Journal of Robotics and Control (JRC)
Volume 5, Issue 4, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i4.21650

Sliding Mode Control Based on Neural State and Disturbance Observers: Application to a Unicycle Robot Using ROS2

Barhoumi Nawress [1*], Asma Najet Lakhal Gharbi [2], Naceur Benhadj Braiek [3]
[1,2,3] Advanced Systems Laboratory, Polytechnic School of Tunisia, University of Carthage Tunisia
Email: [1] nawress.barhoumi@enicar.ucar.tn, [2] asmanajet.lakehal@fsb.ucar.tn, [3] naceurbhb@gmail.com
*Corresponding Author

***Abstract*—The major problem dealing with mobile robots is the trajectory tracking control problem, in the presence of random disturbance and unmeasurable angular velocity. In this paper, we propose a Sliding Mode Control (SMC) based on a Nonlinear Disturbance Observer (NDO) and a Neural State Observer (NSO). The (SMC-NDO) controller displays limitations in mitigating external disturbances. Therefore, this research contribution suggests a novel approach that integrates a Neural State Observer (NSO) into the (SMC-NDO) controller, to significantly enhance the performance of a control system. The combined approach improves disturbance reduction while simultaneously estimating the unmeasurable angular velocity, ultimately leading to more accurate path tracking. Furthermore, the Lyapunov method is used to ensure the stability of the closed-loop control on the one hand, and the stability of the Neural State Observer based on the Back-propagation algorithm on the other hand. Numerical simulations and the implementation of the Simulator in ROS/Gazebo demonstrate better performance of our proposed approach (SMC-NSO-NDO) compared to the Sliding Mode control-based Disturbance Observer (SMC-NDO) and the Sliding Mode Control (SMC). The control proposal in this work is ready for use on most ROS-compatible robots. This experiment should offer an enlightening perspective to robotics researchers.**

Keywords*—*Unicycle Mobile Robot, Sliding Mode Control, Neural State Observer, Disturbance Observer, Robot Operating System ROS.

I. Introduction

In the era of Industry 4.0, mobile robots play a crucial role as they perform various difficult and dangerous tasks on behalf of humans. Therefore, scientists are actively researching effective solutions to problems that may occur in robotics [61], [28], [36], [22], [49], [16], [40], [82], [25], [12], [78], [31], [3], [1].

The 5 most important types of robots in 2023, according to the International Federation of Robotics report, are Mobile cobots [43], mobile robots [35], [55], manipulators [70], Digital Twins [44], and Humanoid Robots [6], [67]. These robots integrate not only Robot Operating System (ROS) technologies for efficient operations [10], [38], [2], [58], [66], [77], [11], [52], [14] and [46] but also, the artificial intelligence (AI) technology, [18], [48], [83], [84], [68], [47], [73] and [33].

Thanks to their advantages, unicycle mobile robots are becoming increasingly popular. For instance, their small size, suitable price, rapidity, and light help avoid traffic jams. Mobile robots have been used to perform specialized tasks in several industries, including services, rescue, military, and disaster relief [32]. However, they are classified as non-holonomic mobiles due to wheel limitations [50], [49], [5], [37], [30], [74], [72] and [26].

In addition, autonomy is a crucial aspect of robots. Thus, researchers are actively working on developing methods in order to enable robots to independently perform tasks without human intervention in different areas, such as perception, decision-making, and control. The primary objective consists of creating robots that can operate effectively and efficiently in real-world environments, [69], [9], and [27].

The choice of the control technique is based on the non-linearity and the existence of internal and external disturbances. On the other hand, several physical constraints such as the uncertainty of the parameters, the errors in the modeling of the physical systems to be controlled, and the imperfection of the measuring instruments (hardware).

The studies [8], [62], [54], and [59] investigate PID control that has been widely used in industry and its application to robotics. However, the PID control system is inefficient when some variables are modified or when there is a change in torque burden, resulting in insufficient responses.

[45] presented a fuzzy flatness technique, where the benefits of the regulator are adjusted online using a suitable fuzzy controller, for handling Pioneer 3dx mobile robots efficiently. In [70], backstopping controllers were implemented on the ground Robot. [34] tested the effectiveness of the (SMC) based on a new navigation planning algorithm that uses a delineation polygon for a mobile robot. To reduce the chattering effect on UMR, the SMC controller is used in [50]. However, in these previous works, this technique does not address powerful white noise disturbances.

In [19], an SMC controller combining a neural network is applied in a mobile robot. However, it is necessary to adjust the neural network parameters using stability theory to determine the update rate that affects the controller's performance. In [42], the SMC control is based on an extended state observer to handle the shortcoming of undesired chattering. However, this technique did not achieve the stability of the estimator based on the neural network

An adaptive control law has been proposed in our case, which is based on the sliding mode control approach. The proposed solution takes into consideration the uncertainties in the parameters of the robot, such as platform mass, inertia, and wheel diameter. The robustness is addressed by this approach when considering external disturbances [29], [51], [17], [23]. The Lyapunov direct method guarantees global asymptotic stability, one of the major advantages of the proposed SMC approach. Thus, the Lyapunov analysis's equality ensured the system stability through adaptive gains obtained by YALMIP solver method. However, the presence of the 'chattering' event caused by the discontinuous part of this control law generates a negative feedback loop. This creates undesirable effects on the studied system.

Hence, previous studies, [13], [60], [50], [39], [24], [80], [81], and [63] delved into the utilization of a disturbance observer in (SMC) to mitigate the impact of weak-amplitude colored external disturbance. Despite its effectiveness in reducing certain disturbances, the sliding mode control based on a disturbance observer is not only insufficient to handle forced random disturbances but also unable to estimate the state vector. Traditional disturbance observer-based control technology is insufficient to enhance the tracking performance of systems [21].

The presence of powerful external white noise disturbance in robotic systems is characterized by random character and uniform intensity at all frequencies. Often, external disturbance cannot be directly measured [76], [53], and [56]. These disturbances significantly reduce achievable performance in terms of tracking accuracy.

UMR issues manifest through unmeasurable angular velocity and unpredictable, potent external disturbances. These factors significantly impact robot control, potentially leading to severe consequences ranging from operational malfunctions to hazardous incidents like rollovers. Most of these robots present nonholonomic constraints [50]. Our problem formulation becomes more complicated due to that.

Therefore, we need a filter to reject disturbance and accurately estimate velocities in dynamic environments. The integration of a Neural State Observer (NSO) is required.

In this context, a Neural State Observer is employed to achieve two main objectives: (1) to accurately estimate the robot's angular velocity in order to realize precise path tracking, and (2) to enhance control power and thus, improve performance. The observer incorporates an artificial neural network (ANN). Thanks to its ability to model and identify complex and highly non-linear dynamic systems, an artificial neural network is a good choice to synthesize the state-variable observer of the unicycle robot. Furthermore, the NSO contains an observation matrix that is adaptively calculated by using the Lyapunov theory based on the backpropagation algorithm. Its characteristics significantly affect the performance and accuracy of the angular velocity state estimation process.

We presented the role of each technique in the proposed controller:

- Despite its robustness, Sliding Mode Control may not accurately estimate or compensate for powerful disturbances.
- The disturbance observer: The sliding controller helps improve performance against disturbances. However, it is limited in the face of random disturbances, and it cannot estimate the robot's velocity.
- The Neural State Observer: The proposed Neural State Observer is applied to improve the reconstruction of the angular velocity, which may not always be available, and help reject disturbances in the controller.

Motivated by the above-reviewed papers and in order to strengthen the arsenal of the existing techniques, we propose in this work a new SMC-NDO-NSO approach applied to the UMR robot. The major contributions of the proposed SMC-NDO-NSO are detailed as follows:

1) Achieve accurate trajectory by estimating velocities on UMR and reject the difficulties imposed by powerful white noise disturbances.
2) The comparative analysis of the SMC-NDO-NSO, SMC-NDO, and SMC controllers shows that the SMC-NDO-NSO model is characterized by its ability to effectively maintain the stability of the robot, demonstrating its flexibility.
3) In order to test the SMC-NDO-NSO controller, we used the Gazebo simulator environment to execute it on the Minilab robot and Husky_robot, which was both effective and performed well.

The remainder of this paper is structured as follows: Section II describes the kinematic and dynamic models of UMR. Section III introduces the adaptive Sliding Mode Control based on disturbance observer. Section IV presents the Neural State Observer based on the backpropagation algorithm, with a special focus on its stability. Then, section V provides details about SMC integrated with the Neural State Observer and Disturbance Observer for the stability analysis of closed-loop control. Finally, VI summarizes the key findings of simulation via MATLAB-SIMULINK and ROS and highlights major directions for future research.

II. Unicycle Robot Model

The model used in this work describes UMR as a two-wheeled non-holonomic mobile robot that can rotate around its axis. Each wheel is controlled independently see Fig. 1

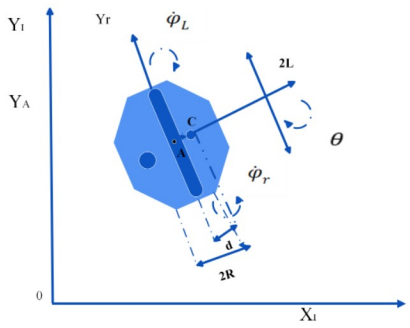

Fig. 1. Unicycle Mobile Robot

A. Kinematic Modeling

Kinematic modeling provides a geometric representation of the movement of the unicycle robot, while kinematic control intervenes to regulate this movement by adjusting the kinematic parameters.

The robot's kinematic model is given by [41].

$$\begin{cases} \dot{x} = v.\cos\theta \\ \dot{y} = \Omega.\sin\theta \\ \dot{\theta} = \Omega \end{cases} \tag{1}$$

with P = (x,y,θ) is the robot position and orientation in the world reference frame, and the pair (v,Ω) is the linear and angular velocities.

The kinematic controller used is the one proposed in [41] and is given by

$$\begin{pmatrix} v_c \\ \Omega_c \end{pmatrix} = \begin{cases} v_r\cos\theta_e + k_x x_e \\ \Omega_r + v_r\left(k_y y_e + k_\theta \sin\theta_e\right) \end{cases} \tag{2}$$

with

- $X_c = \begin{pmatrix} v_c \\ \Omega_c \end{pmatrix}$: the velocity of the kinematic controller;
- $x_e = (\theta)(x_r - x)$, $y_e = (\theta)(y_r - y)$, $\theta_e = (\theta)(\theta_r - \theta)$: the current position errors in the axes X ,Y and θ.

The value of the stability gains $[k_x, k_y, k_\theta]$ in equation (2) is fixed on by demonstrating the stability of the kinematic loop. Indeed, the positive definite Lyapunov function is given

$$V_0 = \frac{1}{2}\left(x_e^2 + y_e^2\right) + \frac{1}{k_y}\left(1 - \cos\theta_e\right) \tag{3}$$

The temporal derivation of V_0 along the trajectory is

$$\dot{V_0} = \dot{x_e}\,x_e + \dot{y_e}\,y_e + \frac{1}{k_y}\left(\dot{\theta_e}\sin\theta_e\right) \tag{4}$$

The tracking error dynamics can be calculated as

$$\begin{pmatrix} \dot{x_e} \\ \dot{y_e} \\ \dot{\theta_e} \end{pmatrix} = \begin{pmatrix} y_e\Omega - v + v_r\cos\theta_e \\ -x_e\Omega + \dot{v}_r\sin\theta_e \\ \Omega_r - \Omega \end{pmatrix} \tag{5}$$

Thus, the following is the final result of the time derivative of V_0 equation (4) along the trajectory

$$\dot{V_0} = -k_x {x_e}^2 + \frac{v_r k_\theta \sin^2\theta_e}{k_y} \tag{6}$$

The parameters k_x, k_y, k_θ must be positive to satisfy the Lyapunov stability for kinematic control.

B. Dynamic Modeling

The dynamic model is required for simulation, robot motion analysis, and the conception of various control algorithms.

From [13], assuming that the disturbance term is a non-zero vector, the nonlinear dynamic modeling is defined by the following

$$\begin{cases} \dot{v}(t) = \frac{m_c a}{m_0}\Omega^2 + \frac{1}{m_0 R}u_1(t) + d_v(t) \\ \dot{\Omega}(t) = \frac{-m_c a}{I_0}v.\Omega + \frac{L}{I_0 R}u_2(t) + d_w(t) \end{cases} \tag{7}$$

with

- $\mathrm{X} = \begin{pmatrix} v \\ \Omega \end{pmatrix}$: the state vector consists of two components: the linear velocity and the angular velocity;
- $u = \begin{cases} u_1 = \tau_R + \tau_L \\ u_2 = \tau_R - \tau_L \end{cases}$ with τ_R:the right wheel torque control; τ_L:the left wheel torque control;
- $d(t) = \begin{pmatrix} d_v \\ d_w \end{pmatrix}$: white noise disturbance;
- $m_0 = \left(m + \frac{2I_\Omega}{R^2}\right)$: the equivalent mass;
- $I_0 = \left(I + \frac{2L^2}{R^2}I_\Omega\right)$: the equivalent inertia of UMR.

The equation (7) of the system can be formulated as such in the following non-linear form

$$\dot{\mathrm{X}}(t) = F(\mathrm{X}(t), u(t)) = f(\mathrm{X}) + g_1(\mathrm{X}).u(t) + g_2(\mathrm{X})\,d(t) \tag{8}$$

with

- $f(\mathrm{X}) = \begin{bmatrix} \frac{m_c a}{m_0}\Omega^2 \\ \frac{-m_c a}{I_0}v.\Omega \end{bmatrix}$: a non-linear function;
- $g_1 = \begin{bmatrix} \frac{1}{m_0 R} & 0 \\ 0 & \frac{L}{I_0 R} \end{bmatrix}$: an invertible function;
- $g_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

The parameters included in vector X are functions of some physical parameters of the robot, such as

- m_c: the mass of the platform;
- m_Ω: the wheel mass;
- m: the total mass of the robot;
- I_Ω: the inertia concerning the wheel axis;
- R: the diameter of the robot wheel;
- L: the mid-distance between the two wheels;
- a: the distance between the midpoint of the two wheels A and the center of gravity C.

where the parameter's value is given in Table II.

Challenges, associated with the nonlinear dynamic model of UMR including the coupling between movements of linear velocity and angular velocity, are interdependent. Thus, modeling and control become more complex while introducing uncertainties, high noise levels, and unknown parameters. That is why powerful and intelligent control is needed to solve these problems effectively.

III. Adaptive Sliding Mode Control Based on Disturbance Observer

Sliding mode control (SMC) is used in systems with disturbances and uncertainties. It employs adaptive gains that can adjust to changing robot dynamics and environmental conditions for better controller performance. Lyapunov-based sliding mode control ensures robot velocity convergence to kinematic control commands.

A. Adaptive sliding mode control

In this paper, a Sliding Mode Control will be proposed for a UMR. The sliding surface is defined by

$$S(t) = \begin{bmatrix} S_1(t) \\ S_2(t) \end{bmatrix} = e(t) + \beta \int e(t)\, dt \tag{9}$$

- $e(t) = [e_v, e_\Omega]^T = \mathrm{X}(t) - X_c(t)$: the error models;
- $\beta > 0$: positive value.

If the derivative of the sliding surface $\dot{S}(t) = 0$, the following equation is obtained

$$\dot{S}(t) = \dot{e}(t) + \beta e(t) = 0 \tag{10}$$

Starting from (8), we can rewrite the sliding surface derivative (10) as

$$\left(f(\mathrm{X}) + g_1(\mathrm{X}).u(t) + g_2(\mathrm{X}).d(t) - \dot{X}_c\right) + \beta.e = 0 \tag{11}$$

The sliding mode control taking into account disturbance is designed as

$$u(t) = g_1^{-1} \begin{pmatrix} \dot{X}_c - \beta e(t) - f(\mathrm{X}) - g_2(\mathrm{X}).d(t) - \\ KTanh(S) - \eta S \end{pmatrix} \tag{12}$$

with

- $K = \begin{pmatrix} k_a \\ k_b \end{pmatrix}$: are gains to ensure stability of SMC control;
- $\eta S(t)$: is feedback control with $\eta > 0$;
- $KTanh(S)$: ensure system robustness.

Equation (8) includes unwanted white noise disturbances $\begin{bmatrix} d_1 & d_2 \end{bmatrix}^T$ represents heavy robot loads. These disturbances degrade the tracking performance and the accuracy of the sliding mode controller (12) as well as increase the system error or deviation from the intended path. To address this issue, advanced strategies are needed to estimate and mitigate the stochastic nature of noise.

B. Adaptive sliding mode control with disturbance observer

The NDO design technique of the system (8) in [15] is reviewed in this section. The disturbance d is an unknown-bounded constant, it follows that

$$\dot{d} = 0 \tag{13}$$

given that equation (8) can be written

$$g_2(\mathrm{X})\, d(t) = \dot{\mathrm{X}}(t) - f(\mathrm{X}) - g_1(\mathrm{X})\, u(t) \tag{14}$$

the initial disturbance observer is suggested in [15] as

$$\dot{\hat{d}} = -l(\mathrm{X})\, g_2(\mathrm{X})\, \hat{d} + l(\mathrm{X}).\left(\dot{\mathrm{X}} - f(\mathrm{X}) - g_1(\mathrm{X})\, u(t)\right) \tag{15}$$

Define

$$\tilde{d} = d - \hat{d} \tag{16}$$

Equations (13) and (15) are used to differentiate (16), explain

$$\dot{\tilde{d}} = -l(\mathrm{X})\, g_2\, \tilde{d} = 0 \tag{17}$$

where $l(x)$ is the disturbance observer gain matrix which can be designed such that the system (17), is exponentially stable for all $\mathrm{X} \in \mathrm{R}^n$. Consequently, disturbance estimate $\hat{d}(t)$ can approach $d(t)$ exponentially as $t \to \infty$. However, the disturbance observer (15) cannot be implemented due to the unavailability of $\dot{\mathrm{X}}$. To solve this, an auxiliary variable is defined

$$z = \hat{d} - p(\mathrm{X}) \tag{18}$$

Define the function $p(\mathrm{X})$ by

$$l(\mathrm{X}) = \frac{\partial p(\mathrm{X})}{\partial \mathrm{X}} \tag{19}$$

After substituting (15) and (19) into (18), we obtain the expression of the nonlinear disturbance observer is given as follows [15]

$$\begin{cases} \dot{z}(\mathrm{X}) = \dot{\hat{d}} - \frac{\partial p(\mathrm{X})}{\partial \mathrm{X}} \dot{\mathrm{X}} \\ = -l(\mathrm{X}) g_2(\mathrm{X}) z(\mathrm{X}) - l(\mathrm{X}) \left(\begin{array}{l} g_2(\mathrm{X}) p(\mathrm{X}) + f(\mathrm{X}) + \\ g_1(\mathrm{X}) u(t) \end{array} \right) \\ \hat{d}(\mathrm{t}) = z(\mathrm{X}) + p(\mathrm{X}) \end{cases} \quad (20)$$

From equations (12) and (20), we obtain the sliding mode control $u(t)$ as follows [15]

$$u(t) = g_1(\mathrm{X})^{-1} \left[\begin{array}{l} \dot{X}_c - \beta.e - f(\mathrm{X}) - g_2(\mathrm{X}).\hat{d}(t) - \\ KTanh(S) - S\frac{|g_2(\mathrm{X})|^2}{\sigma(\mathrm{X})} \end{array} \right] \quad (21)$$

with $\sigma(\mathrm{X}) = \frac{\partial p(\mathrm{X})}{\partial \mathrm{X}} g_2(\mathrm{X}) = l(\mathrm{X}) g_2(\mathrm{X})$ [15].

Fig. 2 shows the SMC-NDO controller. The disturbance Observer has been used to improve SMC control performance by estimating low-amplitude disturbances affecting the system.

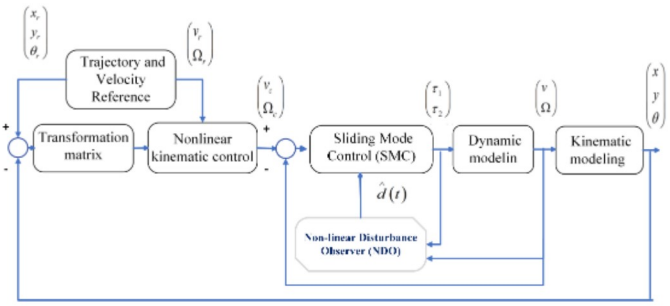

Fig. 2. Sliding mode control scheme based on a disturbance observer

After the running of the SMC-NDO [13], and SMC [50] controllers by MATLAB-SIMULINK, there are still large-amplitude white disturbances in the control despite the presence of the Disturbance Observer.

Motivated by this concept, we propose to combine the system Neural State Observer with the Sliding Mode Controller and the Disturbance Observer. The proposed controller is characterized by its stability and intelligence as well as its ability to handle significant white noise disturbances. In addition, it can also accurately estimate the angular velocity; known as a vector quantity that cannot be measured directly. These characteristics help avoid jitter and instability of control signals. They are particularly crucial for sensitive applications, such as industrial robots used for transporting heavy loads.

IV. Neural State Observation

The proposed Neural State Observer structure and observing error equations are discussed. Indeed, a feedforward neural network is used to replace the nonlinear function.

The equations system of the nonlinear model is express

$$\begin{cases} \dot{\mathrm{X}}(t) = F(\mathrm{X}(\mathrm{t}), \mathrm{u}(\mathrm{t})) = f(\mathrm{X}) + g_1(\mathrm{X}).u(t) + g_2(\mathrm{X}) d(t) \\ \Upsilon = C\mathrm{X} \end{cases} \quad (22)$$

A state observer for (22) can be described by

$$\begin{cases} \dot{\hat{\mathrm{X}}}(t) = F\left(\hat{\mathrm{X}}(t), u(t)\right) + \mathcal{L}C\left(\mathrm{X}(t) - \hat{\mathrm{X}}(t)\right) \\ \hat{\Upsilon}(t) = C\hat{\mathrm{X}}(t) \end{cases} \quad (23)$$

where

$\hat{\mathrm{X}}$ and $\hat{\Upsilon}$ represent the state and output observer, respectively. The observation gain matrix $\mathcal{L}$ is obtained after stability analysis of the neural state observer.

The estimation error is

$$e_o = \mathrm{X}(t) - \hat{\mathrm{X}}(t) \quad (24)$$

The derivative of the estimation error is obtained by

$$\dot{e_o} = F(\mathrm{X}(t), u(t)) - F(\mathrm{X}(t), u(t)) - \mathcal{L}Ce_o(t) \quad (25)$$

According to the approximation property of ANN, the nonlinear function $F(\mathrm{X}(t), u(t))$ can be represented by ANN with constant ideal weights $\mathcal{W}$ and $\mathcal{V}$ as follows. Thus, the ANN functional estimates for $F\left(\hat{\mathrm{X}}, u, t\right)$ is given by

$$\begin{cases} F(\mathrm{X}(t), u(t)) = \mathcal{W}\sigma(\mathcal{V}h) \\ F(\hat{\mathrm{X}}(t), u(t)) = \hat{\mathcal{W}}\sigma\left(\hat{\mathcal{V}}\hat{h}\right) \end{cases} \quad (26)$$

- $h(\mathrm{X}, u)$: is the input vector of neural network.

Neural state observer is described by

$$\begin{cases} \dot{\hat{\mathrm{X}}}(t) = \hat{\mathcal{W}}\sigma\left(\hat{\mathcal{V}}\hat{h}\right) + \mathcal{L}(\Upsilon(t) - \hat{\Upsilon}(t)) \\ \hat{\Upsilon}(t) = C\hat{\mathrm{X}}(t) \end{cases} \quad (27)$$

A. Artificial Neural Network (ANN) Architecture

A neural network consists of three different layers: input, hidden, and output. Each input is weighted and connected to a node in the hidden layer. Each hidden node is connected to each output layer node via weight ((see Table I).

The considered ANN studied in this work is:

- Three layers where the input layer contains four neurons, the first is the state vector $\mathrm{X} = \left[\begin{array}{cc} v & \Omega \end{array} \right]^{\mathrm{T}}$ and the second is the controller $u = \left[\begin{array}{cc} u_1 & u_2 \end{array} \right]^{T}$, 100 neurons in the hidden layer, and the output layer contains two neurons representing linear and angular velocity.
- The activation function of the hidden nodes is a unit sigmoid function to simplify and clarify the stability study development.

TABLE I. NEURAL NETWORK ARCHITECTURE

Input Layers	Hidden Layers	Output Layers
$(v(t), \Omega(t), u(t))$	100	$\left(\hat{v}(t), \hat{\Omega}(t)\right)$

Fig. 3 presents the Neural State Observer, which estimates the position error of the robot system by reconstructing the unmeasurable angular velocity in the SMC-NDO-NSO controller.

The observer incorporates two key elements: 1) a dedicated Artificial Neural Network method (see subsection IV-A) and 2) an adaptive observation matrix $\mathcal{L}$, automatically adjusted using the backpropagation algorithm (see subsection IV-B). This combination facilitates the development of robust and accurate situational awareness by optimizing convergence and effectively filtering noise.

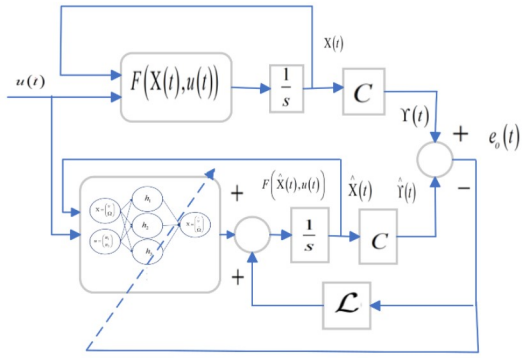

Fig. 3. Structure of a Neural State Observer (NSO)

B. Stability Analysis of a Neural State Observer Using the Backpropagation Algorithm

According to equations (25) and (26), the derivative of the estimating error is expressed as follows

$$\dot{e_o} = \mathcal{W}\sigma(\mathcal{V}h) - \hat{\mathcal{W}}\sigma\left(\hat{\mathcal{V}}\hat{h}\right) - \mathcal{L}Ce_o \tag{28}$$

The nonlinear function $F(\mathrm{X}(t), u(t))$ is assumed to be continuous, and Lipschitz concerning the argument $\mathrm{X}(t)$. Therefore, the Lipschitz property is written as

$$\forall \mathrm{X}(t), \hat{\mathrm{X}}(t), \exists\gamma / \begin{cases} \left\| F(\mathrm{X}(t), u(t)) - F\left(\hat{\mathrm{X}}(t), u(t)\right)\right\| \\ \leq \gamma \left\|\mathrm{X}(t) - \hat{\mathrm{X}}(t)\right\| \end{cases} \tag{29}$$

By selecting a candidate Lyapunov function of the form

$$V = e_o^T P e_o \tag{30}$$

the derivative of this Lyapunov function is then written

$$\begin{cases} \dot{V}(t) = \dot{e_o^T}(t) P e_o(t) + e_o^T(t) P \dot{e_o}(t) = \\ \\ -e_o^T(t)\left[C^T\mathcal{L}^T P + P\mathcal{L}C\right] e_o(t) + \\ 2e_o^T(t) P\left(F(\mathrm{X}(t), u(t)) - F\left(\hat{\mathrm{X}}(t), u(t)\right)\right) \end{cases} \tag{31}$$

Using the Lipschitz property, the second term on the right can be written as

$$2e_o^T(t) P\left\|\mathcal{W}\sigma(\mathcal{V}h) - \hat{\mathcal{W}}\sigma\left(\hat{\mathcal{V}}\hat{h}\right)\right\| \leq 2e_o^T(t) P\gamma\|e_o(t)\| \tag{32}$$

Replacing this result in the expression for the derivative of the Lyapunov function gives

$$\begin{cases} \dot{V}(t) = \dot{e_o^T}(t) P e_o(t) + e_o^T(t) P \dot{e_o}(t) \leq \\ -e_o^T(t)\left[C^T\mathcal{L}^T P + P\mathcal{L}C - 2P\gamma\right] e_o(t) \end{cases} \tag{33}$$

We can add (28) in Lyapunov function (33), yields

$$\begin{aligned} &\dot{V}(t) = \dot{e_o^T}(t) P e_o(t) + e_o^T(t) P \dot{e_o}(t) = \\ &\left(\mathcal{W}\sigma(\mathcal{V}h) - \hat{\mathcal{W}}\sigma\left(\hat{\mathcal{V}}\hat{h}\right) - \mathcal{L}Ce_o(t)\right)^T P e_o(t) + \\ &e_o^T(t) P\left(\mathcal{W}\sigma(\mathcal{V}h) - \hat{\mathcal{W}}\sigma\left(\hat{\mathcal{V}}\hat{h}\right) - \mathcal{L}Ce_o(t)\right) \\ &\leq -e_o^T(t)\left[C^T\mathcal{L}^T P + P\mathcal{L}C - 2P\gamma\right] e_o(t) \end{aligned} \tag{34}$$

Using Schur complement, we transform the algebraic inequality, relation (32), into a matrix inequality as follows:

$$\begin{bmatrix} C^T\mathcal{L}^T P + P\mathcal{L}C & P \\ P & \frac{1}{2\gamma}P \end{bmatrix} > 0 \tag{35}$$

The Lyapunov method is utilized to ensure the stability of the Neural State Observer based on the backpropagation algorithm, which is characterized by its ability to adjust the weights of connections between neurons and minimize the mean squared error between the actual outputs and the desired outputs to ensure the stability of the NSO and its accurate estimation of the system's state vector.

The gain matrix of $\mathcal{L}$ as well as P and γ variables are calculated to solve the optimization problem, using the YALMIP solver. This method also aims to determine the best-fit parameters in order to improve the performance of the controller. The values of the matrix $\mathcal{L}$ and P, as well as the γ value, are illustrated in Section VI.

V. SLIDING MODE CONTROL BASED ON NEURAL STATE OBSERVER AND DISTURBANCE OBSERVER

SMC-NDO-NSO combines the robustness of sliding-mode control, the capability of the NDO disturbance observer, and

NSO additional observation features. This integrated approach is designed to improve the overall performance of the control system, particularly in the presence of white noise disturbances and non-linearities.

Substituting Equations (27) into Equation (21), the robust controller for the nonlinear system (8) is proposed as

$$\begin{cases} u(t) = g_1^{-1}(\mathrm{X}) \begin{bmatrix} \dot{X}_c - \beta\left(\overset{\wedge}{\mathrm{X}}(t) - X_c(t)\right) - \\ f\left(\hat{\mathrm{X}}\right) + g_2(\mathrm{X})\,\hat{d}(t) - \\ KTanh(S) - S\frac{\left|g_2(\mathrm{X})^2\right|}{\sigma(\mathrm{X})} \end{bmatrix} \\ = g_1^{-1}(\mathrm{X}) \begin{bmatrix} \dot{X}_c - \beta\left(\int \begin{pmatrix} \hat{\mathcal{W}}\sigma\left(\hat{\mathcal{V}}\hat{h}\right) + \\ \mathcal{L}(\Upsilon - \overset{\wedge}{\Upsilon}) \end{pmatrix} - X_c\right) \\ -f\left(\hat{\mathrm{X}}\right) + g_2(\mathrm{X})\,\hat{d}(t) - KTanh(S) - \\ S\frac{\left|g_2(\mathrm{X})^2\right|}{\sigma(\mathrm{X})} \end{bmatrix} \end{cases} \tag{36}$$

with

- $e(t) = \overset{\wedge}{\mathrm{X}}(t) - X_c(t)$

To demonstrate the stability of the SMC-NDO-NSO loop, the positive definite Lyapunov function is given

$$V(S,d) = \frac{1}{2}S^2 + \tilde{d}^2 \tag{37}$$

The time derivative of $V(S,d)$ along the state trajectory is

$$\dot{V} = S\,\dot{S} + \tilde{d}\,.\,\dot{\tilde{d}} \tag{38}$$

Considering (8), (9) and (17), Equation (38) can be rewritten as

$$\dot{V} = \begin{pmatrix} S.g_2(\mathrm{X})\,.\,\tilde{d} - \frac{|S.g_2(\mathrm{X})|^2}{\sigma(\mathrm{X})} - \\ KTanh(S) - \frac{\partial p(\mathrm{X})}{\partial(\mathrm{X})} g_2(\mathrm{X})\,.\tilde{d}^2 \end{pmatrix} \tag{39}$$

Substituting (36) into (39), and since $\sigma(\mathrm{X}) = \frac{\partial p(\mathrm{X})}{\partial \mathrm{X}} g_2(\mathrm{X}) > 0$, yields

$$\dot{V} \leq \begin{pmatrix} 2\left|S.g_2(\mathrm{X})\right|\left|\tilde{d}(t)\right| - \sigma(\mathrm{X})\left|\tilde{d}(t)\right|^2 - \\ \frac{|S.g_2(\mathrm{X})|^2}{\sigma(\mathrm{X})} - KTanh(S) \end{pmatrix} \tag{40}$$

The derivative of $V(S,d)$ is as follows

$$\dot{V} = -\left(\sqrt{\sigma(\mathrm{X})}\left|\tilde{d}(t)\right| - \frac{\left|Sg_2(\mathrm{X})^2\right|}{\sqrt{\sigma(\mathrm{X})}}\right)^2 - KSTanh(S) \tag{41}$$

whatever the case $S \neq 0$ and $K > 0$

$$V \leq -KSTanh(S) \tag{42}$$

To satisfy the Lyapunov stability for SMC-NDO-NSO, the parameters k_a and k_b must be positive constants.

Fig. 4 illustrates the SMC-NDO-NSO control, which is responsible for keeping the robot on the reference path to minimize the error between the robot position, and velocities. This control strategy integrates adaptive gain and estimating techniques to reject power disturbances and accurately estimate the robot's velocities simultaneously.

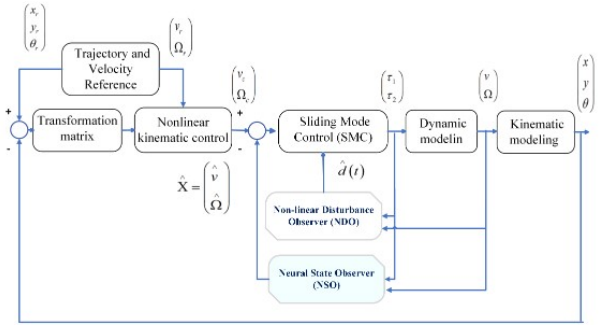

Fig. 4. Proposed control scheme based on SMC-NDO-NSO

VI. Simulation Results

The simulation was performed in Simulink MATLAB 2022b. Notes in Table II are the chosen settings for the UMR.

TABLE II. Unicycle Mobile Robot Parameters

Settings	Values
Platform mass	$m_c = 17\,\mathrm{kg}$
Wheel mass	$m_\Omega = 0.5\,\mathrm{kg}$
Wheel radius	$r = 0.095\,\mathrm{m}$
Half distance between two wheels	$L = 24\,\mathrm{m}$
The distance between wheel mass point A and C	$a = 0.05\,\mathrm{m}$
Inertia relative to the Centre of Gravity	$I_c = 0.537\,\mathrm{kg \cdot m^2}$
Inertia relative to wheel diameter	$I_m = 0.0011\,\mathrm{kg \cdot m^2}$
Inertia relative to the axis of the wheel	$I_\Omega = 0.0023\,\mathrm{kg \cdot m^2}$
Total mass of the robot	$m = 18\,\mathrm{kg}$
Robot inertia	$I = 0.6393\,\mathrm{kg \cdot m^2}$

To verify the effectiveness of the new control, a straight-track simulation was performed.

- Position of the robot:

$$\begin{cases} x_r(t) = t \\ y_r(t) = 3 \\ \theta_r(t) = 0 \end{cases} \quad a(t) = 0 \quad \begin{cases} x_0(t) = 0 \\ y_0(t) = 3 \\ \theta_0(t) = 0 \end{cases}$$

- Reference velocities:
$\begin{cases} v_r(t) = 3m/s \\ \Omega_r(t) = 0rad/s \end{cases} a(t) = 0 \begin{cases} v_0(t) = 1m/s \\ \Omega_0(t) = 0rad/s \end{cases}$
- Kinematic controller parameters:
$k_x = 6\ ;\ k_y = 1\ ;\ k_\theta = 0.2$
- SMC parameters:
$k_a = -20\ ;\ k_b = -20\ ;\ \beta = 1$
- NDO Parameters:
$l(X) = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}; p(\mathrm{X}) = \begin{bmatrix} 0.5 & x_1 \\ 0.5 & x_2 \end{bmatrix};$
$\sigma(\mathrm{X}) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$

The SMC-NDO-NSO control optimization problem aims to determine the optimal gain values to ensure UMR stability. To achieve this, we entered the initial condition of each value in the controllers and then, we solved the optimization problem using both YALMIP and Lyapunov methods (see Equation 42 and Equation 6).

Based on stability analysis of a neural state observer using the back-propagation algorithm for robot modeling, we concluded that the value following of P, $\mathcal{L}$, and γ (see Equation 34) are the most convenient:

$$\begin{cases} P = \begin{bmatrix} 597.1871 & 501.7899 \\ 501.7899 & 363.6224 \end{bmatrix} \\ \mathcal{L} = \begin{bmatrix} 130.9927 & 442.6544 \\ 442.6544 & 165.9741 \end{bmatrix} \\ \gamma = 0.9 \end{cases}$$

During training for velocity estimation, the neural network successfully learned the training data and achieved a performance value of 0.05 and a gradient of $1e-07$ in 6 validation checks within 100 iterations, as shown in Table III.

TABLE III. EVALUATION OF NEURAL NETWORK TRAINING RESULTS

Unit	Initial value	Stopped value	target value
Epoch	0	0	50
Performance	0.566	0.0315	0.05
Gradient	1.34	0.832	$1e-07$
Validation checks	0	0	6

After conducting several experiments, we have noticed that the hidden layer (number 100) yields an optimal layer size for the prediction method.

A. Simulation results: SMC-NDO-NSO controller applied on UMR in MATLAB-SIMULINK 2022b

Fig. 5 shows powerful white external noise disturbance values were used to assess the robustness of the SMC-NDO-NSO controller.

Fig. 6 displays motor torque, linear velocity, angular velocity, and estimation error (e_v, e_Ω) between actual velocities and estimated velocities using NSO. After about 3 seconds, the

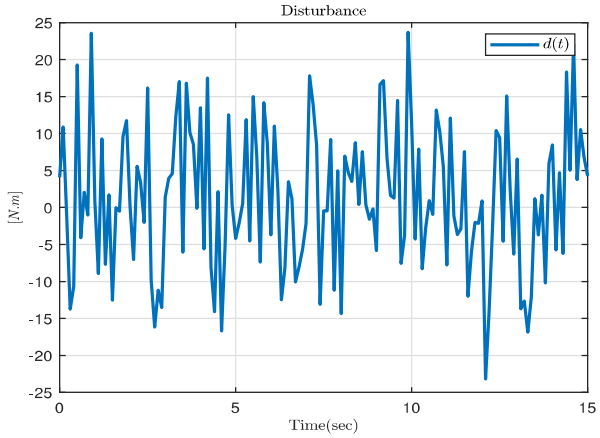

Fig. 5. White noise signal

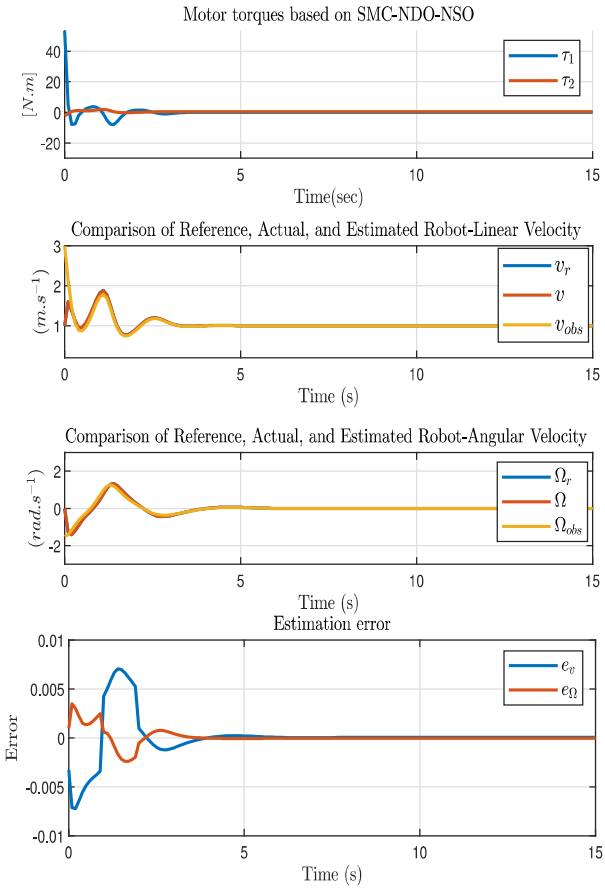

Fig. 6. Torque, Linear and angular velocities, and estimation error

torque (τ_1, τ_1) converges to 0, leading to a stable driving torque for the right and left wheels.

The following figure compares the UMR's reference (v_r, Ω_r), actual (v, Ω) and estimated velocities (v_{obs}, Ω_{obs}). It can be seen that the speeds estimated by NSO optimize the actual velocities to follow the robot's reference velocities. Furthermore, the forced external disturbance is reduced, contributing to the stabilization of both linear velocity (around $1m/s$) and angular velocity (around $0rad/s$) within 5 seconds.

Simulations, in the last figure, illustrate the effective response of the proposed Neural State Observer to estimation errors. It was demonstrated that the error converges to zero, generally less than 0.01 for both linear and angular velocities within 5 seconds, which greatly reduces the discrepancy between $\Upsilon(t)$ and $\hat{\Upsilon}(t)$. This means that the state estimator provides an accurate estimate of the robot's velocity.

The robust SMC-NDO-NSO control system enhances the system's resilience to external disturbances and ensures the robot's stability by enabling it to follow its reference trajectory, through its ability to resist external disturbances and estimate the system's status.

Fig. 7 illustrates the robot trajectory by demonstrating its convergence to the reference trajectory (x_r, y_r, θ_r) and straight-line stabilization and reducing the distance error (e_x, e_x, e_θ) to 0 within 5 seconds. Simulations validate the SMC-NDO-NSO approach and improve the robustness of the robot model. An additional Neural State Observer term enhances disturbance rejection and provides accurate URM tracking velocity estimation.

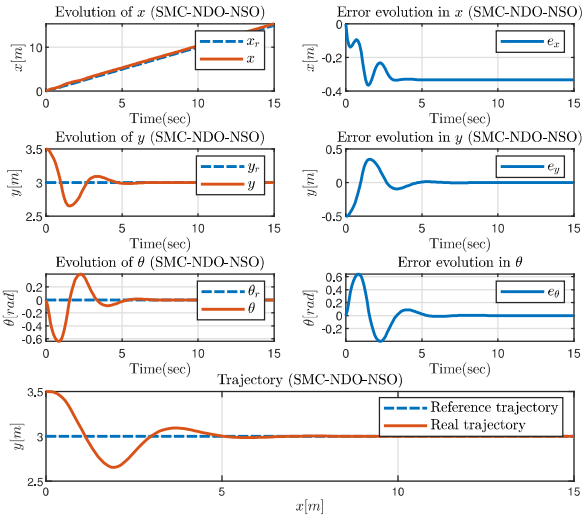

Fig. 7. Trajectories and Tracking errors

1) Simulation results: Difference between SMC-NDO-NSO, SMC-NDO and SMC control: The computational efficiency and performance characteristics of the SMC-NDO-NSO controller will be evaluated objectively using a comparative analysis of the SMC-NDO controller cited by [13] and the SMC controller stated by [50].

Our simulation results indicate that, despite the presence of powerful external disturbances and nonlinearities in the system, the proposed SMC-NDO-NSO scheme improves the disturbance rejection performance, as shown in both Fig. 8. Linear velocity is estimated to be 1, while angular velocity is estimated to be 0 within 3 minutes. On the contrary, we cannot guarantee the stability of linear and angular velocity based on the SMC-NDO and SMC controllers, due to the magnitude of disturbances reaching up to 2.8.

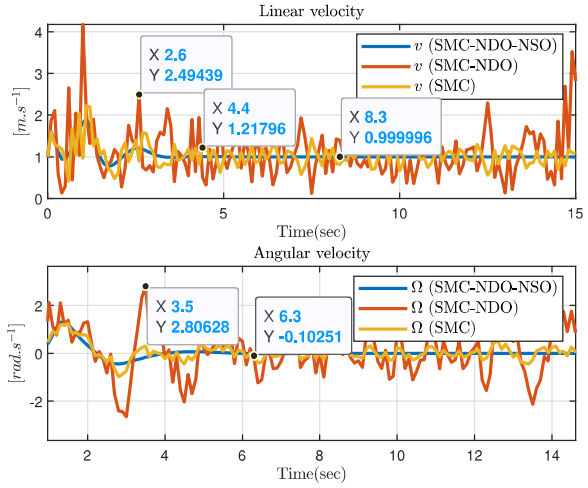

Fig. 8. Comparative Analysis of Velocity Performance: SMC-NDO-NSO, SMC-NDO, and SMC Controllers

In Fig. 9 we observe that the SMC-NDO-NSO controller converges to 0 in less than 5 minutes in a stable manner, allowing the robot to follow its reference trajectory smoothly. It also provides more satisfactory system responses than SMC-NDO [13] and SMC [50] structures that focus on low-amplitude external perturbations.

Fig. 10 presents the difference between the performance of SMC-NDO-NSO controller, which forced the disturbance to converge to 0, and the SMC-NDO controller [13], which was unable to react to the random disturbance whose amplitude was equal to 2.57. The strength of perturbations still gets stuck in the unstable system state, with disturbances large enough to result in a robot crash.

To enhance this comparison, we provide an in-depth analysis of the differences in Stability, Convergence time, and robustness between the three methods.

Stability and Convergence time In the SMC-NDO-NSO controller simulation, we can evaluate the timing stability of the system. In contrast, as for SMC-NDO and SMC controllers,

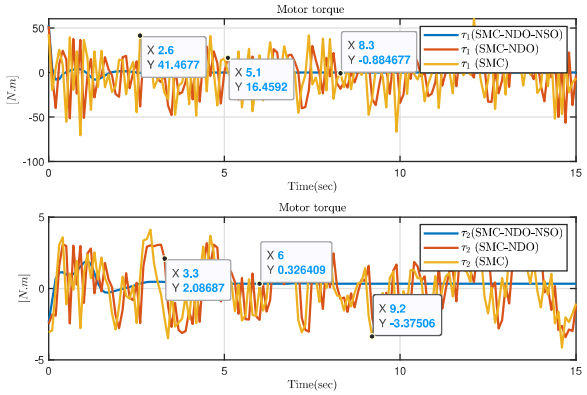

Fig. 9. Comparative Analysis of Motor Torques Performance: SMC-NDO-NSO, SMC-NDO, and SMC Controllers

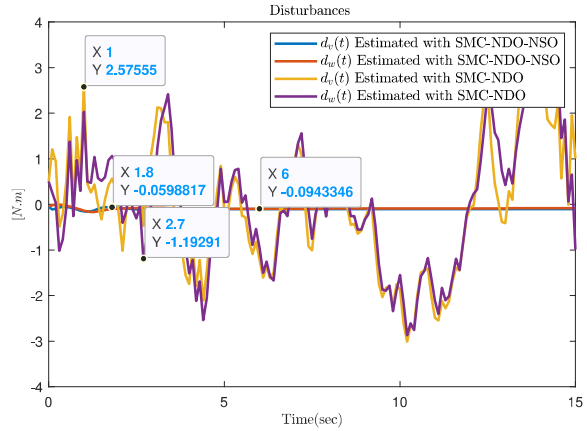

Fig. 10. Comparative Analysis of $d\,(t)$ Performance: SMC-NDO-NSO and SMC-NDO Controllers

assessing the timing stability of the system is difficult due to the severity of disturbances present in the overall system during the simulation.

Robustness against disturbances Based on numerical simulation comparison, we conclude that the SMC-NDO-NSO controller is robust and intelligent, unlike the SMC-NDO and SMC controllers.

B. Simulation results: Setup to test ROS Melodic in Gazebo simulator.

Our experiment was conducted on a robot running in ROS and robot Simulators shown in Fig. 11.

Table IV shows the characteristics of each device used to develop this work

On the host computer, Simulink or MATLAB sends operating data and sets commands to Gazebo on the target Linux machine. The software passes this information to the Gazebo Simulator, which displays sensor data and model information through the

Fig. 11. Implementation of SMC-NDO-NSO control in MiniLab ROS

TABLE IV. DESKTOP COMPUTER EQUIPPED

	Desktop computer N°1	Desktop computer N°2
Hardware	HP corei7 GTX	AMD Ryzen5 RTX
Software	MATLAB 2022b	VMware Workstation
Operating System	Windows 10	Ubuntu 20.04LTS

output terminal in real-time. The ROS device configuration aims to ensure communication with another ROS device show in Fig. 12.

The three major development steps, as shown in Fig. 13, are explained as follows:

Fig. 12. Connect to ROS Device

- In MATLAB-SIMULINK models: Before being implemented in the Simulink model, the SMC-NDO-NSO and SMC-NDO controllers applied to the UMR were extracted from linear and angular velocities estimated in the (.MAT) file format.
- In MATLAB-Simulink support for ROS: the input estimated state, based on SMC-NDO-NSO, showed that the two ports of the bus assignment block, (Linear.X) and (Angular.Z). This technique can collect linear and angular velocity signals.
- We used the Robotics System Toolbox to run our proposed method based on MATLAB/Simulink (Device 1) in the simulation environment (Device 2).

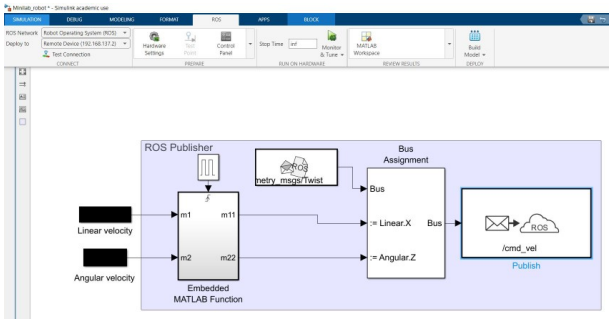

Fig. 13. Interaction of MATLAB® and Simulink® support by ROS to develop the SMC-NDO-NSO controller for autonomous navigation of UMR

The ROS publisher will establish a connection with Device 2 and identify the robot type and its Topic shown in Fig. 14.

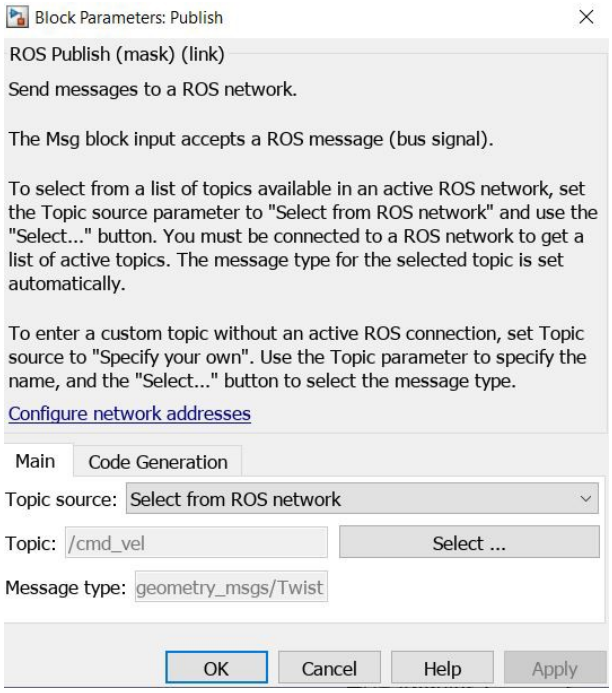

Fig. 14. ROS published: minilab robot

Fig. 15 represents a scenario in which MATLAB-SIMULINK and Gazebo/RVIZ communicate successfully, which is facilitated by the ROS framework to start moving and creating its map using SLAM algorithm.

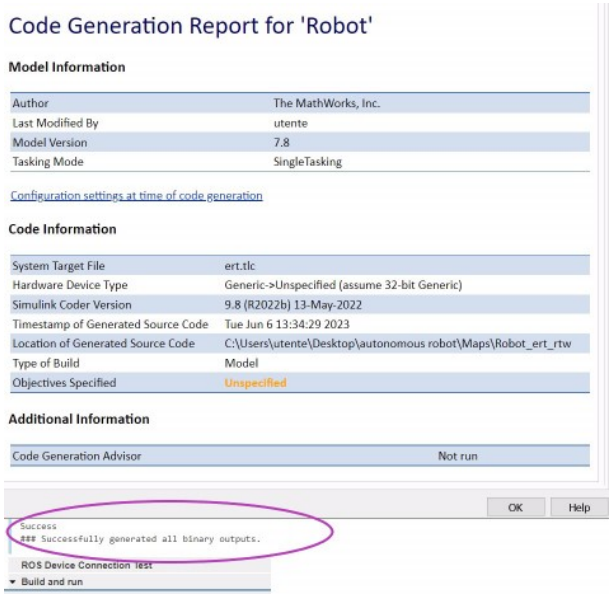

Fig. 15. Successful connection

We conclude that controlling and implementing the robot is straightforward.

Robot Navigation Tools:

1) ROS: Framework for robot control algorithms.
2) Gazebo: Realistic simulation and visualization.
3) GMapping: SLAM algorithm for mapping.
4) RVIZ: Observation and tracking tool for navigation.

To launch Gmapping we use the following command:

```
roslaunch minilab_demo_simulation
minilab_demo_gmapping.launch
```

1) Simulation results in Gazebo simulator Number 1:Test scenario with the MiniLab_robot based on SMC-NDO-NSO control: As illustrated in Fig. 16, the robot, which uses an estimated state vector based on SMC-NDO-NSO, maintains a straight path compatible with the reference trajectory throughout the test duration. Furthermore, the robot continues its normal course even after colliding with an obstacle, maintaining accurate values and output. Subsequently, it proceeds without deviation, navigating with unwavering precision from its initial position to its connection with the wall, guided by the SMC-NDO-NSO control system and empowered by SLAM-based 2D mapping in RVIZ, using a laser scanner.

The robot moves in a straight line, depending on the position of the scanner see Fig. 17, which is characterized by the organization of its precisely determined movement in its axis

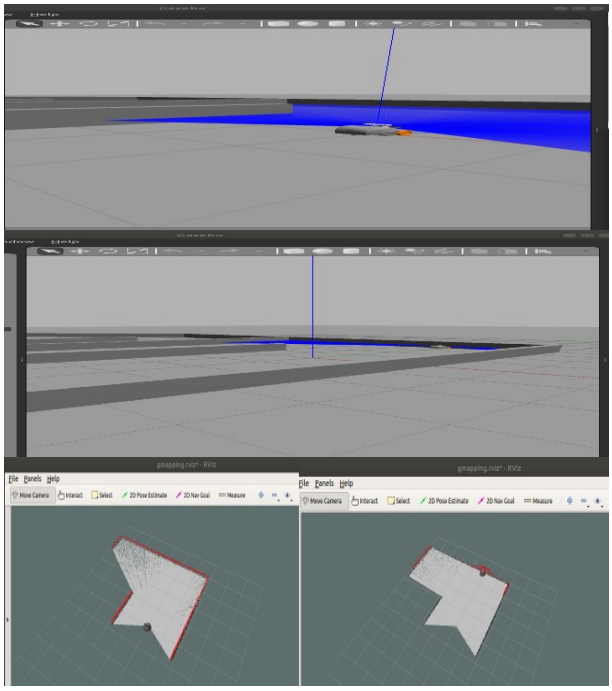

Fig. 16. MiniLab_robot in Gazebo and its map generation process based on SMC-NDO-NSO control

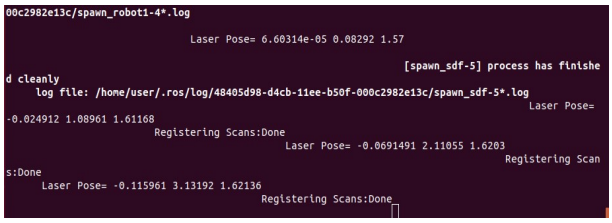

Fig. 17. Position of the laser scanner while turning the MiniLab_robot Based on SMC-NDO-NSO

(x, y, θ). Moreover, once the wall is removed, the robot remains stable.

2) Simulation results in Gazebo simulator Number 2: Test scenario with the MiniLab_robot based on SMC-NDO control: Fig. 18 shows that the robot, using a state vector based on the SMC-NDO control, deviates from the reference trajectory in a real environment as displayed in the Gazebo and RVIZ interface. The points scanned by the robot are wrong and unstable. Therefore, the robot navigates with deviations in its trajectory.

In Fig. 19 shows that the robot moves along a non-straight line because the location of the robot, according to the values (x, y, θ), is disorganized and unstable throughout the runtime and depends on the position of the scanner. Strong disturbances prevent the robot from accurately tracking its reference path.

3) Simulation results in Gazebo simulator Number 3: Test scenario with the Hysky_robot based on SMC-NDO-NSO control.: The ROS publish configuration makes it easy to change the robot type for testing SMC-NDO-NSO control by modifying the topic source and message type.

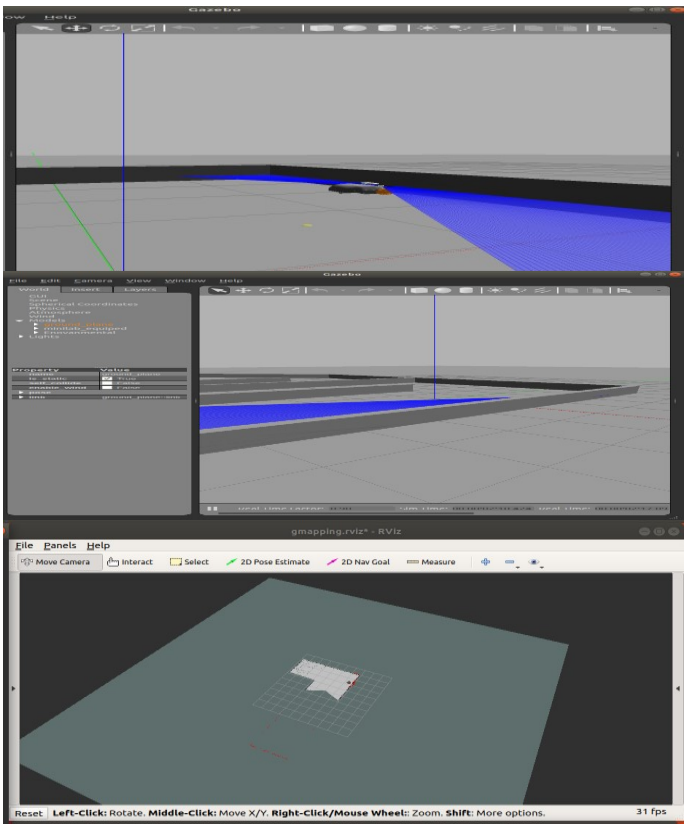

Fig. 18. MiniLab_robot in Gazebo and its map generation process based on SMC-NDO

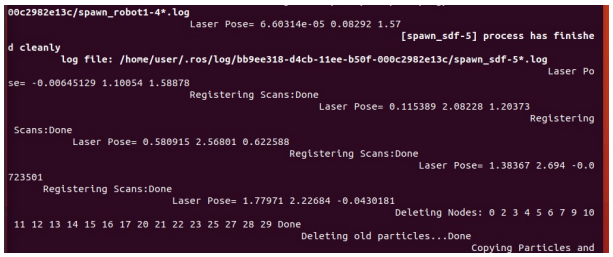

Fig. 19. Position of the laser scanner while turning the MiniLab_robot Based on SMC-NDO

`Topic:/husky_velocity_controller/cmd_vel Messagetype:geometry_msgs/Twist.`

As shown in both Fig. 20 and the laser pose in Fig. 21, the robot follows its linear path precisely.

4) Simulation results in Gazebo simulator Number 4: Test scenario with the Hysky_robot based on SMC-NDO control: At the time of implementation, the results of Laser Pose showed that the robot had lost its way for the first time. It started moving and then flipped over (see Fig. 22 and Fig. 23). In case of danger, we can stop the program.

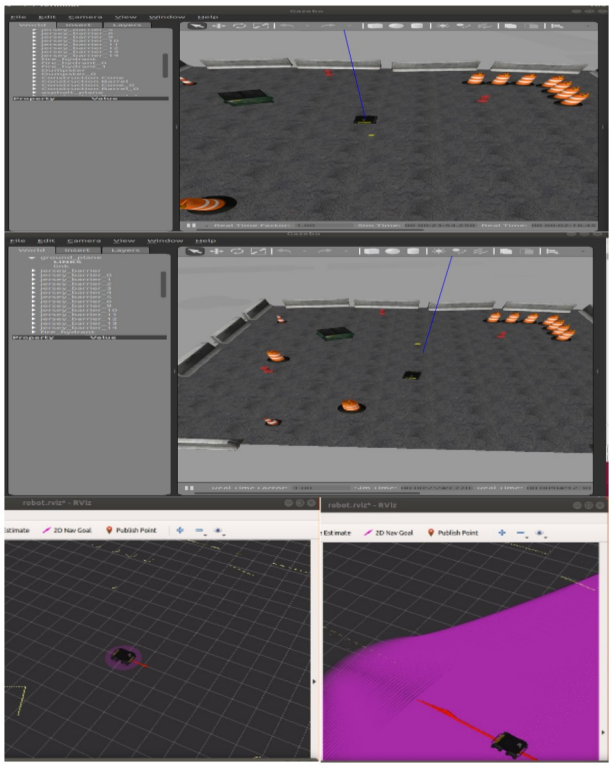

Fig. 20. Hysky_robot in Gazebo and its map generation process based on SMC-NDO-NSO

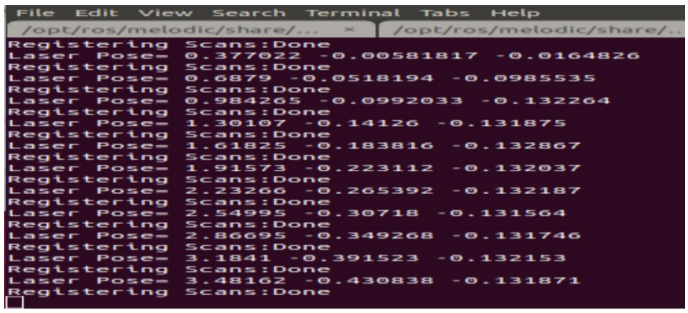

Fig. 21. Position of the laser scanner while turning the Hysky_robot Based on SMC-NDO-NSO

Fig. 24 exhibits the development of robust and intelligent control strategies for robots, with a particular focus on addressing challenges posed by random and non-linear disturbances. The primary purpose consists of ensuring robot stability, enhancing performance, rejecting the noise disturbances encountered, and improving the adaptability of an autonomous robot in dynamic environments.

1) The simulation of a kinematic model and a dynamic model (see section II
2) We simulated kinematic control and a sliding mode controller with a disturbance observer (see section III). Next, we proposed to integrate NSO in SMC-NDO (see Section IV, V)

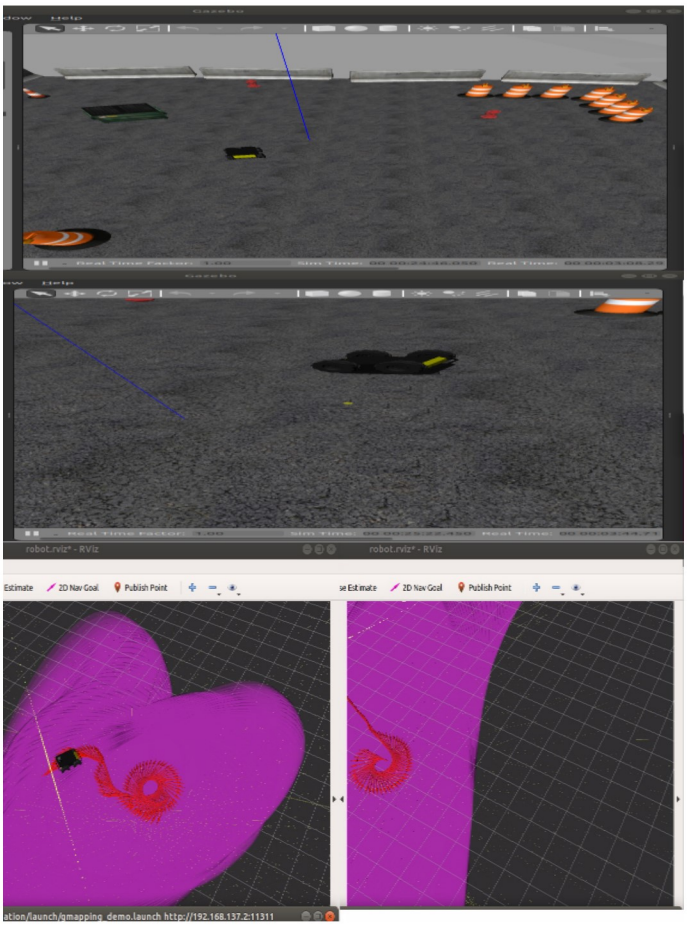

Fig. 22. Hysky_robot in Gazebo and its map generation process based on SMC-NDO

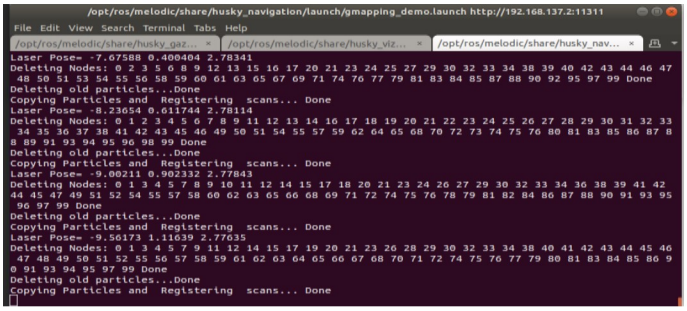

Fig. 23. Position of the laser scanner while turning the Hysky_robot Based on SMC-NDO

3) Our findings showed that by incorporating NSO into the SMC-NDO automatic control, velocities are accurately estimated (see Fig. 6, Fig. 7). Moreover, random and large perturbations imposed on the robot are suppressed (see Fig. 10), to guarantee system stability. Moreover, our SMC controller; named SMC-NDO-NSO has been successfully tested in Gazebo simulation (see Fig. 16, Fig. 18).

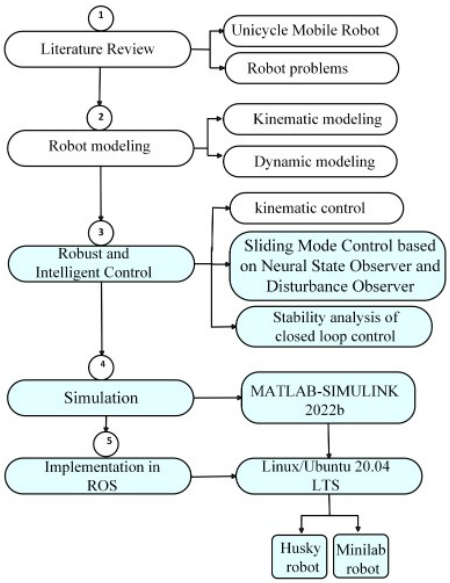

Fig. 24. Workflow chart

The proposed SMC-NDO-NSO control is characterized by two major terms in the world of autonomous robotics:

- **Compatibility:** The migration of the proposed work from the mini-lab robot to the Husky robot was seamless and error-free way, which underscored the success of the implementation process. The successful transfer confirmed flexibility and ensured a smooth transition, as it demonstrated compatible and adaptable communication between MATLAB/Simulink and the Gazebo simulator via ROS.
- **Effectiveness:** The robot's efficiency is attributed to its ability to move on a variety of surfaces. This diversity of mobility helps the robot to work successfully in diverse environments. A robust and intelligent robot controller was designed to enable the robot to move smoothly, either on smooth floors or uneven surfaces.

The robustness of the proposed controller consists of rejecting random disturbances and estimating non-measurable angular velocity in order to ensure robot trajectory stability. We have also highlighted the possibility of setting up simultaneous simulations between Simulink and Gazebo to send commands and receive data from Gazebo for real-world assessment. This technology can be applied to mobile robots carrying heavyweights, found notably in Industries 4.0, such as automated guided vehicles (AGVs), robotic forklifts, and sensitive environments, like military operations.

Testing the controller on non-terrestrial robots such as submarines and drones is essential due to the different environmental pressures they face. Submarines have to deal with water pressure at different depths, while drones navigate through changes in air pressure at different altitudes. Validating the controller's performance under these conditions is essential to ensure its reliability and efficiency in real-world applications.

VII. Conclusion

This paper introduces a novel robot-control method that combines Adaptive Sliding Mode Control, a Nonlinear Disturbance Observer, and a Neural State Observer. The NSO enhances the performance of the SMC-NDO controller.

This improvement ensures the system stability by minimizing error factors that could lead to malfunctions, especially in facing forced white noise disturbances and unmeasured state vectors that affect the robot trajectory. Thanks to its two-layer artificial neural network capability, the state observer was trained using an error-adjusted back propagation-learning algorithm and provided some correction terms to the Neural Network weights in order to improve the robustness of the NSO. Furthermore, it contains an adaptive matrix observation calculated by the YALMIP problem optimizer.

Two tests were used to evaluate our proposed control laws. Firstly, simulations using MATLAB/Simulink 2022b demonstrated the robustness of the proposed SMC-NDO-NSO controller architecture against strong internal and external disturbances. Furthermore, it outperformed both the SMC-NDO and SMC controllers in terms of response and performance. Secondly, by connecting MATLAB® to Gazebo via the ROS interface and experimenting with realistic scenarios, we came up with the conclusion that the SMC-NDO-NSO controller showed more responsiveness, compared to the SMC-NDO controller. SMC-NDO-NSO can be implemented in various types of robots on two kinds of surfaces (smooth and rough surfaces). However, when the force of white disturbances increases, the SMC controller exhibits disturbances after following the reference.

Our future works will put special focus on improving the neural network technique proposed in the state observer, such as modifying the activation function, replacing the neural network with Radial basis function (RBF) networks, or utilizing a Kalman Filter in the proposed controller to reduce more powerful external influences and disturbances.

We are currently working on improving this research as we aspire to inspire further exploration and innovation.

References

[1] A. T. Hassan, L. A. H. Al-Kindi and A. B. Abdulghafour, ""Industrie 4.0" and Smart Manufacturing: A State of the Art Review," *2023 15th International Conference on Developments in eSystems Engineering (DeSE),* pp. 1-6, 2023, doi: 10.1109/DeSE58274.2023.10100115.

[2] A. Bonci, F. Gaudeni, M. C. Giannini, and S. Longhi, "Robot Operating System 2 (ROS2)-Based Frameworks for Increasing Robot Autonomy: A Survey," *applied sciences,* vol. 13, no. 23, 2023, doi: 10.3390/app132312796.

[3] J. Rossmann, "eRobotics Meets the Internet of Things: Modern Tools for Today's Challenges in Robotics and Automation," *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, pp. 318-323, 2015, doi: 10.1109/DeSE.2015.33.

[4] A. Luxenburger *et al*., "Augmented Reality for Human-Robot Cooperation in Aircraft Assembly," *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 263-2633, 2019, doi: 10.1109/AIVR46125.2019.00061.

[5] A. T. Azar *et al*., "Design and Stability Analysis of Sliding Mode Controller for Non-Holonomic Differential Drive Mobile Robots," *machines*, vol. 11, no. 4, 2023, doi: 10.3390/machines11040470.

[6] A. Giallanza, G. L. Scalia, R. Micale, and C. M. L. Fata, "Occupational health and safety issues in human-robot collaboration: State of the art and open challenges," *Safety Science*, vol. 169, 2024, doi: 10.1016/j.ssci.2023.106313.

[7] A. P. Moreira, P. Neto, and F. Vidal, "Special Issue on Advances in Industrial Robotics and Intelligent Systems," *applied sciences,* vol. 13, no. 3, 2023, doi: 10.3390/app13031352.

[8] A. Latif, K. Shankar, and P. T. Nguyen, and Control, "Legged fire fighter robot movement using PID," *Journal of Robotic and Control,* vol. 1, no. 1, pp. 15-18, 2020, doi: 10.18196/jrc.1104.

[9] A. Giallanza, G. L. Scalia, R. Micale, and C. M. L. Fata, "Occupational health and safety issues in human-robot collaboration: State of the art and open challenges," *Safety Science*, vol. 169, 2024, doi: 10.1016/j.ssci.2023.106313.

[10] A. bin Kamarulariffin, A. bin Mohd Ibrahim, and A. Bahamid, "Improving Deep Reinforcement Learning Training Convergence using Fuzzy Logic for Autonomous Mobile Robot Navigation," *International Journal of Advance Computer Sciences and Applications*, vol. 14, no. 11, pp. 935-942, 2023, doi: 10.14569/ijacsa.2023.0141195.

[11] B. Nawress, A. N. G. Lakhal and N. B. Braiek, "Intelligent police robot uses VSLAM for Dynamic Environment based on Object Detection and Face Recognition," *2023 International Conference on Control, Automation and Diagnosis (ICCAD)*, pp. 1-6, 2023, doi: 10.1109/ICCAD57653.2023.10152436.

[12] B. Liu, X. Yang, and J. J. Zhang, "Nonlinear effect of industrial robot applications on carbon emissions: Evidence from China," *Environmental Impact Assessment Review,* vol. 104, 2024, doi: 10.1016/j.eiar.2023.107297.

[13] B. B. Mevo, "Contribution à la commande adaptive et robuste d'un robot mobile de type unicycle avec modèle non-linéaire," *Université du Québec en Abitibi-Témiscamingue*, 2019.

[14] B. Udugama, "Mini bot 3D: A ROS based Gazebo Simulation," *arXiv preprint arXiv:2302.06368,* 2023, doi: 10.48550/arXiv.2302.06368.

[15] M. Chen, and W. H. Chen, "Sliding mode control for a class of uncertain nonlinear system based on disturbance observer," *Adaptive Control and Signal Processing,* vol. 24, no. 1, pp. 51-64, 2010, doi: 10.1002/acs.1110.

[16] C. J. R. Chaka and P. i. T. E. Learning, "Fourth industrial revolution—a review of applications, prospects, and challenges for artificial intelligence, robotics and blockchain in higher education," *Research and Practice in Technology Enhanced Learning,* vol. 18, 2023, doi 10.58459/rptel.2023.18002 .

[17] C. C. Soon *et al*., "Chattering Analysis of an Optimized Sliding Mode Controller for an Electro-Hydraulic Actuator System," *Journal of Robotics and Control,* vol. 3, no. 2, pp. 160-165, 2022, doi: 10.18196/jrc.v3i2.13671.

[18] D. Marfuah, N. K. Ulya, D. P. D. Kusudaryati, A. S. Wardana, and E. Nugroho, "Current Trends in Intelligent Control Neural Networks for Thermal Processing (Foods): Systematic Literature Review," *Journal of Robotics and Control,* vol. 3, no. 4, pp. 519-527, 2022, doi: 10.18196/jrc.v3i4.15232.

[19] T. Ye, Z. Luo, and G. Wang, "Adaptive sliding mode control of robot based on fuzzy neural network," *Journal of Ambient Intelligence and Humanized Computing,* vol. 11, no. 12, pp. 6235-6247, 2020, doi: 10.1007/s12652-020-01809-2.

[20] D. T. Tran, N. M. Hoang, N. H. Loc, Q. T. Truong, and N. T. Nha, "A Fuzzy LQR PID Control for a Two-Legged Wheel Robot with Uncertainties and Variant Height," *Journal of Robotics and Control,* vol. 4, no. 5, pp. 612-620, 2023, doi: 10.18196/jrc.v4i5.19448.

[21] D. Huang, C. Yang, Z. Ju, and S. L. Dai, "Disturbance observer enhanced variable gain controller for robot teleoperation with motion capture using wearable armbands," *Autonomous Robots,* vol. 44, pp. 1217-1231, 2020, doi: 10.1007/s10514-020-09928-7.

[22] F. Hariz, Y. Bouslimani and M. Ghribi, "High-Resolution Mobile Mapping Platform Using 15-mm Accuracy LiDAR and SPAN/TerraStar C-PRO Technologies," in *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 4, no. 2, pp. 122-135, 2023, doi: 10.1109/JMASS.2023.3240892.

[23] X. Feng, and C. Wang, "Adaptive neural network tracking control of an omnidirectional mobile robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* vol. 237, no. 3, pp. 375-387, 2023, doi: 10.1177/09596518221135904.

[24] F. Zhang *et al*., "Universal nonlinear disturbance observer for robotic manipulators," *International Journal of Advance Robotic Systems,* vol. 20, no. 2, 2023, doi: 10.1177/17298806231167669.

[25] F. Sherwani, M. M. Asad and B. S. K. K. Ibrahim, "Collaborative Robots and Industrial Revolution 4.0 (IR 4.0)," *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pp. 1-5, 2020, doi: 10.1109/ICETST49965.2020.9080724.

[26] F. Ore, J. L. J. Sánchez, M. Wiktorsson, and L. Hanson, "Design method of human–industrial robot collaborative workstation with industrial application," *International Journal of Computer Integrated Manufacturing,* vol. 33, no. 9, pp. 911-924, 2020, doi: 10.1080/0951192X.2020.1815844.

[27] H. Wang, C. Liu, Q. Zhang, and L. Wang, "Efficient Nonlinear Model Predictive and Active Disturbance Rejection Control for Trajectory Tracking of Unmanned Vehicles," *Research Square,* 2024, doi: 10.21203/rs.3.rs-3858747/v1.

[28] H. Kagermann and W. Wahlster, "Ten years of Industrie 4.0," *sci,* vol. 4, no. 3, 2022, doi: 10.3390/sci4030026.

[29] Y. Hu, H. Yan, H. Zhang, M. Wang and L. Zeng, "Robust Adaptive Fixed-Time Sliding-Mode Control for Uncertain Robotic Systems With Input Saturation," in *IEEE Transactions on Cybernetics*, vol. 53, no. 4, pp. 2636-2646, 2023, doi: 10.1109/TCYB.2022.3164739.

[30] H. -W. Chae, J. -H. Choi and J. -B. Song, "Robust and Autonomous Stereo Visual-Inertial Navigation for Non-Holonomic Mobile Robots," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9613-9623, 2020, doi: 10.1109/TVT.2020.3004163.

[31] H. N. Huynh, H. Assadi, E. R. Lorphèvre, O. Verlinden, and K. Ahmadi, "Modelling the dynamics of industrial robots for milling operations," *Robotics and Computer-Integrated Manufacturing,* vol. 61, 2020, doi: 10.1016/j.rcim.2019.101852.

[32] A. A. Housein, G. Xingyu, W. Li, and Y. Huang, "Extended Kalman filter sensor fusion in practice for mobile robot localization," *The Science and Information Organization,* vol. 13, no. 2, 2022, doi: 10.14569/IJACSA.2022.0130204.

[33] H. M. Alshanbari, H. Iftikhar, F. Khan, M. Rind, Z. Ahmad, and A. Al. A. H. E. Bagoury, "On the Implementation of the Artificial Neural Network Approach for Forecasting Different Healthcare Events," *diagnostics,* vol. 13, no. 7, pp. 1-17, 2023, doi: 10.3390/diagnostics13071310.

[34] I. Reguii, I. Hassani, and C. Rekik, "Mobile Robot Navigation Using Planning Algorithm and Sliding Mode Control in a Cluttered Environment," *Journal of Robotics and Control,* vol. 3, no. 2, pp. 166-175, 2022, doi: 10.18196/jrc.v3i2.13765.

[35] J. Hazik, M. Dekan, P. Beno, and F. Duchon, "Fleet Management System for an Industry Environment," *Journal of Robotic and Control*, vol. 3, no. 6, pp. 779-789, 2022, doi: 10.18196/jrc.v3i6.16298.

[36] J. Cornejo *et al*., "Industrial, collaborative and mobile robotics in Latin America: Review of mechatronic technologies for advanced automation," *Emerging Science Journal,* vol. 7, no. 4, pp. 1430-1458, 2023, doi: 10.28991/ESJ-2023-07-04-025.

[37] J. Wang, H. Dong, F. Chen, M. T. Vu, A. D. Shakibjoo, and A. Mohammadzadeh, "Formation Control of Non-Holonomic Mobile Robots: Predictive Data-Driven Fuzzy Compensator," *mathematics,* vol. 11, no. 8, 2023, doi: 10.3390/math11081804.

[38] J. D. Téllez, R. S. G. Ramírez, J. P. Pérez, J. E. Carreón, M. A. C. Rosales, "ROS-based Controller for a Two-Wheeled Self-Balancing Robot," *Journal of Robotics and Control,* vol. 4, no. 4, pp. 491-499, 2023, doi: 10.18196/jrc.v4i4.18208.

[39] J. Gil, S. You, Y. Lee, and W. Kim, "Nonlinear sliding mode controller using disturbance observer for permanent magnet synchronous motors under disturbance," *Expert Systems with Applications,* vol. 214, 2023, doi: 10.1016/j.eswa.2022.119085.

[40] J. Arents and M. Greitans, "Smart industrial robot control trends, challenges and opportunities within manufacturing," *applied sciences,* vol. 12, no. 2, 2022, doi: 10.3390/app12020937.

[41] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," *Proceedings., IEEE International Conference on Robotics and Automation*, pp. 384-389 vol. 1, 1990, doi: 10.1109/ROBOT.1990.126006.

[42] K. Liu, H. Ji, and Y. Zhang, "Extended state observer based adaptive sliding mode tracking control of wheeled mobile robot with input saturation and uncertainties," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* vol. 233, no. 15, pp. 5460-5476, 2019, doi: 10.1177/0954406219849445.

[43] K. Niquet, N. Patel, F. Schrödel, M. Jahn and S. Varelmann, "Prototype of an intelligent textile based robot skin to expand potential applications for cobot's," *2023 8th International Conference on Mechanical Engineering and Robotics Research (ICMERR)*, pp. 12-17, 2023, doi: 10.1109/ICMERR59784.2023.10379882.

[44] K. P. Venkatesh, G. Brito, and M. N. K. Boulos, "Health digital twins in life science and health care innovation," *Annual Review of Pharmacology and Toxicology,* vol. 64, pp. 159-170, 2024, doi: 10.1146/annurev-pharmtox-022123-022046.

[45] S. Khesrani, A. Hassam, O. Boutalbi, and M. Boubezoula, "Motion planning and control of nonholonomic mobile robot using flatness and fuzzy logic concepts," *International Journal of Dynamics and Control,* vol. 9, no. 4, pp. 1660-1671, 2021, doi: 10.1007/s40435-020-00754-4.

[46] K. M. Saipullah, W. H. M. Saad, S. H. Chong, M. I. Idris, S. A. Radzi, "ROS 2 Configuration for Delta Robot Arm Kinematic Motion and Stereo Camera Visualization," *Journal of Robotics and Control,* vol. 3, no. 3, pp. 320-327, 2022, doi: 10.18196/jrc.v3i3.14436.

[47] M. Zhitenko and M. Polyakov, "Development of a Robot Arm with Neural Network Control," *2023 3rd International Conference on Technology Enhanced Learning in Higher Education (TELE)*, pp. 23-27, 2023, doi: 10.1109/TELE58910.2023.10184332.

[48] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics,* vol. 3, pp. 54–70, 2023, doi: 10.1016/J.Cogr.2023.04.001

[49] M. Schulze, F. Graaf, L. Steffen, A. Roennau and R. Dillmann, "A Trajectory Planner For Mobile Robots Steering Non-Holonomic Wheelchairs In Dynamic Environments," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3642-3648, 2023, doi: 10.1109/ICRA48891.2023.10161082.

[50] M. Mera, H. Ríos, and E. A. Martínez, "A sliding-mode based controller for trajectory tracking of perturbed unicycle mobile robots," *Control Engineering Practice,* vol. 102, 2020, doi: 10.1016/j.conengprac.2020.104548.

[51] N. M. Alyazidi, A. M. Hassanine, M. S. Mahmoud, and A. Ma'arif, "Enhanced Trajectory Tracking of 3D Overhead Crane Using Adaptive Sliding-Mode Control and Particle Swarm Optimization," *Journal of Robotics and Control,* vol. 5, no. 1, pp. 253-262, 2024, doi: 10.18196/jrc.v5i1.18746.

[52] M. Choi *et al*., "MSC-RAD4R: ROS-Based Automotive Dataset With 4D Radar," in *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7194-7201, 2023, doi: 10.1109/LRA.2023.3307005.

[53] R. Nabil, Z. Mokhtar, C. Soufyane, "Hybrid Approach to implement adaptive Neuro-Fuzzy Inference system for Trajectory Tracking navigation control of a wheeled mobile robot," *Przeglad Elektrotechniczny*, vol. 2023, no. 12, 2023, doi: 10.15199/48.2023.12.21.

[54] J. S. J. C. C. Amorim, J. H. O. Fernandes, M. S. Machado, A. L. C. Canella and M. F. Pinto, "Design of a Control Approach to Assist the Performance of a Competitive Line Follower Robot," *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, pp. 354-359, 2022, doi: 10.1109/LARS/SBR/WRE56824.2022.9996051.

[55] O. Y. Ismael, M. Almaged, and A. I. Abdulla, "Nonlinear Model Predictive Control-based Collision Avoidance for Mobile Robot," *Journal of Robotic and Control,* vol. 5, no. 1, pp. 142-151, 2024, doi: 10.18196/jrc.v5i1.20615.

[56] R. Fareh, S. Khadraoui, M. Y. Abdallah, M. Baziyad, and M. Bettayeb, "Active disturbance rejection control for robotic systems: A review," *Mechatronics,* vol. 80, 2021, doi: 10.1016/j.mechatronics.2021.102671.

[57] O. K. Nicesio, A. G. Leal and V. L. Gava, "Quantum Machine Learning for Network Intrusion Detection Systems, a Systematic Literature Review," *2023 IEEE 2nd International Conference on AI in Cybersecurity (ICAIC)*, pp. 1-6, 2023, doi: 10.1109/ICAIC57335.2023.10044125.

[58] R. S. Pol *et al*., "Autonomous Differential Drive Mobile Robot Navigation with SLAM, AMCL using ROS," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 5s, pp. 46-53, 2024.

[59] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control,* vol. 9, pp. 818-827, 2021, doi: 10.1007/s40435-020-00665-4.

[60] R. D. Xi *et al*., "Design and implementation of an adaptive neural network observer–based backstepping sliding mode controller for robot manipulators," *Transactions of the Institute of Measurement and Control,* vol. 46, no. 6, 2023, doi: 10.1177/01423312231190169.

[61] R. Sekhar, P. Shah, and I. Iswanto, "Robotics in industry 4.0: A bibliometric analysis (2011-2022)," *Journal of Robotic and Control,* vol. 3, no. 5, pp. 583-613, 2022, doi: 10.18196/jrc.v3i5.15453.

[62] R. Farkh, and K. Aljaloud, "Vision navigation based PID control for line tracking robot," *Intelligent Automation & Soft Computing,* vol. 35, no. 1, pp. 901-911, 2023, doi: 10.32604/iasc.2023.027614.

[63] S. H. T. Pham, C. D. Nguyen, K. D. Giap, and N. T. T. Vu, "Tracking controller for uncertain wheel mobile robot: adaptive sliding mode control approach," *Bulletin of Electrical Engineering and Informatics,* vol. 12, no. 6, pp. 3345-3353, 2023, doi: 10.11591/eei.v12i6.5586.

[64] S. Mutawe, M. Hayajneh and S. BaniHani, "Robust Path Following Controllers for Quadrotor and Ground Robot," *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1-6, 2021, doi: 10.1109/ICECCE52056.2021.9514140.

[65] S. Y. Bhuran and G. Phadke, "Movement Control of Amphibious Robot Pid Tuning Using Partial Swarm Optimization," *Available at SSRN 4699805*, 2024, doi: 10.2139/ssrn.4699805.

[66] S. Wang, "Comparative research on path planning algorithms for autonomous mobile robots based on ROS," in *International Conference on Algorithm, Imaging Processing, and Machine Vision (AIPMV 2023)*, vol. 12969, pp. 219-224, 2024, doi: 10.1117/12.3014664.

[67] S. Panagou, W. P. Neumann, and F. Fruggiero, "A scoping review of human robot interaction research towards Industry 5.0 human-centric workplaces," *International Journal of Production Research,* vol. 62, no. 3, pp. 974-990, 2024, doi: 10.1080/00207543.2023.2172473.

[68] S. Li, H. -T. Nguyen and C. C. Cheah, "A Theoretical Framework for End-to-End Learning of Deep Neural Networks With Applications to Robotics," in *IEEE Access*, vol. 11, pp. 21992-22006, 2023, doi: 10.1109/ACCESS.2023.3249280.

[69] S. Jestl, "Industrial robots, and information and communication technology: The employment effects in EU labour markets," *Regional Studies,* pp. 1-18, 2024, doi: 10.1080/00343404.2023.2292259.

[70] S. Mustary, M. A. Kashem, M. A. Chowdhury, and M. M. Rana, "Mathematical model and evaluation of dynamic stability of industrial robot manipulator: Universal robot," *Systems and Soft Computing,* vol. 6, 2024, doi: 10.1016/j.sasc.2023.200071.

[71] T. Ravichandran and F. Karray, "Knowledge based approach for online self-tuning of PID-control," *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol.4, pp. 2846-2851, 2001, doi: 10.1109/ACC.2001.946328.

[72] T. T. Mac, B. Q. Dat, and T. N. Sy, "A novel hedge algebra formation control for mobile robots," *Robotics and Autonomous Systems,* vol. 172, 2024, doi: 10.1016/j.robot.2023.104607.

[73] V. B. V. Nghia, T. Van Thien, N. N. Son, and M. T. Long, "Adaptive neural sliding mode control for two wheel self balancing robot," International Journal of Dynamics and Control, *International Journal of Dynamics and Control*, vol. 10, no. 3, pp. 771-784, 2022, doi: 10.1007/s40435-021-00832-1.

[74] W. Xue, B. Zhou, F. Chen, H. Taghavifar, A. Mohammadzadeh and E. Ghaderpour, "A Constrained Fuzzy Control for Robotic Systems," in *IEEE Access*, vol. 12, pp. 7298-7309, 2024, doi: 10.1109/ACCESS.2024.3352129.

[75] X. Wang, D. Ding, X. Ge, and Q. L. Han, "Neural-network-based control for discrete-time nonlinear systems with denial-of-service attack: The adaptive event-triggered case," *International Journal of Robust and Nonlinear Control,* vol. 32, no. 5, pp. 2760-2779, 2022, doi: 10.1002/rnc.5831.

[76] W. R. Abdul-Adheem, and I. K. Ibraheem, "Improved sliding mode nonlinear extended state observer based active disturbance rejection control for uncertain systems with unknown total disturbance," *International*

Journal of Advanced Computer Science and Applications(ijacsa), vol. 7, no. 12, 2016, doi: 10.14569/IJACSA.2016.071211.

[77] Y. Zhang, Y. Wu, K. Tong, H. Chen, and Y. Yuan, "Review of Visual Simultaneous Localization and Mapping Based on Deep Learning," *remote sensing,* vol. 15, no. 11, 2023, doi: 10.3390/rs15112740.

[78] Y. Yang, Y. Cai, Y. Jung Yoon, H. Zhao, and S. K. Gupta, "Sensor-Based Planning and Control for Conformal Deposition on a Deformable Surface Using an Articulated Industrial Robot," *Journal of Manufacturing Science and Engineering,* vol. 146, no. 1, 2024, doi: 10.1115/1.4063560.

[79] S. Yu, J. Lu, G. Zhu, and S. Yang, "Event-triggered finite-time tracking control of underactuated MSVs based on neural network disturbance observer," *Ocean Engineering,* vol. 253, 2022, doi: 10.1016/j.oceaneng.2022.111169.

[80] Y. Hu, B. Li, B. Jiang, J. Han, and C. Y. Wen, "Disturbance Observer-Based Model Predictive Control for an Unmanned Underwater Vehicle," *control and systems engineering,* vol. 12, no. 1, 2024, doi: 10.3390/jmse12010094.

[81] J. Zhai, H. Wang, J. J. N. C. Tao, and Applications, "Disturbance-observer-based adaptive dynamic surface control for nonlinear systems with input dead-zone and delay using neural networks," *Neural Computing and Applications,* vol. 35, no. 5, pp. 4027-4049, 2023, doi: 10.1007/s00521-022-07865-3.

[82] Z. Li, S. Li and X. Luo, "An overview of calibration technology of industrial robots," in *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 23-36, 2021, doi: 10.1109/JAS.2020.1003381.

[83] Z. Chen *et al*., "An Overview of In Vitro Biological Neural Networks for Robot Intelligence," *Cyborg and Bionic Systems*, vol. 4, 2023, doi: 10.34133/cbsystems.0001.

[84] Z. Liu, K. Peng, L. Han, and S. Guan, "Modeling and control of robotic manipulators based on artificial neural networks: a review," *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering,* vol. 47, pp. 1307–1347, 2023, doi: 10.1007/s40997-023-00596-3.