzzy tracking control for mecanum-wheel mobile robots with guaranteed transient performance. *Nonlinear Dyn* 107(1):921–937. <https://doi.org/10.1007/s11071-021-06985-0>
10. Zou X, Zhao T, Dian S (2022) Finite-time adaptive interval type-2 fuzzy tracking control for mecanum-wheel mobile robots. *Int J Fuzzy Syst* 24(3):1570–1585. <https://doi.org/10.1007/s40815-021-01211-w>
11. Siciliano B, Khatib O (2016) Springer handbook of robotics, 2nd edn. Springer, Berlin, Heidelberg
12. Tork N, Amirkhani A, Shokouhi SB (2021) An adaptive modified neural lateral-longitudinal control system for path following of autonomous vehicles. *Eng Sci Technol Int J* 24(1):126–137. <https://doi.org/10.1016/j.jestech.2020.12.004>
13. Nascimento TP, Dórea CET, Gonçalves LMG (2018) Nonholonomic mobile robots' trajectory tracking model predictive control: a survey. *Robot* 36(5):676–696. <https://doi.org/10.1017/S0263574717000637>
14. Ren C, Li C, Hu L, Li X, Ma S (2022) Adaptive model predictive control for an omnidirectional mobile robot with friction compensation and incremental input constraints. *Trans Inst Meas Control* 44(4):835–847. <https://doi.org/10.1177/01423312211021321>
15. Wu H-M, Karkoub M (2022) Frictional forces and torques compensation based cascaded sliding-mode tracking control for an uncertain omnidirectional mobile robot. *Meas Control* 55(3–4):178–188. <https://doi.org/10.1177/00202940221092033>
16. Zijie N, Qiang L, Yonjie C, Zhijun S (2019) Fuzzy control strategy for course correction of omnidirectional mobile robot. *Int J Control Autom Syst* 17(9):2354–2364. <https://doi.org/10.1007/s12555-018-0633-5>
17. Krichen N, Masmoudi MS, Derbel N (2021) Autonomous omnidirectional mobile robot navigation based on hierarchical fuzzy systems. *Eng Comput* 38(2):989–1023. <https://doi.org/10.1108/EC-08-2019-0380>
18. Huang J-T, Chiu C-K (2021) Adaptive fuzzy sliding mode control of omnidirectional mobile robots with prescribed performance. *Process* 9(12):2211. <https://doi.org/10.3390/pr9122211>
19. Abiyev RH, Akkaya N, Günsel I (2019) Control of omnidirectional robot using z-number-based fuzzy system. *IEEE Trans Syst Man Cybern: Syst* 49(1):238–252. <https://doi.org/10.1109/TSMC.2018.2834728>

20. Zijie N, Peng Z, Cui Y, Jun Z (2022) PID control of an omnidirectional mobile platform based on an rbf neural network controller. *Ind Robot* 49(1):65–75. <https://doi.org/10.1108/IR-01-2021-0015>
21. Pham TT, Le MT, Nguyen C-N (2021) Omnidirectional mobile robot trajectory tracking control with diversity of inputs. *Int J Mech Eng Robot Res* 10(11):639–644. <https://doi.org/10.18178/ijmerr.10.11.639-644>
22. Wang C, Wang D, Han Y (2021) Neural network based adaptive dynamic surface control for omnidirectional mobile robots tracking control with full-state constraints and input saturation. *Int J Control Autom Syst* 19(12):4067–4077. <https://doi.org/10.1007/s12555-020-0582-7>
23. Kim DHT, Manh TN, Manh CN, Nguyen ND, Tien DP, Van MT, Xuan MP (2021) Adaptive control for uncertain model of omnidirectional mobile robot based on radial basis function neural network. *Int J Control Autom Syst* 19(4):1715–1727. <https://doi.org/10.1007/s12555-019-1004-6>
24. Zheng W, Ito T (2019) Dynamic surface control-based adaptive neural tracking for full-state constrained omnidirectional mobile robots. *Adv Mech Eng* 11(4):1687814019846750. <https://doi.org/10.1177/1687814019846750>
25. Jiang Y, Yang C, Wang M, Wang N, Liu X (2018) Bioinspired control design using cerebellar model articulation controller network for omnidirectional mobile robots. *Adv Mech Eng* 10(8):1687814018794349. <https://doi.org/10.1177/1687814018794349>
26. Scarselli F, Tsoi AC (1998) Universal approximation using feed-forward neural networks: A survey of some existing methods, and some new results. *Neural Netw* 11(1):15–37. [https://doi.org/10.1016/S0893-6080\(97\)00097-X](https://doi.org/10.1016/S0893-6080(97)00097-X)
27. Dos Santos JDB, Bessa WM (2019) Intelligent control for accurate position tracking of electrohydraulic actuators. *Electron Lett* 55(2):78–80. <https://doi.org/10.1049/el.2018.7218>
28. Bessa WM, Kreuzer E, Lange J, Pick M-A, Solowjow E (2017) Design and adaptive depth control of a micro diving agent. *IEEE Robot Autom Lett* 2(4):1871–1877. <https://doi.org/10.1109/LRA.2017.2714142>
29. Amirkhani A, Barshooi AH (2022) Consensus in multi-agent systems: a review. *Artif Intell Rev* 55(5):3897–3935. <https://doi.org/10.1007/s10462-021-10097-x>
30. Raj L, Czmerk A (2017) Modelling and simulation of the drivetrain of an omnidirectional mobile robot. *Autom* 58:232–243. <https://doi.org/10.1080/00051144.2017.1391612>
31. Barreto Sobrinho JCL, Conceição AGS, Dórea CET, Martinez L, De Pieri ER (2014) Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot. *IEEE/ASME Trans Mechatron* 19(2):467–476. <https://doi.org/10.1109/TMECH.2013.2243161>
32. Lima GS, Trimpe S, Bessa WM (2020) Sliding mode control with gaussian process regression for underwater robots. *J Intell Robot Syst* 99(3):487–498. <https://doi.org/10.1007/s10846-019-01128-5>
33. Lima GS, Porto DR, de Oliveira AJ, Bessa WM (2021) Intelligent control of a single-link flexible manipulator using sliding modes and artificial neural networks. *Electron Lett* 57(23):869–872. <https://doi.org/10.1049/el12.12300>
34. da Silva Lima G, Moreira Bessa W (2022) Sliding mode control with gaussian process regression for underactuated mechanical systems. *IEEE Lat Am Trans* 20(6):963–969. <https://doi.org/10.1109/TLA.2022.9757739>
35. Bessa WM, Otto S, Kreuzer E, Seifried R (2019) An adaptive fuzzy sliding mode controller for uncertain underactuated mechanical systems. *J Vib Control* 25(9):1521–1535. <https://doi.org/10.1177/1077546319827393>
36. Bessa WM, De Paula AS, Savi MA (2014) Adaptive fuzzy sliding mode control of a chaotic pendulum with noisy signals. *Z Angew Math Mech* 94(3):256–263. <https://doi.org/10.1002/zamm.20120214>
37. Bessa WM, De Paula AS, Savi MA (2012) Sliding mode control with adaptive fuzzy dead-zone compensation for uncertain chaotic systems. *Nonlinear Dyn* 70(3):1989–2001. <https://doi.org/10.1007/s11071-012-0591-z>
38. Bessa WM, De Paula AS, Savi MA (2013) Adaptive fuzzy sliding mode control of smart structures. *Eur Phys J Spec Top* 222(7):1541–1551. <https://doi.org/10.1140/epjst/e2013-01943-7>
39. Slotine J-JE, Li W (1991) *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, New Jersey
40. Ioannou P, Fidan B (2006) *Adaptive Control Tutorial*. SIAM, Philadelphia
41. Bessa WM, Brinkmann G, Duecker DA, Kreuzer E, Solowjow E (2018) A biologically inspired framework for the intelligent control of mechatronic systems and its application to a micro diving agent. *Math Probl Eng* 2018:1–16. <https://doi.org/10.1155/2018/9648126>
42. Sanderson C, Curtin R (2016) Armadillo: a template-based C++ library for linear algebra. *J Open Source Softw* 1(2):26. <https://doi.org/10.21105/joss.00026>
43. Shtessel Y, Edwards C, Fridman L, Levant A (2014) *Sliding mode control and observation*. Birkhäuser, New York, NY

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.