Engineering Research Journal

journal homepage: https://erj.journals.ekb.eg/

Trajectory Tracking of Wheeled Mobile Robot Through System Identification and Control Using Deep Neural Network

Mahmoud Gamal Abdalnasser[1,*], A. A. Ibrahim[1], Mahmoud Elsamanty[1,2]

[1]Mechanical Engineering Department, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt

[2]Mechatronics and Robotics Department, School of Innovative Design Engineering, Egypt-Japan University of Science and Technology, Alexandria, Egypt

*Corresponding Author E-mail: mahmoud.abdenaser@feng.bu.edu.eg

Abstract

Trajectory tracking is a fundamental requirement for wheeled mobile robots, particularly in industrial applications that demand precise motion control. This experimental study presents a trajectory-tracking control strategy for a four-wheeled mecanum robot. The work begins with the derivation of the robot kinematic model. Following this, a system identification approach is employed to capture the robot's dynamic behavior accurately. This involves collecting and analyzing input-output data to develop a comprehensive dynamic model of the robot. A deep neural network (DNN) is utilized as a black-box model to effectively learn and represent the complex nonlinear behavior of the system. The DNN is trained on extensive input-output data, ensuring it can generalize well to various operational scenarios. Once the dynamic model is established, a separate DNN-based controller is designed. This controller leverages the insights gained from the dynamic model to generate precise control signals, enabling the robot to follow the desired trajectory accurately. The proposed system identification method demonstrates remarkable accuracy, achieving a 99.98% fit to the training data, which is indicative of the model's robustness and reliability. To validate the effectiveness of the approach, experimental tests are conducted using an infinity-shaped trajectory. The results are highly promising, with the controller achieving precise tracking marked by a mean squared error of 0.0005 meter. This level of precision highlights the potential of deep learning techniques in addressing complex control challenges in wheeled mobile robots. The combination of system identification and deep learning offers a powerful toolset for developing advanced control systems.

Keywords: Wheeled Mobile Robots, Trajectory Tracking, Mecanum Robot, System Identification, Deep Neural Networks.

DOI: 10.21608/erj.2024.304953.1075
Received 16 July 2024; Received in revised form 07 August 2024; Accepted 09 August 2024
Available online 01 September 2024

1. Introduction

Robotics is a key technology in Industry 4.0, revolutionizing the manufacturing industry by enhancing automation systems. It enables precise and cost-effective execution of repetitive tasks, significantly improving efficiency and productivity [1]. The integration of advanced robotics into industrial processes has led to the development of smarter and more adaptive manufacturing environments. These systems can operate continuously with minimal human intervention, reducing labor costs and minimizing human error. Wheeled mobile robots (WMRs) are an integral component of this robotic revolution, finding a wide range of applications across various industries. Their ability to navigate both static and dynamic environments enhances their utility. These robots are equipped with sophisticated sensors and control algorithms, allowing them to perform complex maneuvers and navigate through cluttered spaces with high precision. This flexibility makes them suitable for a variety of tasks. In agricultural applications, WMRs are utilized for activities such as planting, harvesting, and monitoring crop health, thereby increasing agricultural productivity and sustainability [2]. In manufacturing lines, these robots streamline production processes by transporting materials and components between different stages of production, enhancing overall workflow efficiency [3]. Warehouses employ WMRs in storage and retrieval systems, where they autonomously manage inventory, optimize space utilization, and expedite order fulfillment [4-6].

Furthermore, WMRs assist humans in hospitality and healthcare settings by providing services such as room delivery, patient monitoring, and mobility assistance, thereby improving service quality and patient care [7]. In material handling, these robots are used to transport heavy loads, reducing the physical strain on human workers and increasing operational safety [8- 11]. The versatility of WMRs in these diverse applications underscores their importance in modern industrial and service sectors, driving continuous innovation and development in the field of robotics.

System identification is crucial for understanding the dynamics of linear and nonlinear systems. Through this process, accurate representations of how systems respond to inputs can be built. System identification techniques are broadly categorized into black-box and gray-box approaches. Black-box methods rely only on input/output data, assuming no prior knowledge of the system internal dynamics [12].

In contrast, gray-box approaches combine knowledge of the system dynamics with data-driven insights, leveraging both existing understanding and information extracted from input/output data [13]. Parameters can be estimated by first defining the equations of motion for the system. Then, a simulation is created using Simscape physics-based components, and the output data from the model is compared to experimental data. Finally, a least squares estimation technique is employed to determine the optimal parameter values [14].

Neural networks have become increasingly important in robotics, particularly in the areas of system identification and control. Their ability to learn and generalize complex nonlinear system dynamics from sensor data has proven highly effective, enabling more sophisticated and adaptive robotic systems [10]. The versatility of neural networks allows them to be applied in various configurations and architectures, each suited to different aspects of robotic control and identification. One of the most widely used neural network structures in robotics is the feed-forward neural network. These networks are trained using back-propagation algorithms and are adept at identifying model systems by learning the underlying patterns in data [15-17]. The back-propagation algorithm iteratively adjusts the network weights to minimize the error between the predicted and actual outputs, thereby enhancing the model accuracy.

A notable modification of the traditional multilayer feed-forward neural network is the PID neural network. This architecture incorporates dynamic characteristics into the hidden layer units, making it particularly suitable for both control and identification tasks [18]. The PID neural network leverages the principles of proportional-integral-derivative control, integrating these with neural network learning capabilities to manage dynamic systems more effectively. With the advent of deep learning, more advanced neural network structures have been developed. Deep recurrent neural networks (RNNs), for instance, have shown great promise in accurately identifying complex system behaviors due to their ability to capture temporal dependencies in data [19]. RNNs, with their feedback loops, are particularly useful in scenarios where the system past states significantly influence its future states.

Moreover, the combination of recurrent neural networks with state estimation techniques, such as the extended Kalman filter, has yielded promising results. This hybrid approach enhances the neural network ability to control and identify single-input single-output systems by effectively estimating the system states and improving the overall control accuracy [20 - 21]. The integration of these advanced neural networks into robotic systems represents a significant step forward in achieving more autonomous and intelligent robotic behavior, capable of adapting to a wide range of dynamic and complex environments.

In the context of wheeled mobile robots, different strategies can be used for control based on deep neural networks. For example, one approach utilizes a neural network to learn robot kinematics, specifically the relationship between wheel rotations, and base movement. By using sensor feedback to adjust the control signals, the system enables the robot to accurately follow a specific trajectory [22]. Another method leverages two interconnected Artificial Neural Networks to achieve optimal path planning and obstacle avoidance. The first ANN processes raw data from a LIDAR sensor to identify obstacles and determine their positions. The second ANN then takes this obstacle information, combines it with the robot current position and desired goal position, and generates an optimal trajectory for the robot to follow [23]. Finally, a hybrid controller for trajectory tracking combines two main components: a Neural Network-Based Kinematic Controller, which determines the gains of a kinematic controller, and a Model Reference Adaptive Controller, which ensures that the robot actual behavior matches that of a desired reference model [24].

The inherent challenges in trajectory tracking for Wheeled Mobile Robots are often addressed through the implementation of advanced controllers. However, the complexity and computational demands of such controllers can pose significant obstacles in hardware realization. Consequently, the development of controllers characterized by simplified structures and reduced computational burdens is essential to enhance tracking accuracy without necessitating sophisticated hardware. In this context, a neural network controller for Mecanum-Wheel Mobile Robots has been proposed [25]. This approach utilizes a two-layer feedforward neural network, trained to minimize discrepancies between its output and a predetermined reference control signal. Through continuous adaptation, the controller achieves rapid convergence to desired trajectories. Trajectory tracking in mobile robotics, particularly within unstructured and uncertain environments, remains a persistent area of concern. Interval type-2 fuzzy-like PID controllers, employing type-2 fuzzy logic sets, have demonstrated superior robustness and performance compared to their traditional type-1 counterparts. The optimization of these controllers is often achieved through the firefly optimization algorithm, resulting in minimized tracking errors and smooth velocity profiles, even in the presence of disturbances [26]. Omni-directional mobile robots equipped with Mecanum wheels exhibit exceptional maneuverability due to their independent rotational and translational capabilities.

Model predictive control algorithms have proven effective for trajectory tracking in such systems, as evidenced by simulation results [27]. For nonholonomic mobile robots, neural adaptive controllers based on dynamic models have been developed to ensure precise trajectory tracking [28 - 29]. These controllers incorporate global tracking errors into learning laws, guaranteeing asymptotic stability. Furthermore, a novel machine learning technique has been introduced to enhance trajectory tracking through neural networks [30]. This approach facilitates automatic online gain tuning, thereby reducing the need for manual adjustments. Consequently, enhanced precision and performance are achieved across diverse trajectories without requiring pre-learning phases, demonstrating significant improvements over conventional control methodologies.

Despite these advancements, there remains a relatively limited number of comprehensive studies applying neural network-based control strategies to multi-input, multi-output (MIMO) systems, particularly in the realm of wheeled mobile robots, leaving a need for more robust solutions capable of handling the intricacies of MIMO systems. This experimental research proposes to address this by developing a deep learning-based control framework tailored for MIMO wheeled mobile robots. The primary aim is to derive a dynamic model using deep neural networks that can accurately capture the complex interactions between multiple input and output variables. Subsequently, the research will design a neural network controller that leverages this dynamic model to achieve precise trajectory tracking and robust performance in various operational conditions. By doing so, this study seeks to advance the field of robotic control, providing a scalable and adaptable solution for the next generation of industrial and service robots.

2. Kinematics of Wheeled Mobile Robot

The robot is equipped with four mecanum wheels, each featuring passive rollers capable of spinning freely around their axes. These rollers are positioned at an angle relative to the wheel axis of rotation, allowing the robot to perform sideways movements without requiring traditional turning. This unique design endows the robot with exceptional maneuverability as shown in fig.1.

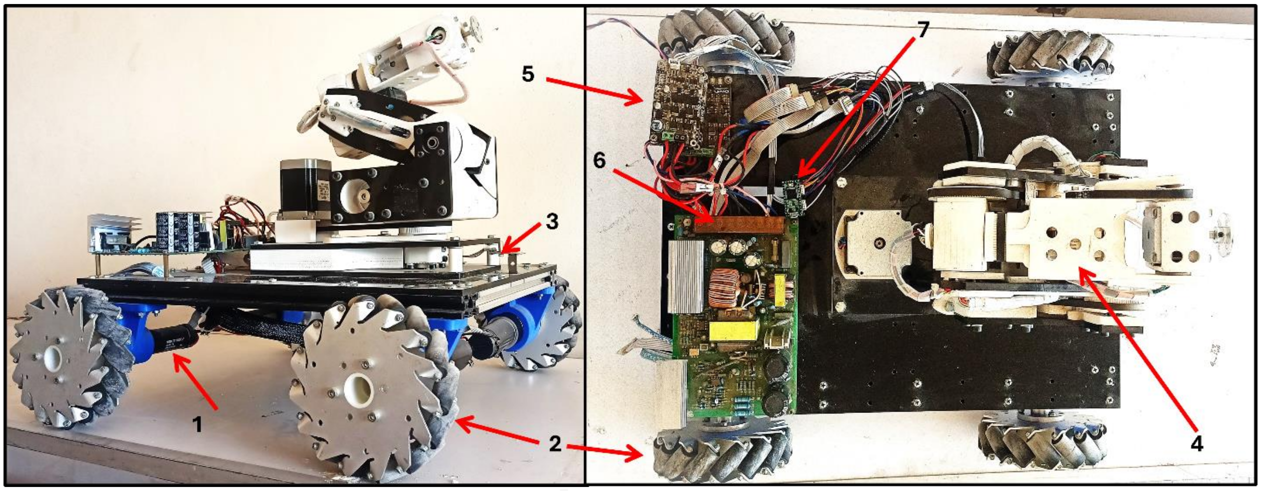

Fig. 1.a front view and top view of Wheeled mobile robot with mecanum wheels.
1 dc motor with encoder,2 mecanum wheel,3 mpu6050 imu sensor,4 arm robot,5 Cytron motor driver,6power supply,7 teensy3.2 microcontroller

In the world frame ***{I}***, the robot position is denoted by the coordinates ***[x, y, ψ]***, where ***x*** indicates the forward position, ***y*** represents the lateral position, and ***ψ*** denotes the angular orientation. Conversely, the robot velocity in the robot frame ***{B}*** is expressed by the vector ***[u, v, r]***, where ***u*** is the forward velocity, ***v*** is the lateral velocity, and r is the angular velocity. Fig.1.b illustrates the relationship between the world frame ***{I}*** and the robot frame ***{B}***.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \mathrm{ucos}(\psi) - \mathrm{vsin}(\psi) \\ \mathrm{usin}(\psi) + \mathrm{vcos}(\psi) \\ r \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{1}$$

The relationship between the robot input velocity command $\boldsymbol{\xi}$ and the derivative of its generalized coordinates $\dot{\eta}$ is given by $\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\psi})\boldsymbol{\xi}$. Here, $\boldsymbol{\eta}$ is a vector representing the robot generalized coordinates, which include its position and orientation. The vector $\boldsymbol{\xi}$ contains the robot input velocity commands. The Jacobian matrix $\boldsymbol{J}(\boldsymbol{\psi})$, which depends on the robot′s current orientation $\boldsymbol{\psi}$, maps these input commands to the corresponding changes in the generalized coordinates. The parameters $\boldsymbol{\alpha_i}$, and $\boldsymbol{\rho_i}$ denote the radius of the wheel and the roller, respectively. The variables $\boldsymbol{\omega_i}$, and $\dot{\boldsymbol{\beta}}_{\boldsymbol{i}}$ represent the angular velocity of the wheel and the rate of change of the roller angle, respectively. Moreover, $\boldsymbol{\phi_i}$ the angle between roller and x-frame. $\boldsymbol{\theta_{Bi}}$ the angle between frame $\{\boldsymbol{B}\}$ and $\{\boldsymbol{C}\}$ as shown in Fig. 2.

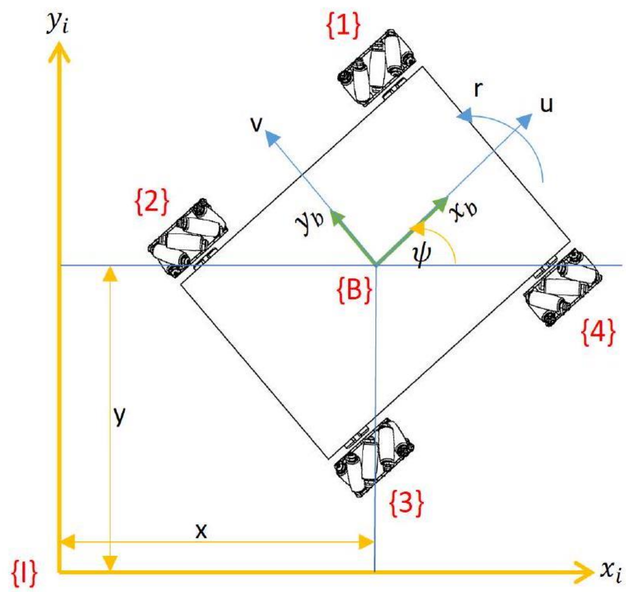

***Fig. 1.b** Geometry and coordinate frames of the wheeled mobile robot.*

Equation 2 describes the velocity of the ith roller in the y-direction of frame $\{C\}$

$$\dot{Y}_{C_i} = \dot{\beta}_i\, \rho_i \cos{(\phi_i)} \tag{2}$$

Equation 3 describes the velocity of the ith roller in the x-direction of frame {C}

$$\dot{X}_{C_i} = \omega_i\, a_i - \dot{\beta}_i\, \rho_i \sin{(\phi_i)} \tag{3}$$

$$\omega_i = [1/a_i \quad 1/a_i\ \tan(\phi_i)] \begin{bmatrix} \dot{X}_{C_i} \\ \dot{Y}_{C_i} \end{bmatrix} \tag{4}$$

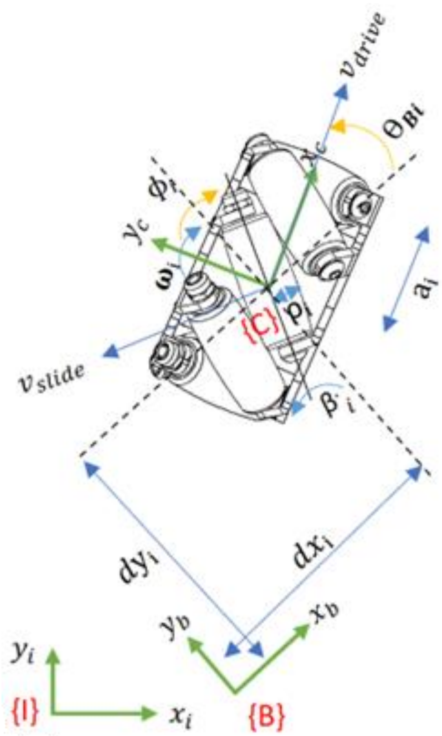

Fig. 2: Geometry of the mecanum wheel with the roller

The velocity of the frame $\{C\}$ relative to frame $\{B\}$

$$\binom{B}{c_i} v = \begin{bmatrix} \cos(\theta_{Bi}) & -\sin(\theta_{Bi}) \\ \sin(\theta_{Bi}) & \cos(\theta_{Bi}) \end{bmatrix} \begin{bmatrix} \dot{X}_{c_i} \\ \dot{Y}_{c_i} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -dy_i \\ 0 & 1 & dx_i \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} \dot{X}_{c_i} \\ \dot{Y}_{c_i} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{Bi}) & \sin(\theta_{Bi}) \\ -\sin(\theta_{Bi}) & \cos(\theta_{Bi}) \end{bmatrix} \begin{bmatrix} 1 & 0 & -dy_i \\ 0 & 1 & dx_i \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{6}$$

General wheel kinematics can be derived from equation 7.

$$\omega_i = \begin{bmatrix} \frac{1}{a_i} & \frac{1}{a_i} \tan(\phi_i) \end{bmatrix} \begin{bmatrix} \cos(\theta_{Bi}) & \sin(\theta_{Bi}) \\ -\sin(\theta_{Bi}) & \cos(\theta_{Bi}) \end{bmatrix} \begin{bmatrix} 1 & 0 & -dy_i \\ 0 & 1 & dx_i \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{7}$$

Using the parameters listed in Table 1 for the wheeled mobile robot, we apply the general wheel kinematics to derive the inverse kinematics. This derivation yields the mathematical relationship between the wheel velocities and the robot linear and angular Velocities.

$$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix}=\frac{1}{a}\begin{bmatrix}1&-1&-(d+L)\\1&1&-(d+L)\\1&-1&d+L\\1&1&d+L\end{bmatrix}\begin{bmatrix}u\\v\\r\end{bmatrix} \tag{8}$$

Table 1: kinematic parameter of the wheeled mobile robot

Wheel number	dx	dy	a	Θ	ϕ
1	L	d	a	0°	−45°
2	−L	d	a	0°	45°
3	−L	−d	a	0°	−45°
4	L	−d	a	0°	45°

Furthermore, from the derived inverse kinematics equation, we can obtain the forward kinematics.

$$\begin{bmatrix}u\\v\\r\end{bmatrix}=\xi=\frac{a}{4}\begin{bmatrix}1&1&1&1\\-1&1&-1&1\\\frac{-1}{d+L}&\frac{-1}{d+L}&\frac{1}{d+L}&\frac{1}{d+l}\end{bmatrix}\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix} \tag{9}$$

3. System Identification using Artificial neural network

System identification is the process by which a mathematical model for a dynamic system is developed through the analysis of observed data. In this domain, artificial neural networks (ANNs) have emerged as a powerful tool due to their capability to approximate complex nonlinear relationships and learn from data. In the context of robotics, the effectiveness of ANNs in learning intricate nonlinear dynamics from sensor data has been demonstrated, significantly enhancing system identification and control capabilities. By utilizing these networks, complex behaviors of robotic systems can be captured, allowing for more precise modeling and control. This capability is particularly crucial for applications such as the identification and management of the dynamics of wheeled mobile robots, where accurate modeling directly impacts the robot performance and maneuverability.

ANNs are structured with distinct layers: input, hidden, and output. Within each layer, fundamental mathematical operations are executed by each node. These operations involve the matrix multiplication of the input with associated weights, to which a bias term is added. Subsequently, a nonlinear activation function is applied to the result, a procedure known as forward propagation as shown in equation 10.

$$a^{(l+1)}=\sigma\left(W^{(l)}\cdot a^{(l)}+b^{(l)}\right) \tag{10}$$

where $a^{(l)}$ is the activation vector of the current layer, $a^{(l+1)}$ the activation vector of the next layer, $W^{(l)}$ the weight matrix of the current layer, σ is the nonlinear activation function, $b^{(l)}$ the bias vector of the current layer. Subsequently, the output of the final layer is compared with the actual output to compute the error, the mean square error of the last neuron given by.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{n} (y_i - a_i)^2 \tag{11}$$

where N is the number of data points, y_i is the target value of the data point i, a_i is the predicted value for data point i. This error is then backpropagated through the network, necessitating adjustments to the weights and biases across each layer to minimize the error based on the gradient function iteratively.

$$W^{(l)} := W^{(l)} - \alpha \cdot \frac{\partial MSE}{\partial W^{(l)}} \tag{12}$$

$$b^{(l)} := b^{(l)} - \alpha \cdot \frac{\partial MSE}{\partial b^{(l)}} \tag{13}$$

where α is the learning rate, $\frac{\partial MSE}{\partial W^{(l)}}$ and $\frac{\partial MSE}{\partial b^{(l)}}$ are the partial derivatives of the error function with respect to the weights and biases, respectively. The partial derivatives can be calculated using the chain rule.

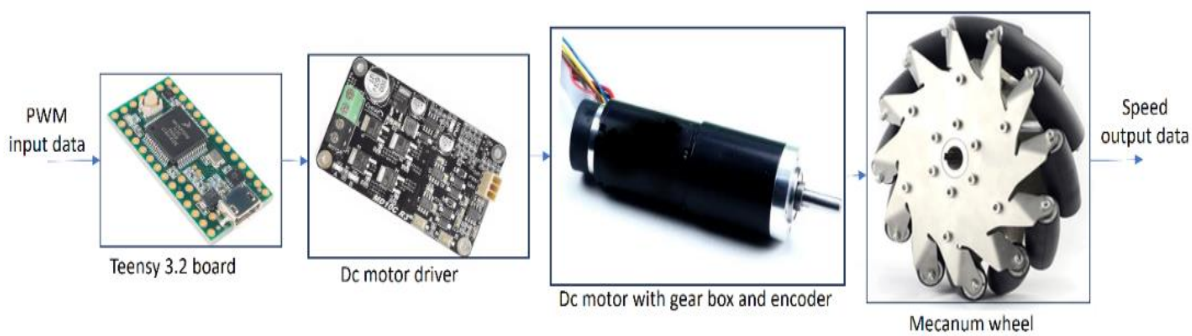

Fig. 3: system identification setup for a wheeled mobile robot

$$\frac{\partial\, MSE}{\partial\, W^{(l)}} = \left(\frac{\partial\, MSE}{\partial\, a^{(l+1)}}\right) \cdot \left(\frac{\partial\, a^{(l+1)}}{\partial\, W^{(l)}}\right) \tag{14}$$

$$\frac{\partial\, MSE}{\partial\, b^{(l)}} = \left(\frac{\partial\, MSE}{\partial\, a^{(l+1)}}\right) \cdot \left(\frac{\partial\, a^{(l+1)}}{\partial\, b^{(l)}}\right) \tag{15}$$

where $\left(\frac{\partial\, MSE}{\partial\, a^{(l+1)}}\right)$ can be calculated using mean square error and $\left(\frac{\partial\, MSE}{\partial\, a^{(l+1)}}\right)$ can be calculated using the activation function and forward propagation.

The process of system identification entails data collection from the real system, as shown in Fig. 3. The system comprises pulse-width modulation (PWM) signals as a sine wave with different frequencies and amplitude from -255 to 255 generated by the teensy 3.2 microcontroller. At the same time, the output is represented by the velocity (RPM) of the DC motor measured using the encoder attached to the shaft of the DC motor sampled at 10 MS, as shown in Fig. 4.

Following data collection, preprocessing steps are implemented, which involve removing any outliers present in the dataset. Subsequently, the dataset is split into training and testing subsets. Typically, the testing data constitutes 10% of the entire dataset, while the remaining portion is allocated to training. Within the training subset, the data is further divided into batches, with each batch containing 16 samples. This batching process aids in enhancing computational efficiency during the training phase and facilitates better convergence of the model. The data consists of 5959 data points, the artificial neural network (ANN) architecture comprises an input layer with five inputs, namely $u(k), u(k-1)$, $\mathrm{u(k-2)}, \mathrm{y(k-1)}$, and $\mathrm{y(k-2)}$. This is followed by two hidden layers, each containing ten neurons. Finally, the network concludes with an output layer consisting of a single neuron representing the output $\mathrm{y(k)}$ as shown in figure 5, Input Layer $\in \mathbb{R}^5$ Hidden Layer $\in \mathbb{R}^{10}$ Hidden Layer $\in \mathbb{R}^{10}$ Output Layer $\in \mathbb{R}^1$.

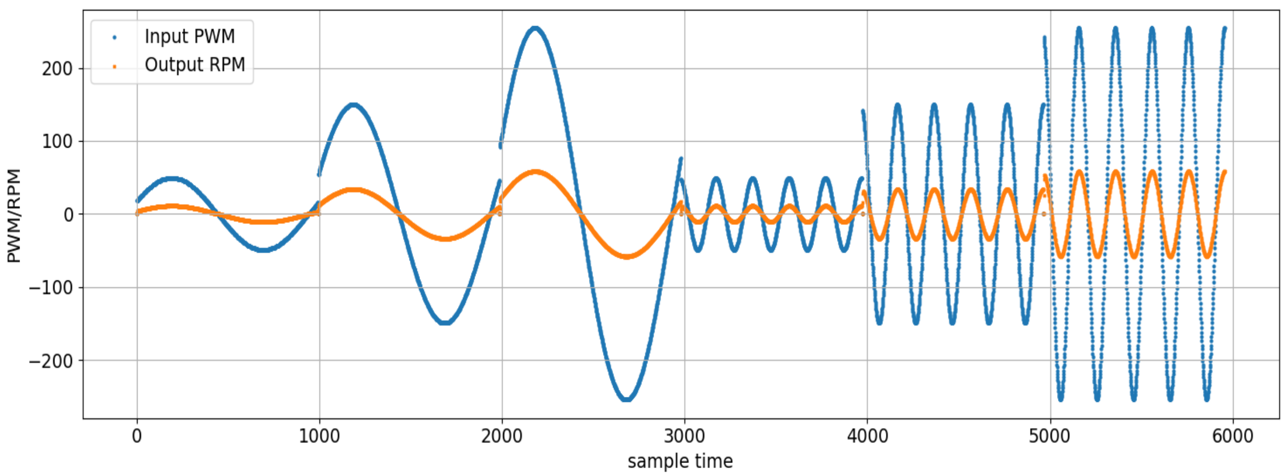

Fig. 4: Input-Output sine wave for the system

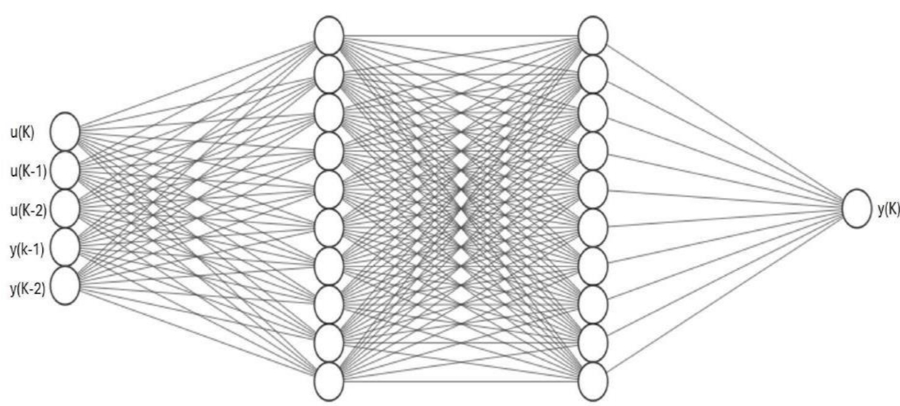

Fig. 5: structure of the ANN

The annotation model was trained for 5 epochs using the Mean Square Error loss function and the Adam optimization algorithm. A learning rate of 0.001 was employed, with Beta1 and Beta2 values set to 0.9 and 0.999, respectively. An epsilon value of 1e-8 was utilized. The hidden layers employed the ReLU activation function, while no activation function was applied to the output layer. During each iteration, the losses for individual batches were computed, and the mean loss was derived. This methodology was employed to monitor the advancement of training and ascertain the model learning efficacy. Furthermore, the evaluation of test data loss was conducted to appraise the model performance on previously unseen data, serving to mitigate the risk of overfitting, as shown in Figure 6. The model exhibited a notable alignment with the dataset, reflected in an R2 score of 99.98%. This high R2 score indicates that the model proficiently accounted for nearly all variability within the dataset.

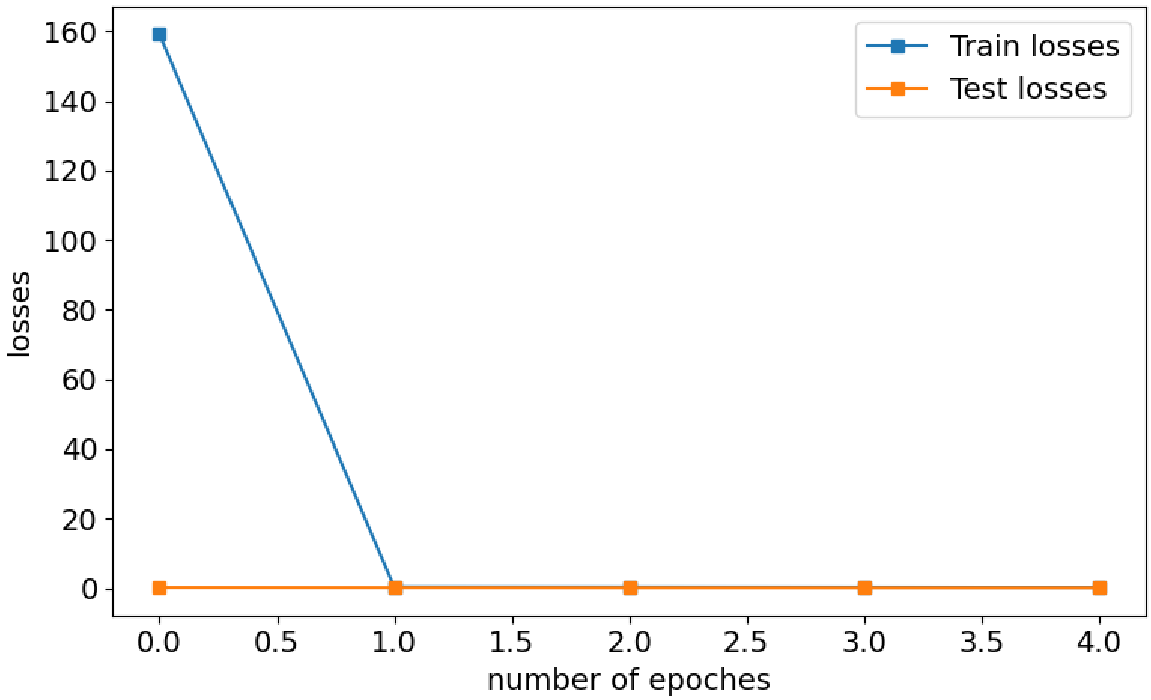

Fig. 6: the losses of train test data at each epoch of training

4. System control using Artificial neural network

A PID controller is a type of feedback controller that utilizes three constants ($Kp, Ki,$ and Kd) to adjust the output based on the error between the desired and actual outputs. These constants must be carefully tuned to achieve optimal performance. In contrast, an ANN-based controller does not require manual tuning of constants. Instead, the ANN learns the relationship between the inputs and outputs of the system through training data, allowing it to automatically adjust its weights and biases to achieve optimal control performance. The system will consist of two ANNs: one dedicated to controlling the system and the other for simulating its behavior. The control ANN can be trained to learn the optimal control strategy based on data generated by the simulation ANN. This setup allows the control ANN to adapt to changes in the system dynamics and environment, leading to improved control performance, as shown in Fig. 7.

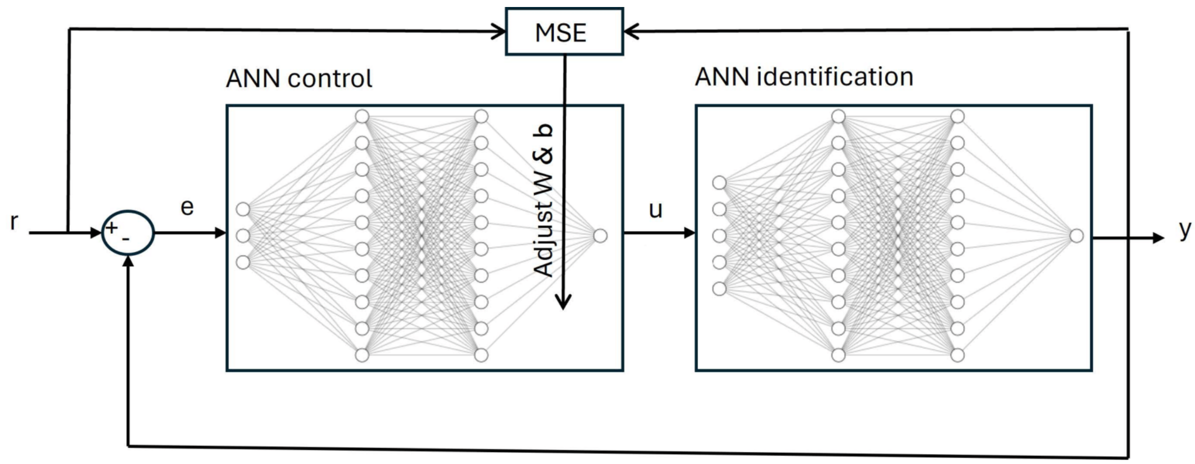

Figure 7: control system using ANN

The control mechanism is implemented via an artificial neural network (ANN) with an architecture consisting of an input layer with 3 neurons representing the error, integral of past errors, and derivative of error; two hidden layers, each containing 10 neurons; and an output layer with 1 neuron. The ANN is employed to control a system by adjusting its inputs based on the error between the desired and actual outputs. Information about the state of error of the system is provided to the ANN through the input layer, while the hidden layers are responsible for learning the relationship between the inputs and outputs. The control signal, used to adjust the system input, is generated by the output layer. The control model was trained for 2 epochs using the Mean Square Error loss function and the Adam optimization algorithm. A learning rate of 0.001 was set, with Beta1 and Beta2 values of 0.9 and 0.999, respectively. An epsilon value of $1X10^{-8}$ was utilized. The ReLU activation function was used in the hidden layers, while no activation function was applied to the output layer.

Initially, the ANN is deployed to control the system with randomly initialized weights and biases. Consequently, the system output may not accurately track the desired reference signal, resulting in errors in the control process. However, through the backpropagation process, the ANN internal parameters (weights and biases) are adjusted based on the errors observed between the system output and the reference signal, as shown in Fig. 8. As more training data is fed into the ANN and the backpropagation process continues, the network gradually learns to minimize these errors, as illustrated in Fig. 9.

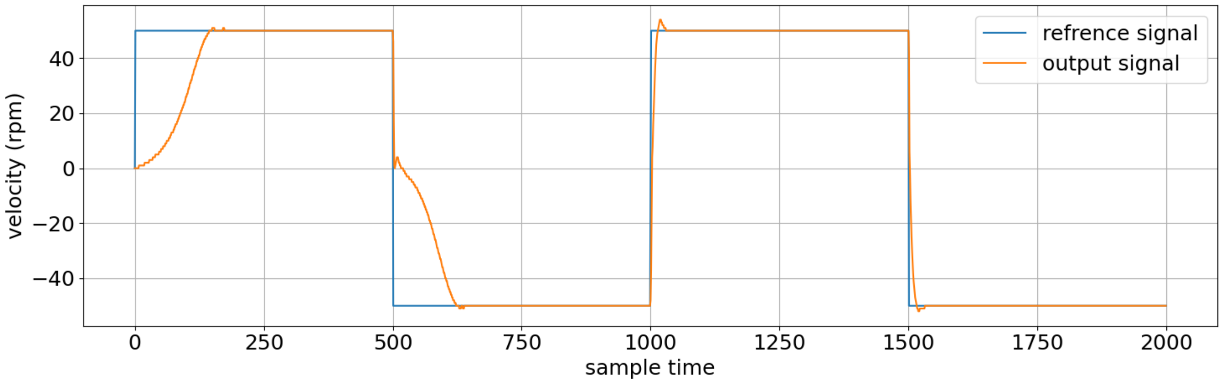

Fig. 8: ANN training

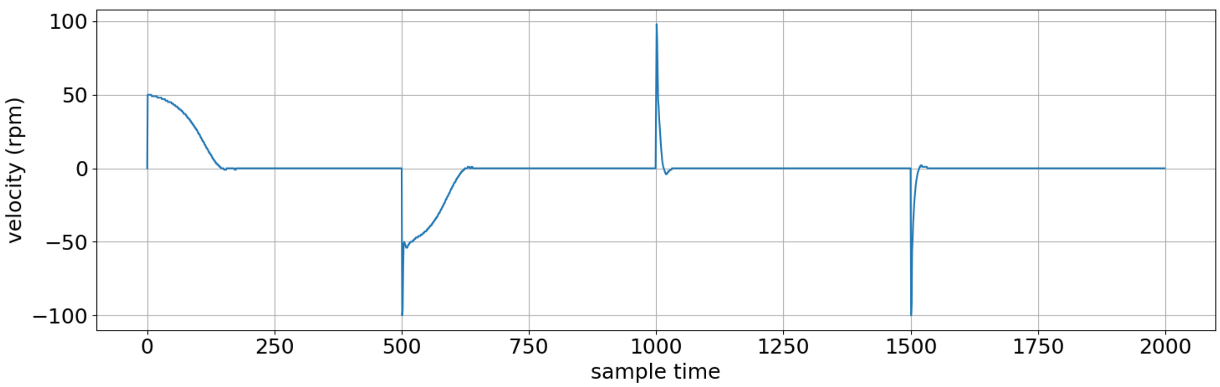

Fig. 9: ANN control losses

5. Results

This section provides a comprehensive summary of the results obtained from the trajectory tracking experiments performed with the wheeled mobile robot. The experiments involved executing a figure-eight (∞) trajectory, as depicted in Fig. 10. The primary objective was to evaluate the robot ability to follow this complex path accurately. During the experiments, the robot positional deviations along the x and y axes were meticulously recorded at various time intervals, as shown in Fig. 11. These deviations were analyzed to determine the precision of the trajectory tracking and to identify any significant errors or drifts from the intended path.

The evaluation focused on quantifying the difference between the desired trajectory and the actual path taken by the robot. This involved calculating the root mean square error (RMSE) for both the x and y coordinates to provide a numerical measure of the tracking accuracy. Additionally, the stability of the control system was assessed by examining the consistency of the robot path over multiple trials. The results highlighted the performance of the control algorithms in handling the dynamic aspects of the robot movement, offering insights into areas where further optimization might be needed. Furthermore, the impact of different control parameters on the tracking performance was explored. Adjustments to the control gains and feedback mechanisms were made to observe their effects on the robot ability to maintain the desired trajectory. These experiments provided valuable data on the robustness and adaptability of the control system under various conditions, contributing to a deeper understanding of the system capabilities and limitations.

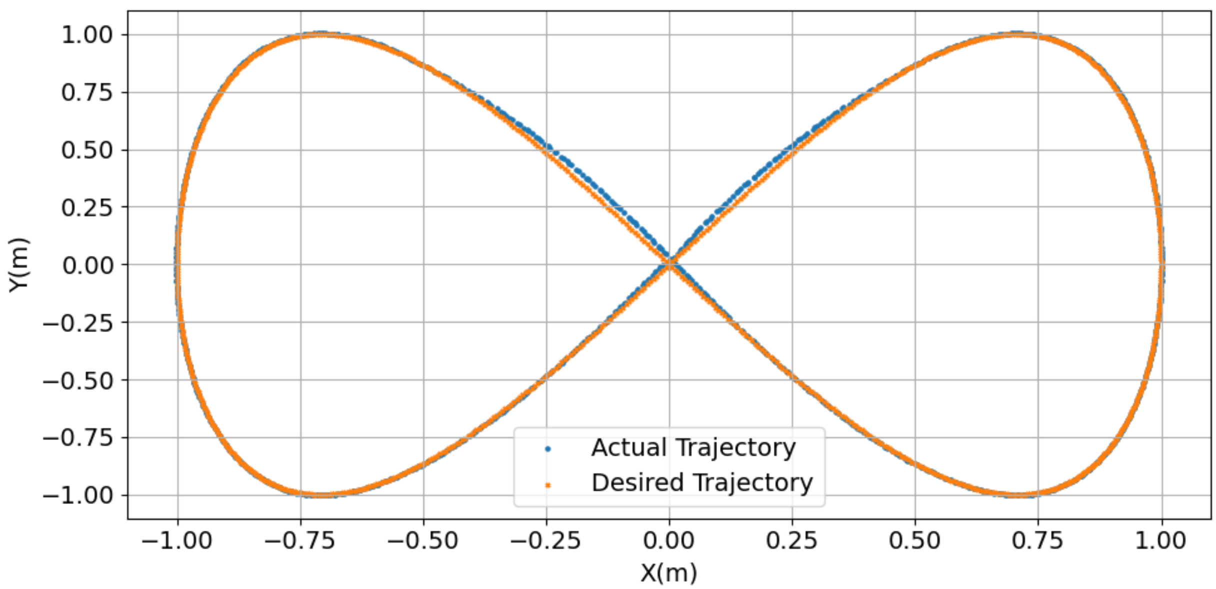

Fig. 10: actual and desired trajectory of inf path

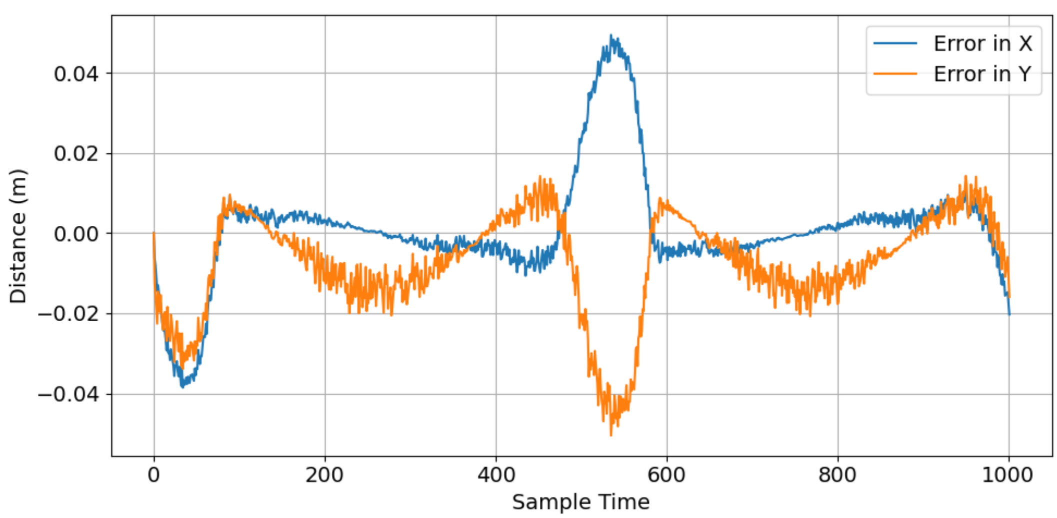

Fig. 11: error in x and y between desired and actual trajectory

6. Conclusions

The research demonstrated a highly effective trajectory-tracking control strategy for a four-wheeled mecanum robot, achieving precise trajectory tracking with a mean square error of 0.0005 meters using Deep Neural Networks (DNNs) for both system identification and control. By training on extensive input-output data, the DNN captured the complex nonlinear dynamics of the robot, creating a highly accurate dynamic model. A separate DNN-based controller was then designed to generate precise control signals, leveraging insights from the dynamic model. Experimental validation using an infinity-shaped trajectory showed the controller remarkable accuracy. These findings highlight the potential of deep learning techniques for addressing complex control challenges in wheeled mobile robots. The study suggests that DNNs can provide robust solutions for trajectory tracking in industrial applications, and future research could explore scalability to other robot types and optimize neural network architectures for improved real-time performance. In conclusion, this research successfully demonstrated that DNNs could enhance precise trajectory tracking in wheeled mobile robots, laying the groundwork for future advancements in robotic control systems.

References

[1] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Substantial capabilities of robotics in enhancing industry 4.0 implementation," Cognitive Robotics, vol. 1, pp. 58–75, 2021, doi: https://doi.org/10.1016/j.cogr.2021.06.001.

[2] Z. Fan, Q. Qiu, and Z. Meng, "Implementation of a four-wheel drive agricultural mobile robot for crop/soil information collection on the open field," in 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), May 2017, pp. 408–412. doi: 10.1109/YAC.2017.7967443.

[3] G. Ciubucciu, R. Solea, A. Filipescu, and A. Filipescu, "Visual servoing and obstacle avoidance method based control autonomous robotic systems servicing a mechatronics manufacturing line," in 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Sep. 2017, pp. 874–879. doi: 10.1109/IDAACS.2017.8095212.

[4] C. Özbaran, S. Dilibal, and G. Sungur, "Mechatronic System Design of A Smart Mobile Warehouse Robot for Automated Storage/Retrieval Systems," in 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Oct. 2020, pp. 1–6. doi: 10.1109/ASYU50717.2020.9259882.

[5] M. Abdelrahim, A. Hassan, D. Ojo, M. Hosny, H. H. Ammar and M. El-Samanty, "A Novel Design of a T-Model Three Mecanum Wheeled Mobile Robot," 2022 International Conference on Advanced Robotics and Mechatronics (ICARM), Guilin, China, 2022, pp. 509-513, doi: 10.1109/ICARM54641.2022.9959189.

[6] M. H. Barakat, H. H. Ammar and M. Elsamanty, "Experimental Path tracking optimization and control of a nonlinear skid steering tracked mobile robot," 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2020, pp. 434-438, doi: 10.1109/NILES50944.2020.9257956.

[7] R. A. Deshmukh and M. A. Hasamnis, "A navigation scheme for autonomous mobile service robots working in GPS denied commercial indoor spaces," in 2023 International Conference on Communication, Circuits, and Systems (IC3S), May 2023, pp. 1–5. doi: 10.1109/IC3S57698.2023.10169514.

[8] H. Lei, X. Canzhen, H. Fuchun, F. Chunming, and H. Shaoyong, "Design and analysis of omnidirectional mobile material handling robots that can adapt to complex environments," in 2023 IEEE

3rd International Conference on Electronic Technology, Communication and Information (ICETCI), May 2023, pp. 1606–1615. doi: 10.1109/ICETCI57876.2023.10176548.

[9] M. Elsamanty, M. Fanni and A. Ramadan, "Novel hybrid ground/aerial autonomous robot," 2012 First International Conference on Innovative Engineering Systems, Alexandria, Egypt, 2012, pp. 103-108, doi: 10.1109/ICIES.2012.6530853.

[10] M. Elsamanty, A. Khalifa, M. Fanni, A. Ramadan and A. Abo-Ismail, "Methodology for identifying quadrotor parameters, attitude estimation and control," 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, NSW, Australia, 2013, pp. 1343-1348, doi: 10.1109/AIM.2013.6584281.

[11] M. Elsamanty, M. Fanni, A. Ramadan and A. Abo-Ismail, "Modeling and control of a novel Hybrid Ground Aerial Robot," 2013 IEEE International Conference on Mechatronics and Automation, Takamatsu, Japan, 2013, pp. 1559-1565, doi: 10.1109/ICMA.2013.6618146.

[12] G. S. Rivera, N. D. Muñoz Ceballos, and H. M. Vásquez Carvajal, "Methodology for Identification, Modeling and Control of a Robot Locomotion Motors," in 2020 IX International Congress of Mechatronics Engineering and Automation (CIIMA), Nov. 2020, pp. 1–6. doi: 10.1109/CIIMA50553.2020.9290301.

[13] D. F. Z. Romero, A. S. Vélez, and J. B. Gómez-Mendoza, "Non-linear Grey-Box Models Applied to DC Motor Identification," in 2019 IEEE 4th Colombian Conference on Automatic Control (CCAC), Oct. 2019, pp. 1–5. doi: 10.1109/CCAC.2019.8921181.

[14] M. Shehata, M. Elshami, Q. Bai, and X. Zhao, "Parameter Estimation for Multibody System Dynamic Model of Delta Robot From Experimental Data," IFAC-PapersOnLine, vol. 54, no. 14, pp. 72–77, 2021, doi: https://doi.org/10.1016/j.ifacol.2021.10.331.

[15] Z.-J. Fu, W.-F. Xie, X. Han, and W.-D. Luo, "Nonlinear Systems Identification and Control Via Dynamic Multitime Scales Neural Networks," IEEE Trans Neural Netw Learn Syst, vol. 24, no. 11, pp. 1814–1823, Nov. 2013, doi: 10.1109/TNNLS.2013.2265604.

[16] C. Liang and S. Lijun, "Research on Formation Pressure Prediction Based on Neural Network System Identification Theory," in 2020 2nd International Conference on Information Technology and Computer Application (ITCA), Dec. 2020, pp. 702–705. doi: 10.1109/ITCA52113.2020.00151.

[17] P. Gautam, "System identification of nonlinear Inverted Pendulum using artificial neural network," in 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Dec. 2016, pp. 1–5. doi: 10.1109/ICRAIE.2016.7939522.

[18] A. A. Alphonse, A. A. Abbas, A. M. Fathy, N. S. Elsayed, H. H. Ammar and M. M. Elsamanty, "Modelling of Continuum Robotic Arm Using Artificial Neural Network (ANN)," 2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2019, pp. 191-195, doi: 10.1109/NILES.2019.8909308.

[19] H. Shu, X. Wang, and Z. Huang, "Identification of multivariate system based on PID neural network," in 2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP), Nov. 2015, pp. 199–202. doi: 10.1109/ICICIP.2015.7388168.

[20] M. Schüssler, T. Münker, and O. Nelles, "Deep Recurrent Neural Networks for Nonlinear System Identification," in 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Dec. 2019, pp.

448–454. doi: 10.1109/SSCI44817.2019.9003133.

[21] G. D. Şen, G. Ö. Günel, and M. Güzelkaya, "Extended Kalman Filter Based Modified Elman-Jordan Neural Network for Control and Identification of Nonlinear Systems," in 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Oct. 2020, pp. 1–6. doi: 10.1109/ASYU50717.2020.9259812.

[22] N. Tan, Z. Zhu, and P. Yu, "Neural-Network-Based Control of Wheeled Mobile Manipulators with Unknown Kinematic Models," in *2020 International Symposium on Autonomous Systems, ISAS 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 212–216. doi: 10.1109/ISAS49493.2020.9378850.

[23] P. Bozek, Y. L. Karavaev, A. A. Ardentov, and K. S. Yefremov, "Neural network control of a wheeled mobile robot based on optimal trajectories," *Int J Adv Robot Syst*, vol. 17, no. 2, Mar. 2020, doi: 10.1177/1729881420916077.

[24] N. Hassan and A. Saleem, "Neural Network-Based Adaptive Controller for Trajectory Tracking of Wheeled Mobile Robots," *IEEE Access*, vol. 10, pp. 13582–13597, 2022, doi: 10.1109/ACCESS.2022.3146970.

[25] Ly, T. T. K., Thanh, N. T., Thien, H., & Nguyen, T. H. (2023). A neural network controller design for the mecanum wheel mobile robot. Engineering, Technology &Amp; Applied Science Research, 13(2), 10541-10547. https://doi.org/10.48084/etasr.5761

[26] Li, Y., Ge, S., Dai, S., Zhao, L., Yan, X., Zheng, Y., … & Shi, Y. (2019). Kinematic modeling of a combined system of multiple mecanum-wheeled robots with velocity compensation. Sensors, 20(1), 75. https://doi.org/10.3390/s20010075

[27] Zheng, L. (2022). Predictive control of the mobile robot under the deep long-short term memory neural network model. Computational Intelligence and Neuroscience, 2022, 1-12. https://doi.org/10.1155/2022/1835798

[28] Wang, Y., Liu, X., Yang, X., Hu, F., Jiang, A., & Yang, C. (2018). Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy. Applied Sciences, 8(2), 231. https://doi.org/10.3390/app8020231

[29] AYEB*, A. and CHATTI, A. (2019). Indirect neural adaptive control for wheeled mobile robot. International Journal of Innovative Technology and Exploring Engineering, 9(1), 2138-2145. https://doi.org/10.35940/ijitee.a4692.119119

[30] Trujillo, D., Morales, L. A., Chávez, D., & Pozo, D. F. (2023). Trajectory tracking control of a mobile robot using neural networks. Emerging Science Journal, 7(6), 1843-1862. https://doi.org/10.28991/esj-2023-07-06-01