computers

MDPI

Article

A Fuzzy-Immune-Regulated Single-Neuron Proportional–Integral–Derivative Control System for Robust Trajectory Tracking in a Lawn-Mowing Robot

Omer Saleem [1], Ahmad Hamza [1] and Jamshed Iqbal [2,*]

[1] Department of Electrical Engineering, National University of Computer and Emerging Sciences, Lahore 54770, Pakistan
[2] School of Computer Science, Faculty of Science and Engineering, University of Hull, Hull HU6 7RX, UK
* Correspondence: j.iqbal@hull.ac.uk

Abstract: This paper presents the constitution of a computationally intelligent self-adaptive steering controller for a lawn-mowing robot to yield robust trajectory tracking and disturbance rejection behavior. The conventional fixed-gain proportional–integral–derivative (PID) control procedure lacks the flexibility to deal with the environmental indeterminacies, coupling issues, and intrinsic nonlinear dynamics associated with the aforementioned nonholonomic system. Hence, this article contributes to formulating a self-adaptive single-neuron PID control system that is driven by an extended Kalman filter (EKF) to ensure efficient learning and faster convergence speeds. The neural adaptive PID control formulation improves the controller's design flexibility, which allows it to effectively attenuate the tracking errors and improve the system's trajectory tracking accuracy. To supplement the controller's robustness to exogenous disturbances, the adaptive PID control signal is modulated with an auxiliary fuzzy-immune system. The fuzzy-immune system imitates the automatic self-learning and self-tuning characteristics of the biological immune system to suppress bounded disturbances and parametric variations. The propositions above are verified by performing the tailored hardware in the loop experiments on a differentially driven lawn-mowing robot. The results of these experiments confirm the enhanced trajectory tracking precision and disturbance compensation ability of the prescribed control method.

Keywords: lawn-mowing robot; steering control; single-neuron PID; Kalman filtering; fuzzy immune system; trajectory tracking; disturbance rejection

check for updates

Citation: Saleem, O.; Hamza, A.; Iqbal, J. A Fuzzy-Immune-Regulated Single-Neuron Proportional–Integral–Derivative Control System for Robust Trajectory Tracking in a Lawn-Mowing Robot. *Computers* **2024**, *13*, 301. https://doi.org/10.3390/computers13110301

Academic Editor: Robertas Damaševičius

Received: 19 October 2024
Revised: 14 November 2024
Accepted: 16 November 2024
Published: 19 November 2024

1. Introduction

Wheeled mobile robots (WMRs) are considered a core constituent of contemporary robotics, with applications spanning industrial automation, service robotics, and autonomous vehicles [1]. Their popularity arises from their relatively simple mechanical structure, energy efficiency, and ease of control, making them versatile for various tasks in both structured and unstructured environments [2,3]. The growing demand for automation in agricultural and domestic applications has significantly increased interest in WMRs capable of performing repetitive tasks with minimal human intervention [4]. One such application is lawn mowing, where autonomous robots offer a sustainable and labor-saving alternative to traditional methods [5–7]. The challenge in developing effective lawn-mowing robots, however, lies in achieving precise and robust trajectory tracking over uneven terrains and amidst various environmental disturbances [8]. To address these challenges, advanced control strategies must be employed to enhance the system's adaptability and reliability [9]. Agile feedback control strategies ensure robust trajectory tracking for lawn-mowing robots, where precision and resilience to environmental disturbances are necessary for optimum performance [10].

1.1. Literature Review

The development of WMRs has seen significant advancements, particularly in developing control systems that ensure accurate and efficient locomotion of the robotic system [11]. The kinematics of a differentially driven robot offer straightforward navigation in planar environments with minimal computational requirements [12]. However, as WMRs are increasingly deployed in complex and dynamic environments, more sophisticated control systems have been developed to enhance their performance [13]. Thus, agile feedback control strategies are crucial in ensuring robust trajectory tracking for lawn-mowing WMRs, where precision, adaptability, and resilience to environmental disturbances are necessary for optimal performance [14].

Control systems for WMRs can be categorized into classical, modern, intelligent, and inverted control approaches, each offering unique advantages and challenges [10,15]. The straightforward design and efficacy of the proportional–integral–derivative (PID) controllers make them popular in trajectory-tracking applications [16]. Nevertheless, traditional PID controllers often struggle with nonlinearities and time-varying uncertainties in dynamic environments [17]. The fractional order PID controller offers improved design flexibility and increased degrees of freedom, making the control procedure more responsive to nonlinearities and random perturbations [18]. Nevertheless, it also introduces several new variables that complicate the controller parameterization [19]. Additionally, the computational implementation of the non-integer order integral and differential operators used in fractional controllers is quite complex and thus typically necessitates specialized numerical techniques and approximation methods for real-time applications. Gain-scheduled adaptive PID controllers can vary the controller parameters online based on the current state error, improving performance over a broad range of operating conditions [20]. However, establishing sufficient closed-loop stability conditions poses a challenging problem. Sliding mode controllers (SMCs) offer high robustness against parameter variations and external disturbances, making them ideal for trajectory tracking on rough and unpredictable terrains [21,22]. However, it unavoidably introduces high-frequency oscillations and chattering, which cause actuator saturation or mechanical wear of the actuator(s) [23]. Model predictive control (MPC) provides a more flexible framework by optimizing control actions over a future time horizon while adhering to system constraints [24]. This allows for better handling of complex dynamics and environmental uncertainties [25]. Despite its robustness, MPC is computationally intensive, which can be a drawback for real-time applications [26].

The pervasive intelligent control strategies can learn from data and adapt to changing conditions, offering superior performance in unpredictable environments [27]. Intelligent control systems, which include approaches such as fuzzy logic and neural networks-based methods, have shown significant promise in enhancing the trajectory tracking accuracy of WMRs [28]. Fuzzy logic control provides an intelligent method for handling parametric uncertainties in the WMR's environment [29]. By incorporating a fuzzy-inference system (FIS), the controller can handle the inherent fuzziness in real-world conditions like uneven terrain, obstacles, or changes in grass height [30]. The fuzzy rules tend to improve the system's adaptability without requiring an exact mathematical model [31]. Artificial neural networks (ANN) can also be used to approximate nonlinear control laws for trajectory tracking [32]. A neural controller learns the robot's dynamics and environment through training, providing self-adapting capabilities that make it highly suitable for environments with unknown disturbances or complex dynamics [33,34]. However, it requires extensive training data and computational resources [35].

The single-neuron adaptive PID controller (SN-APID) is a specific neural network control procedure where only a single neuron is commissioned to modify the PID controller gains in real-time, enhancing its capability to handle environmental indeterminacies while preserving computational economy of the control algorithm [36,37]. An online training process is typically used to adaptively self-adjust the weights associated with the single-neuron model. A plethora of neural network training techniques have been published [38].

However, the majority of them suffer from poor learning rates and excessive sensitivity the operating conditions [39]. The extended Kalman filter (EKF) and its variants have recently gained a lot of attention because they use fewer neurons and epochs [40]. As compared to the conventional backpropagation algorithm, the EKF yields improved convergence time and faster learning speeds [41]. The EKF has demonstrated promising results for both offline as well as online training of various recurrent and feedforward neural network applications [42].

Another promising intelligent control approach is the integration of fuzzy logic and immune regulation mechanisms, which mimic the human immune system's ability to detect, adapt, and respond to disturbances [43]. This combination offers a robust and adaptive control framework capable of maintaining high performance in uncertain environments [44]. The fuzzy-immune-based controller is highly resilient and adaptable, making it ideal for environments where disturbances can affect the robot's performance [45].

1.2. Main Contributions

This article proposes a fuzzy-immune-regulated single-neuron PID control system specifically designed for trajectory tracking in a lawn-mowing robot. The single-neuron adaptive PID (SN-APID) control law enhances the controller's adaptability by self-adjusting its gains in real-time, while the serially connected fuzzy-immune block acts as an auxiliary self-regulation mechanism that improves the closed-loop system's resilience to disturbances. The SN-APID control law is driven by a preset cubic polynomial of the tracking error variable. The modified error signal creates amplified and attenuated regions of error, resulting in a stiff and soft control application, respectively. The fuzzy-immunological computations further refine the system's adaptability, offering a cascaded control strategy that dynamically reconfigures the control trajectory as a function of state variations and control input dynamics. By integrating these elements, the proposed control procedure achieves robust trajectory tracking, even under disturbance conditions, such as varying terrain gradients and obstacles. The key contributions of this paper are listed below:

1. Computation of the WMR's modified tracking error along the linear and angular positions using a pre-configured odd-powered cubic polynomial.
2. Constitution of error-cube-driven SN-APID controllers to track the WMR's linear and angular positions.
3. Formulation of the cascaded fuzzy-immune regulator to robustify the APID control signals.
4. Verification of the prescribed control scheme by performing credible experiments, in the physical environment, on a lawn-mowing WMR platform.

Feedback control strategies for robust trajectory tracking in lawn-mowing robots require adaptability and resilience to handle environmental disturbances and nonlinear system dynamics. Thus, the proposed hybrid control strategy integrates fuzzy-immune regulation with single-neuron PID control to provide a pragmatic approach for robust and precise trajectory tracking. The fuzzy-immune regulation addresses uncertainties and enhances adaptability to disturbances, whereas the single-neuron PID controller dynamically adjusts the control gains to respond to changing conditions, such as varied terrain or unexpected disturbances. This approach increases the robot's ability to maintain robust trajectory tracking in uncertain and nonlinear environments.

The rest of the article is structured as follows: the kinematic modeling and the velocity control procedure for a differentially driven WMR are discussed in Section 2. This section also describes the systematic constitution of the EKF-driven SN-APID control law. The proposed fuzzy-immune-regulated SN-APID control system is described in Section 3, along with its design process and associated mathematical formulae. Experimental results and their comparative analysis are presented in Section 4 to authenticate the proposed controller's efficacy in the physical environment. The paper is finally concluded with key insights and future research directions in Section 5.

2. System Description

Differentially driven WMRs are widely favored in various applications due to their simple design, controllability, and versatility in navigating different types of terrain [12]. They are typically equipped with two independently controlled wheels, each capable of rotating at different velocities to achieve both forward motion and steering [29]. Understanding the motion dynamics of a differential drive WMR typically requires a kinematic model that captures the robot's configuration and wheel dynamics.

2.1. WMR's Kinematic Model

The mathematical relationship between the WMR's overall motion and the individual wheel velocities is described by its kinematic model [46]. Specifically, this involves mapping the WMR's angular and linear velocities to the velocities of left and right wheels [47]. Assuming that the WMR operates on a plane and the wheels do not slip, its motion can be described via the following set of velocity components.

$$\dot{z}(t) = \begin{bmatrix} \dot{x}(t) & \dot{y}(t) & \dot{\theta}(t) \end{bmatrix}^T \tag{1}$$

where $\dot{x}(t)$ is the WMR's linear velocity along the x-axis dictating the forward (or backward) motion, $\dot{y}(t)$ is its linear velocity along the y-axis dictating the sideways motion, and $\dot{\theta}(t)$ is its angular velocity around the vertical axis dictating the rotation. The configuration of a differentially driven WMR in a global coordinate frame is summarized in terms of the following state variables.

$$z(t) = \begin{bmatrix} x(t) & y(t) & \theta(t) \end{bmatrix}^T \tag{2}$$

where $x(t)$ is the robot's position along the x-axis, $y(t)$ is the robot's position along the y-axis, and $\theta(t)$ is the robot's heading angle (or orientation). The forward kinematics of the WMR are described through the formulations in (3) and (4) of the WMR's linear velocity $v(t)$ and angular velocity $\dot{\theta}(t)$.

$$v(t) = \frac{r}{2}(v_R(t) + v_L(t)) \tag{3}$$

$$\dot{\theta}(t) = \frac{r}{L}(v_R(t) - v_L(t)) \tag{4}$$

where $v_R(t)$ is the velocity of right wheel, $v_L(t)$ is the velocity of left wheel, r is the wheel's radius, and L is the gap between the two wheel's centers (wheel base). The posture of the WMR about a given set of coordinates $(x_o, \quad y_o, \quad \theta_o)$ is shown in Figure 1.

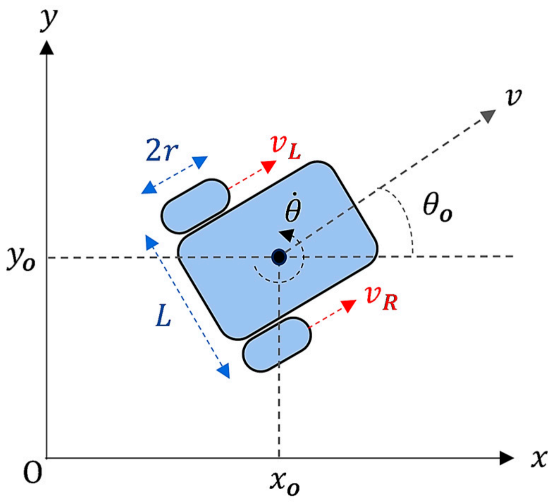

Figure 1. Position of the WMR.

The linear velocity components of the WMR along the x-axis and y-axis are represented, as shown in (5) and (6).

$$\dot{x}(t) = v(t)\cos\theta \tag{5}$$

$$\dot{y}(t) = v(t)\sin\theta \tag{6}$$

The WMR's forward kinematics are expressed in terms of $\dot{x}(t)$, $\dot{y}(t)$, and $\dot{\theta}(t)$, as shown below.

$$\dot{x}(t) = \left(\frac{r}{2}\cos\theta\right)v_R(t) + \left(\frac{r}{2}\cos\theta\right)v_L(t) \tag{7}$$

$$\dot{y}(t) = \left(\frac{r}{2}\sin\theta\right)v_R(t) + \left(\frac{r}{2}\sin\theta\right)v_L(t) \tag{8}$$

$$\dot{\theta}(t) = \frac{r}{L}v_R(t) - \frac{r}{L}v_L(t) \tag{9}$$

The equations above are rearranged, as shown in (10) and (11), to express $v_R(t)$ and $v_L(t)$ in terms of $\dot{x}(t)$, $\dot{y}(t)$, and $\dot{\theta}(t)$, [48].

$$v_R(t) = \frac{\cos\theta}{r}\dot{x}(t) + \frac{\sin\theta}{r}\dot{y}(t) + \frac{L}{r}\dot{\theta}(t) \tag{10}$$

$$v_L(t) = \frac{\cos\theta}{r}\dot{x}(t) + \frac{\sin\theta}{r}\dot{y}(t) - \frac{L}{r}\dot{\theta}(t) \tag{11}$$

The above relationships can be represented in matrix form, as shown in (12), facilitating the transformation from robot velocities into wheel velocities.

$$\begin{bmatrix} v_R(t) \\ v_L(t) \end{bmatrix} = \mathbf{T}\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} \tag{12}$$

where **T** is the transformation matrix. It is defined as shown in (13), [48].

$$\mathbf{T} = \begin{bmatrix} \frac{\cos\theta}{r} & \frac{\sin\theta}{r} & \frac{L}{r} \\ \frac{\cos\theta}{r} & \frac{\sin\theta}{r} & -\frac{L}{r} \end{bmatrix} \tag{13}$$

The WMR platform used for experimentation in this study has a wheel base of $L = 0.60$ m and wheel radius of $r = 0.15$ m (See Figure 1).

2.2. Measurement of Robot's Orientation and Position

The proposed robotic platform uses two different sensors to measure the WMR's orientation and position. The wheel encoders measure wheel rotation directly to estimate the robot's orientation and position. However, wheel slippage, wheel wear, and interaction with uneven terrains tend to distort its measurements by accumulating errors over time. The IMUs, on the other hand, provide the measurements of angular velocity and acceleration to estimate sudden changes in orientation and position. Since they have no contact with the wheel, their measurements remain unaffected by the terrain or wheel slippage. However, the measurements drift over time due to sensor noise and bias.

This study uses a complementary filter to compensate for the limitations of each sensor type (encoder drift and IMU noise) by fusing high-frequency data from the IMU with low-frequency data from wheel encoders [49]. It leverages the IMU's responsiveness and the encoder's long-term precision to provide a more accurate and reliable estimate of the system's position and orientation. Apart from reducing cumulative drift, it is computationally efficient and relatively simple to implement [49].

The detailed implementation of the complementary filter is discussed as follows. The IMU provides the robot's angular velocity ($\dot{\theta}_{imu}$) as well as its linear velocities in the x and y directions (as $\dot{x}_{imu}$ and $\dot{y}_{imu}$). The encoder provides the robot's angular position (θ_{en}) as well as its position in the x and y directions (as x_{en} and y_{en}). The complementary filters used to acquire the value of each state variable are formulated as shown below [49].

$$\theta(t) = (\alpha)\theta_{en} + (1-\alpha)\int \dot{\theta}_{imu}dt \tag{14}$$

$$x(t) = (\beta)x_{en} + (1-\beta)\int \dot{x}_{imu}dt \tag{15}$$

$$y(t) = (\beta)y_{en} + (1-\beta)\int \dot{y}_{imu}dt \tag{16}$$

where α and β are the filter's weighting coefficients that balance the contribution of the encoder and IMU data. Typically, α and β are restricted between [0 and 1]. Since the robot is moving on an uneven terrain, the values of α and β, therefore, are kept low to impose more weight on the IMU data. This avoids measurement distortion caused by wheel slippage. The weighting coefficients are manually tuned by trial and error. Their values are selected as $\alpha = 0.38$ and $\beta = 0.42$. The acquired state variables $\theta(t)$, $x(t)$, and $y(t)$ are fed to the velocity control architecture to compute the desired corrective actions. Smoothing out the sensor noise and drift indirectly minimizes the control chattering.

2.3. Velocity Control Architecture

This section presents a detailed discussion regarding the high-level velocity control design for the differentially driven WMR. To track the reference trajectory, the WMR requires accurate wheel velocity commands to help move it from its current position to the reference position [46]. For this purpose, the velocity control architecture, shown in Figure 2, is proposed in this research. The proposed control design acquires the instantaneous error in the positions $x(t)$ and $y(t)$ as well as the angular position $\theta(t)$ of the robot using the onboard wheel encoders and inertial measurement unit, respectively. The tracking errors $\varepsilon_x(t)$, $\varepsilon_y(t)$, and $\varepsilon_\theta(t)$ between the desired position and the current position are defined as shown in (17).

$$\varepsilon_x(t) = x(t) - x_{ref}(t),\ \varepsilon_y(t) = y(t) - y_{ref}(t),\ \varepsilon_\theta(t) = \theta(t) - \theta_{ref}(t) \tag{17}$$

where $x_{ref}(t)$ is the reference position along the x-axis, $y_{ref}(t)$ is the reference position along the y-axis, and $\theta_{ref}(t)$ is the reference orientation of the robot, as shown in Figure 3. These tracking error signals are modulated through a customized odd-powered cubic polynomial, enhancing the influence of large errors while reducing the impact of smaller errors [50].

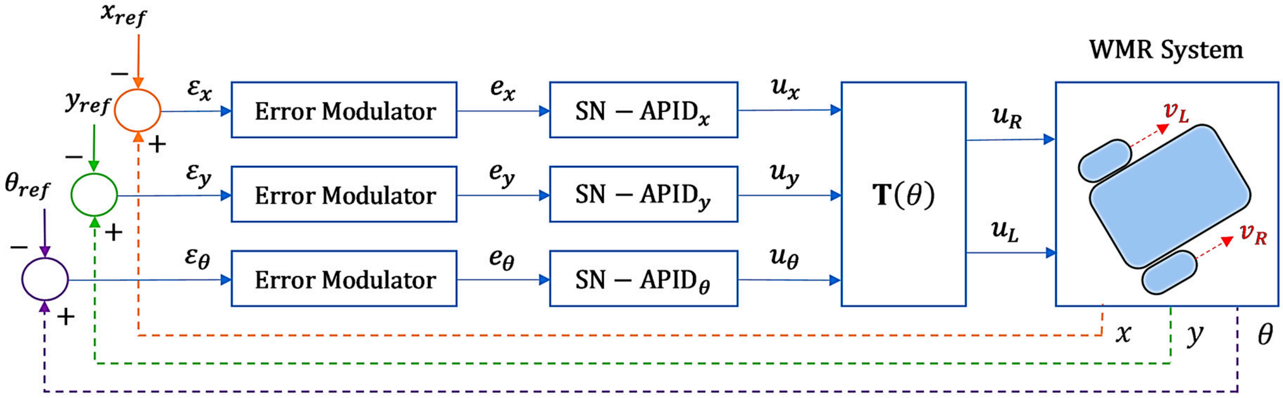

Figure 2. Baseline velocity control architecture of the WMR.

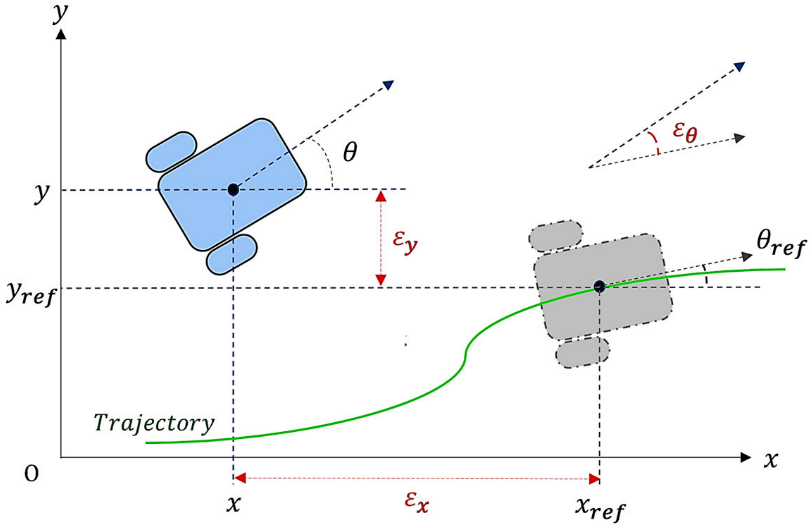

Figure 3. Trajectory-tracking errors of the WMR.

The nonlinear modulation is carried out as shown below.

$$e_z(t) = \beta_z(\varepsilon_z(t)) + \alpha_z(\varepsilon_z(t))^3 \tag{18}$$

$$\text{such that } z = x,\ y,\ or\ \theta$$

where $e_z(t)$ is the modified error signal associated with each state variable, β_z is the attenuation coefficient, and α_z is the transition coefficient. This arrangement distinctly improves the flexibility of the control procedure by creating an error amplification and an error attenuation region across the entire error regime of the system [51], as shown in Figure 4. The customized cubic polynomials help generate an aggressive control input under disturbance conditions due to the nonlinear magnification of the classical error signal $\varepsilon_z(t)$, as well as a gentle control input under equilibrium (steady-state) conditions due to the nonlinear attenuation of $\varepsilon_z(t)$. The aggressive control effort yields a faster response speed while strongly attenuating the disturbances, whereas the gentle control effort allows for a smoother oscillation-free response at or around the reference positions. The setting of the transition coefficient α_z decides the transition pattern of the error signals from the attenuation region to the amplification region. Its value is limited between 0 and 5 [51]. The coefficient β_z modifies the size of the attenuation region, which regulates the gentle control effort. The impact of error attenuation continues to get weaker if the value of β_z is set larger than unity because this setting relocates the transition point closer to the origin. This arrangement continually degrades the influence of the gentle control effort. Thus, the value of β_z is limited between 0 and 1 [51]. The nonlinear modulation of the error signals is realized via the following cubic polynomials.

$$e_x(t) = \beta_x(\varepsilon_x(t)) + \alpha_x(\varepsilon_x(t))^3 \tag{19}$$

$$e_y(t) = \beta_y\left(\varepsilon_y(t)\right) + \alpha_y\left(\varepsilon_y(t)\right)^3 \tag{20}$$

$$e_\theta(t) = \beta_\theta(\varepsilon_\theta(t)) + \alpha_\theta(\varepsilon_\theta(t))^3 \tag{21}$$

The values of the transition and attenuation coefficients are manually calibrated offline via the trial-and-error method by minimizing the integral-squared-error (ISE) cost function expressed in (22).

$$J = \int_0^t \left[(e_x(t))^2 + \left(e_y(t)\right)^2 + (e_\theta(t))^2\right] d\tau \tag{22}$$

The tuning procedure is concluded when the global minimum cost of J is acquired. The following set of parameter values are thus acquired: $\beta_x = 0.28$, $\alpha_x = 1.87$, $\beta_y = 0.24$, $\alpha_y = 1.91$, $\beta_\theta = 0.22$, and $\alpha_\theta = 1.82$.

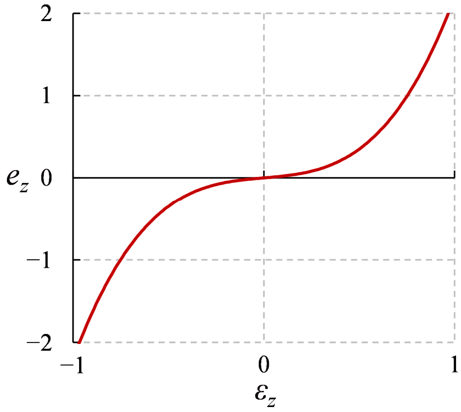

Figure 4. Behavior of modified error signal vs. linear error signal.

The modified error signals are then fed to their respective pre-calibrated SN-APID controllers to compute the robot's velocity control commands, which direct the robot's linear and angular positions. The ubiquitous PID controller is typically constituted by linearly combining the system's state error, error integral, and error derivative. The proportional control term attenuates the present state error, the error integral term dampens the overshoots and oscillations while improving the system's trajectory tracking behavior, and the derivative control term accelerates the transient reaction and forecasts future state changes for the system. The formulations of the PID controllers synthesized to regulate the linear position and orientation of the WMR are presented in (23)–(25).

$$u_x(t) = k_{P,x}e_x(t) + k_{I,x}\int_0^t e_x(t)d\tau + k_{D,x}\dot{e_x}(t) \tag{23}$$

$$u_y(t) = k_{P,y}e_y(t) + k_{I,y}\int_0^t e_y(t)d\tau + k_{D,y}\dot{e_y}(t) \tag{24}$$

$$u_\theta(t) = k_{P,\theta}e_\theta(t) + k_{I,\theta}\int_0^t e_\theta(t)d\tau + k_{D,\theta}\dot{e_\theta}(t) \tag{25}$$

where $k_{P,x}$, $k_{I,x}$, and $k_{D,x}$ represent the proportional gain, integrator gain, and differentiator gain associated with the x-axis position PID controller, respectively. Similarly, $k_{P,y}$, $k_{I,y}$, and $k_{D,y}$ represent the proportional gain, integrator gain, and differentiator gain associated with the y-axis position PID controller, respectively. Finally, $k_{P,\theta}$, $k_{I,\theta}$, and $k_{D,\theta}$ represent the proportional gain, integrator gain, and differentiator gain associated with the angular position PID controller, respectively.

Attaining a non-trivial optimum set of PID gains that yield the best control performance under every operating condition is indeed a challenging task. Hence, in this research, the PID gains of each controller are dynamically adjusted online by using the single neuron-based adaptation mechanism discussed in Section 2.4. The robot velocity control commands, $u_x(t)$, $u_y(t)$, and $u_\theta(t)$, yielded by the respective SN-APID controllers are subjected to the transformation matrix **T** to generate the corresponding wheel velocity control commands, $u_R(t)$ and $u_L(t)$.

The utilization of three individual SN-APID controllers is beneficial as compared to using a single multiple-input and multiple-output (MIMO) neural network structure with three inputs (e_x, e_y, and e_θ) and two outputs (u_R and u_L). The proposed control architecture with three single-neuron models offers three key benefits for real-time control systems: modularity, computational efficiency, and the capacity to effectively control each axis independently. The responsiveness and adaptability of the control systems in a real-time situation tend to become degraded using a MIMO network, as it necessitates extensive training which typically leads to slower convergence.

Compared to MIMO networks, the implementation of three single-neuron models is intrinsically simpler, requiring fewer parameters and less complicated training. These attributes increase the algorithm's computational efficiency, which is crucial for real-time applications. In contrast, multiple outputs and hidden layers in a MIMO network usually require more processing power and time to train, which inadvertently introduces control lag. Each single-neuron model in the proposed architecture focuses on independently regulating the dynamics of the state variable (x, y, and θ) assigned to it. Because of this independence, the weights of each neuron can be dynamically adjusted via the EKF to better suit the distinct properties of each control axis, resulting in precise control modifications and enhanced responsiveness. EKF parameter estimation becomes more complex in a MIMO system with multiple neurons, making it more difficult to effectively modify the weights. The additional computational load affects the system's adaptability and convergence. Finally, with three separate neurons, there is a minimal risk of unintended cross-coupling between control actions, as each neuron operates solely on one state variable.

2.4. Baseline SN-APID Controller

An artificial neuron is a mathematical representation of a biological neuron [52]. Despite its simplicity, an artificial neuron can perform a variety of functions. They are able to learn a specific task by acquiring knowledge from the environmental data. The information thus acquired is preserved in the synaptic weights, w.

2.4.1. Control Law Formulation

The control output of an artificial neuron is calculated as indicated in (26), [36,37].

$$u_z(t) = f_z\left(\boldsymbol{w}_z^T(t)\,\boldsymbol{m}_z(t)\right) \tag{26}$$

$$\text{such that } z = x,\ y,\ or\ \theta$$

where u_z represents the neuron's final control output for the linear (or angular) velocity regulation, $\boldsymbol{w}_z$ represents the weight vector, $\boldsymbol{m}_z$ represents the input vector, and $f_z(.)$ is the nonlinear activation function driven by the product of $\boldsymbol{w}_z^T$ with $\boldsymbol{m}_z$. Various approaches are available in the scientific literature to precisely train the single neuron. In this work, the EKF is used for the neuron's online training. The single-neuron PID controller is synthesized by fusing the artificial neuron's self-learning and self-tuning capacity with a PID controller. This combination allows the PID gain settings to be dynamically updated in response to the tracking error variations. Consequently, the suggested control method achieves better flexibility and resilience to disturbances.

The vector $\boldsymbol{m}_z$ contains the error, error-integral, and error-derivative variables linked with a particular state of the robot. The input variables $m_{1,z}(t)$, $m_{2,z}(t)$, and $m_{3,z}(t)$ are expressed in (27).

$$m_{1,z}(t) = e_z(t),\ m_{2,z}(t) = \int_0^t e_z(t)d\tau,\ m_{3,z}(t) = \dot{e_z}(t) \tag{27}$$

The coefficients of the weight vector w are self-adjusted online via the EKF algorithm discussed in the next subsection. It is to be noted that PID gains $k_{P,z}$, $k_{I,z}$, and $k_{D,z}$ are indirectly represented by the online self-tuning weights $w_{1,z}(t)$, $w_{2,z}(t)$, and $w_{3,z}(t)$, respectively. This arrangement, in turn, transforms the controller into an online adaptive PID controller. The function $h_z(t)$ represents the neuron's raw output. It is formulated as a weighted sum of input variables, as shown in (28), [36].

$$h_z(t) = \sum_{j=1}^{3} w_{j,z}(t)\, m_{j,z}(t) \tag{28}$$

To introduce nonlinearity into the model, the neuron's raw output is fed to the hyperbolic tangent function, $\tanh(.)$. This activation function enables the neural model to address the intrinsic unmodeled nonlinearities. It exhibits the odd symmetry of a traditional signum function while avoiding the hard limits imposed by it. Its waveform is shown in Figure 5. The $\tanh(.)$ function normalizes the neuron's output between $[-1$ and $1]$ and modulates it with a pre-determined scaling constant $\delta_{z,o}$. This nonlinear modulation yields the neuron's final output as expressed in (29), [36].

$$u_z(t) = \delta_{z,o} \tanh(h_z(t)) \tag{29}$$

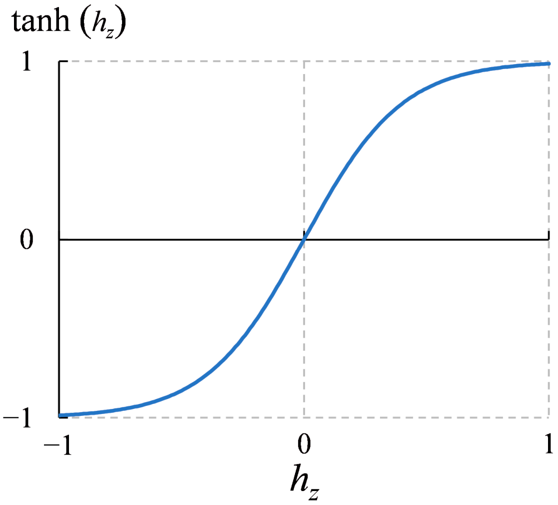

Figure 5. Waveform of the hyperbolic tangent function.

The values of the scaling constants are restricted between 0 and 1 to prevent actuator saturation post-transformation. These constants are calibrated offline by minimizing the ISE-based objective function J, expressed in (22). The following set of parameter values are thus acquired: $\delta_{x,o} = 0.32$, $\delta_{x,o} = 0.29$, and $\delta_{\theta,o} = 0.21$. The schematic diagram of the single-neuron adaptive PID (SN-APID) control scheme is depicted in Figure 6.

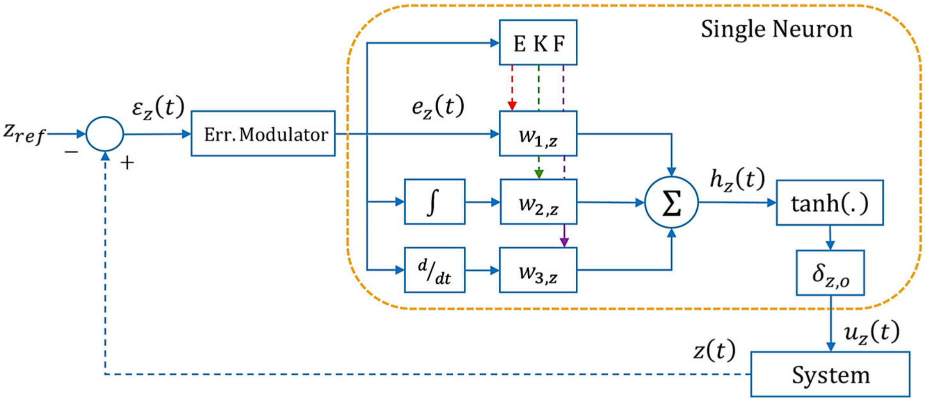

Figure 6. Schematic diagram of the SN-APID control scheme.

2.4.2. EKF-Based Online Training

The primary goal of the training procedure is to online adapt the coefficients of weight vector $\boldsymbol{w}_z$ that efficiently reduce the prediction error while handling the nonlinearities associated with the control problem [36].

Hence, in this neural control problem, the EKF training process is utilized owing to its fast convergence speed, improved learning, and ability to handle nonlinear mapping of the neural network. The EKF is a probabilistic estimator that predicts the measurement that could result from estimating the system states [42]. The estimation procedure is carried out recursively until the algorithm has converged to the optimum estimate. The weights of the single neuron are treated as the state variables to be estimated after every sampling interval. The EKF-driven online-training process is performed individually for each PID controller. Consider the nonlinear system in (30), [53].

$$s_{n+1} = f(s_{\,n}) + \varphi_n,\ d_n = g(s_n) + \omega_n \tag{30}$$

where s_n is the system's state vector at the n^{th} instance, φ_n is the process noise, d_n is the observation vector, $g(.)$ is a nonlinear state function, and ω_n is the observation noise [40]. The composition of the system's state vector for the online adaptation of the weighting coefficients is expressed as shown in (31).

$$s_n = \begin{bmatrix} w_{1,z} & w_{2,z} & w_{3,z} \end{bmatrix} \tag{31}$$

The EKF estimation is carried out by using the following system of equations [53].

$$F_n = I \tag{32}$$

$$G_n = \frac{\partial g(s_n)}{\partial q_n} = \frac{\partial z}{\partial s_n} = \begin{bmatrix} \frac{\partial z}{\partial w_{1,z}} & \frac{\partial z}{\partial w_{2,z}} & \frac{\partial z}{\partial w_{3,z}} \end{bmatrix} \tag{33}$$

$$K_n = P_n {G_n}^T \left(R_n + G_n P_n {G_n}^T \right)^{-1} \tag{34}$$

$$\hat{s}_{n+1} = \hat{s}_n + \eta K_n e_n \tag{35}$$

$$P_{n+1} = P_{\,n} - K_n G_n P_n + Q_n \tag{36}$$

where $\hat{s}_n$ is the updated estimate of the state (weight) vector, G_n is the estimated state's measurement model, K_n is the Kalman gain-vector, P_n refers to the covariance of prediction error, R_n is the covariance matrix of ω_n, Q_n is the covariance matrix of φ_n, and η is a diagonal matrix of the learning rates used to adjust each weight independently [40]. It is formulated as $\eta = diag\begin{bmatrix} \eta_1 & \eta_2 & \eta_3 \end{bmatrix}$. The same set of learning gains is used for each of the three SN-APID controllers. The partial derivative of the system with respect to the synaptic weight(s) is evaluated as shown in (37), [53].

$$\frac{\partial z}{\partial w_{j,z}} = \frac{\partial z}{\partial u_z} \times \frac{\partial u_z}{\partial h_z} \times \frac{\partial h_z}{\partial w_{j,z}} \tag{37}$$

$$\text{such that } \frac{\partial z}{\partial u_z} = \frac{z(t) - z(t-1)}{u_z(t) - u_z(t-1)},\ \frac{\partial u_z}{\partial h_z} = \delta_z \operatorname{sech}^2(h_z(t)),\ \frac{\partial h_z}{\partial w_{j,z}} = m_{j,z}(t).$$

where $j = 1,\ 2,$ or $3.$ The coefficients of the diagonal matrices P and Q are manually set as $P_{11} = P_{22} = P_{33} = 1$ and $Q_{11} = Q_{22} = Q_{33} = 0.1$. The other parameters of the EKF are tuned offline by minimizing the cost function J, mentioned in (19). The coefficient of R is set as 0.0014. The settings of the EKF's learning rates are $\eta_1 = 0.281$, $\eta_2 = 0.322$, and $\eta_3 = 0.021$. The initial weights associated with each neuron are chosen randomly.

The EKF filter inherently reduces the impact of noise and chattering in the estimated values of weights, which ensures that the consequent control actions are smooth and responsive to the system's current state. The SN-APID control law augmented with the EKF algorithm, for each state variable, is thus formulated in (38)–(40).

$$u_x(t) = \delta_{x,o}\tanh\big(h_x(t)\big) \tag{38}$$

$$u_y(t) = \delta_{y,o}\tanh\big(h_y(t)\big) \tag{39}$$

$$u_\theta(t) = \delta_{\theta,o}\tanh\big(h_\theta(t)\big) \tag{40}$$

The PID gains are updated online after every sampling instant. The corresponding velocity control signals, $u_x(t)$, $u_y(t)$, and $u_\theta(t)$ are then transformed to deliver the updated wheel velocity-control signals, $u_R(t)$ and $u_L(t)$, of the WMR by using the following transformation, as shown in Figure 2.

$$\begin{bmatrix} u_R(t) \\ u_L(t) \end{bmatrix} = \mathbf{T}\begin{bmatrix} u_x(t) \\ u_y(t) \\ u_\theta(t) \end{bmatrix} \tag{41}$$

3. Proposed Fuzzy-Immune-Regulated SN-APID Control Procedure

The immune-inspired control systems are modeled after the human immune system, which adapts to external disturbances by detecting and neutralizing threats [45]. In trajectory tracking, an immune-regulated control system detects deviations from the desired path and quickly adapts the control response to maintain the trajectory under adverse conditions.

The biological immune system, which is primarily made up of lymphocytes and antibody molecules, is resistant to pathogen infiltration [54]. The *B*-cells, suppressor *T*-cells (T_S cells), and helper *T* cells (T_H cells) aid in the production of lymphocytes. The B-cell surface receptors gauge the extent of the foreign antigen invasion. As per their diagnosis, the B cells stimulate a proper concentration of T_H cells to aid in the production of plasma cells. To repel the antigen infiltration, the plasma cells produce antibodies. The production of T_S cells is triggered to inhibit the growth of antibodies as the antigen attack weakens. The T_S and T_H cells synergistically work together to balance the process of antibody formation by activating and inhibiting it in turn. This action efficiently recovers the biological system and increases its resistance to the invading antigens. The following set of formulations is used to describe the B-cell concentration produced during this procedure [54].

$$c(n) = T_H(n) - T_S(n) \tag{42}$$

$$\text{such that } T_H(n) = \rho\, d(n),\ T_S(n) = \rho\, \lambda\sigma\big(c(n), \dot{c}(n)\big)q(n)$$

where n is the generation of antigen and antibody proliferation, $c(n)$ is the stimulus (or B-cell concentration), $T_H(n)$ is the T_H cell concentration, $T_S(n)$ is the T_S cell concentration, $q(n)$ is the antigen concentration, $\sigma(.)$ is a preset nonlinear scaling function that adaptively modulates the antibody inhibition rate, and ρ and λ are positive weighting coefficients that ascertain the immune system's response speed and damping strength. The total stimulation is expressed in (43), [54].

$$c(n) = \rho\big(1 - \lambda\sigma\big(c(n), \dot{c}(n)\big)\big)q(n) \tag{43}$$

In practice, bounded external disturbances have the potential to deteriorate the performance of robotic systems. In order to effectively reject the disturbances while maintaining the tracking accuracy, a fuzzy-immune regulator is therefore developed in this study. A logic map that compares the components of a vertebrate immune response system with those of the artificial immune-regulated WMR system is shown below.

Biological system		Physical system
Immune system	→	WMR
The n^{th} generation of antibody reproduction	→	The sampling interval of the WMR
The antigen concentration, $q(n)$	→	The normalized control variable, $\tanh(h_z(t))$
The B-cell stimulation, $c(n)$	→	The control signal, $u_z(t)$

The following artificial immune control equation is derived from the map above.

$$u_z(t) = \delta_z(t) \times \tanh(h_z(t)) \tag{44}$$

The fuzzy-immune regulator modifies the control input $u_z(t)$ as per the variations in $\tanh(h_z(t))$ as well as the self-adjusting gain $\delta_z(t)$. The gain $\delta_z(t)$ is dynamically self-adjusted as a function of control input variables, as shown in (45).

$$\delta_z(t) = \delta_{z,o}\left(1 - \lambda_z \sigma\left(u_z, \dot{u}_z\right)\right) \tag{45}$$

where $\delta_{z,o}$ is the base value of the gain. The fuzzy-immune-regulated control law individually directing the robot's motion along the linear and rotational axes is shown in Figure 7.

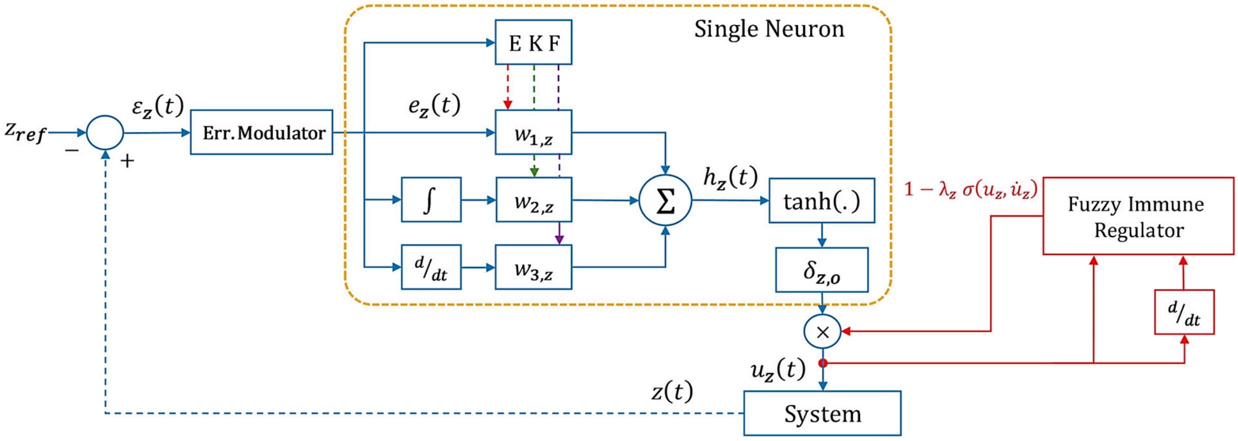

Figure 7. Schematic of the fuzzy-immune-regulated SN-APID control scheme.

The traditional two-input fuzzy-inference mechanism is used to realize the stimulation factor $\sigma(.)$, which utilizes a two-input qualitative rule base to describe the prescribed immunological defense system [45]. The fuzzy system utilizes the control variable $u_z(t)$ and its derivative $\dot{u}_z(t)$ as its inputs. Its output is the stimulation factor $\sigma\left(u_z, \dot{u}_z\right)$. Seven linguistic variables Positive Big (PB), Positive Medium (PM), Positive Small (PS), Zero (Z), Negative Small (NS), Negative Medium (NM), Negative Big (NB) are used to categorize the universe of the inputs and the output. Thus, the fuzzy implication is implemented by using 49 logical rules. The variances of the output and input variables are normalized within the range $[-1, 1]$. The stimulus $\sigma(.)$ is synthesized as per the immune regulation rules presented in Table 1 [45].

The fuzzy implication is carried out via the min–max inference technique expressed in (46).

$$\mu_i = min\left(\tau_i(u_z), \tau_i\left(\dot{u}_z\right)\right) \tag{46}$$

where i is the rule number, μ is the membership function's (MF) degree, and $\tau_i(.)$ is the triangular MF formulated as shown in (47).

Table 1. Fuzzy rule base for the stimulus factor [45].

u_z↓/$\dot{u}_z$→	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
NB	PB	PB	PM	PM	PS	PS	Z
NM	PB	PM	PM	PS	PS	Z	NS
NS	PM	PM	PS	PS	Z	NS	NS
Z	PM	PS	PS	Z	NS	NS	NM
PS	PS	PS	Z	NS	NS	NM	NM
PM	PS	Z	NS	NS	NM	NM	NB
PB	Z	NS	NS	NM	NM	NB	NB

$$\tau(\epsilon) = \begin{cases} 1 + \frac{\epsilon - c_i}{b_i^{-}}, & -b_i^{-} \le \epsilon - c_i \le 0 \\ 1 - \frac{\epsilon - c_i}{b_i^{+}}, & 0 \le \epsilon - c_i \le b_i^{+} \\ 0, & \text{other wise} \end{cases} \tag{47}$$

where ϵ represents the fuzzy input variable u_z or $\dot{u}_z$, and c_i, b_i^{+}, b_i^{-}, are the centroid, right-half width, and left-half width of the input MF, respectively. The fuzzy implication and aggregation are performed using symmetrical MFs. Figures 8 and 9 illustrate the input and output fuzzy MF waveforms, respectively. The centroid method of defuzzification is used to determine the crisp value of $\sigma(u_z, \dot{u}_z)$, as shown in (48), [45].

$$\sigma(u_z, \dot{u}_z) = \frac{\sum_{i=1}^{N} \mu_i \, c_i^o}{\sum_{i=1}^{N} \mu_i} \tag{48}$$

where c_i^o is the output MF's centroid and $N = 49$ is the total number of rules. Using this method, the nonlinear stimulation factor $\sigma(u_z, \dot{u}_z)$ is recalculated and updated online after each sample period. The proposed fuzzy-immune-regulated SN-APID control laws for the x-axis, y-axis, and orientation are shown below.

$$u_x(t) = \delta_{x,o} \, (1 - \lambda_x \sigma(u_x, \dot{u}_x)) \tanh(h_x(t)) \tag{49}$$

$$u_y(t) = \delta_{y,o} (1 - \lambda_y \sigma(u_y, \dot{u}_y)) \tanh(h_y(t)) \tag{50}$$

$$u_\theta(t) = \delta_{\theta,o} \, (1 - \lambda_\theta \sigma(u_\theta, \dot{u}_\theta)) \tanh(h_\theta(t)) \tag{51}$$

The parameters λ_x, λ_y, and λ_θ are tuned by minimizing the cost function J, mentioned in (19). These parameters are restricted within the range [0, 1]. The optimized values of these parameters are: $\lambda_x = 0.84$, $\lambda_y = 0.75$, and $\lambda_\theta = 0.68$. The SN-APID signal modulated by using the fuzzy-immune regulator (FIR) is referred to as "FSN-APID". The schematic diagram of the FSN-APID is illustrated in Figure 10.

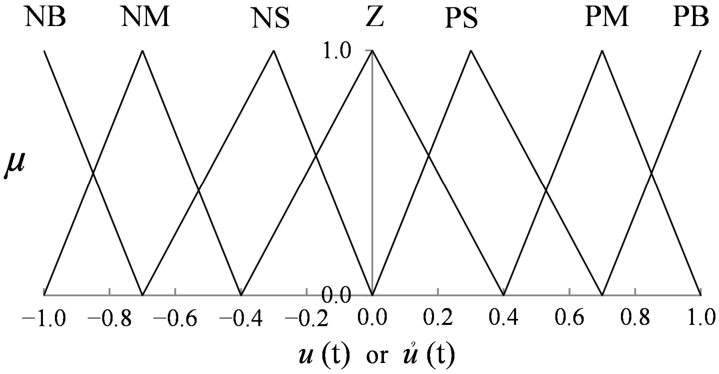

Figure 8. Input fuzzy MFs of u and $\dot{u}$.

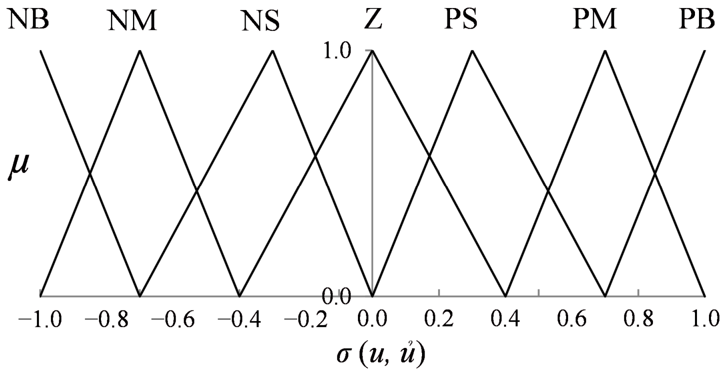

Figure 9. Output fuzzy MFs of σ.

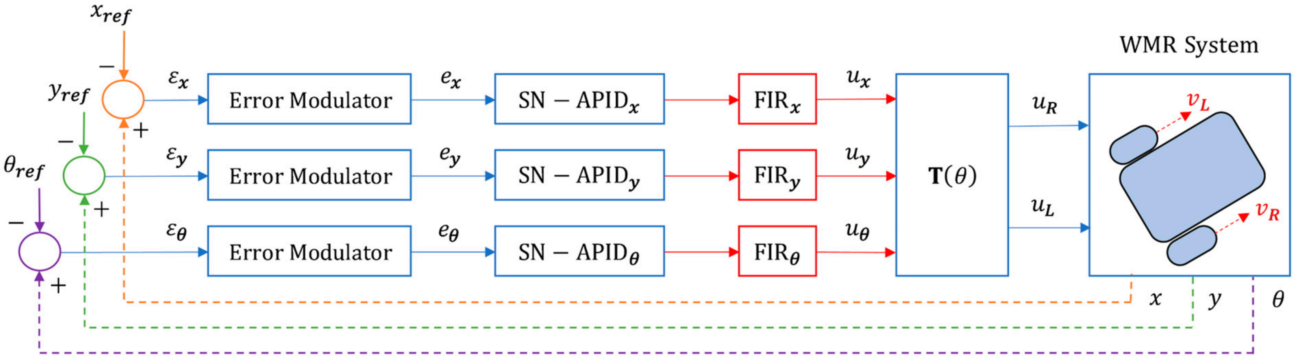

Figure 10. Schematic diagram of the FSN-APID control architecture.

4. Results and Discussion

The results of the customized hardware experiments performed to assess the proposed controller's effectiveness under the influence of bounded environmental disturbance are discussed in this section.

4.1. Experimental Setup

The experiments were performed using a differentially driven lawn-mowing robot setup. The lawn-mowing robot chassis is demonstrated in Figure 11. It comprises two motorized wheels. The rotational speed of each wheel is measured via dedicated rotary encoders attached to the shafts of the respective motors. The raw encoder measurements are acquired by an 8-bit embedded microcontroller at a sampling rate of 0.2 kHz [55]. The robot's trajectory tracking profile, and hence, the error is recorded with the aid of an inertial measurement unit, commissioned at the top of the robot assembly. For velocity control computations, the digital sensor measurements are serially communicated (at a baud rate of 9600 bps) to a customized computer application that is developed using the MATLAB-Simulink R2018b Software (MATLAB 9.5 version), via an onboard wireless Bluetooth transceiver. The prescribed computations are executed on a 1.8 GHz and 64-bit personal computer with 8.0 GB RAM. Customized first-order (low-pass) Butterworth filters are applied to the WMR's velocity control signals, $u_R(t)$ and $u_L(t)$, to smoothly attenuate the high-frequency components. The filter's cutoff frequency is set to 30 Hz in order to eliminate unwanted frequencies while maintaining the control signal's integrity. The WMR's pilot test runs were used to experimentally determine this cutoff frequency setting. The robot's trajectory tracking profile, as well as its tracking error along the x-axis, y-axis, and orientation were captured in real time for graphical visualization via the aforementioned sensors. The onboard microcontroller receives the filtered velocity control signals, for each of the two motors, serially after each sample instant and transforms them into corresponding pulse-width-modulated (PWM) signal. The computed PWM signals are amplified and then applied to the onboard H-Bridge DC motor driver circuits.

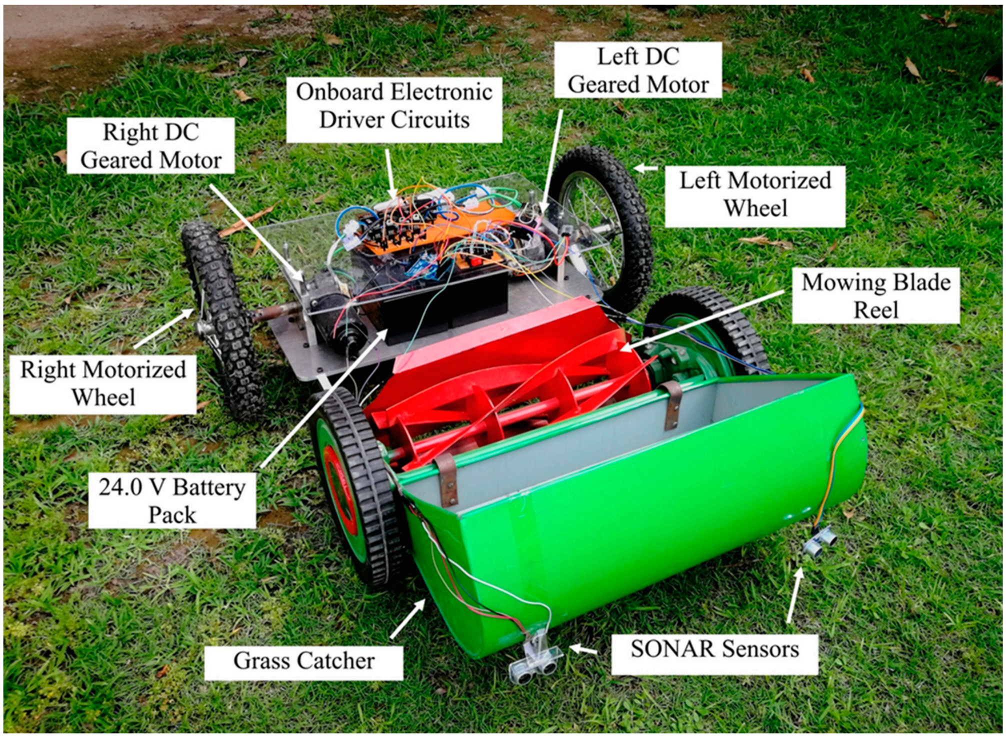

Figure 11. Lawn mowing robot chassis used for experimental analysis.

These motor driver circuits are used to actuate their respective 90 Watt and 24.0 V permanent magnet DC geared motors. The entire robot is powered by a +24.0 V battery pack, which is assembled by serially connecting two 12.0 V, 10.0 AH sealed lead-acid batteries. The motor drivers are retrofitted with snubbing circuits to compensate for the flyback effects.

4.2. Tests and Results

The experimental trials were carried out by tracking a rose-curve trajectory, which was generated after each step time T as shown in (52), [36].

$$x_r = a\cos(0.04\pi T),\ y_r = a\sin(0.04\pi T),\ \theta_r = \frac{\pi}{4} \tag{52}$$

where $a = 4 + \cos(0.2\pi T)$

The rose-curve reference trajectory, shown in Figure 12, was chosen owing to the complexity of its geometric features. This trajectory does not require precise polynomial formulations to guarantee smooth transitions. However, the variations in curvature and transitions along the path help to better ascertain the disturbance rejection and nonlinear dynamics handling capability of the designed tracking controller. The following three customized experiments were performed to benchmark the efficacies of the proposed FSN-APID control technique in comparison to the baseline SN-APID control technique.

A. *Trajectory tracking under nominal conditions*: This test case examines the controller's time-varying reference trajectory tracking accuracy in a disturbance-free grassy field. Each control procedure is tasked to track the rose-curve trajectory expressed above. The corresponding state variations, exhibited by each control procedure, are displayed in Figure 13. The error profiles of each state variable are shown in Figure 14. The resulting rose-curve trajectory tracking plots are displayed in Figure 15.

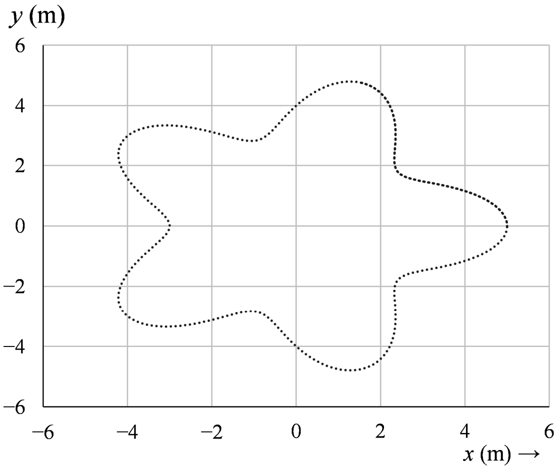

Figure 12. The rose-curve reference trajectory.

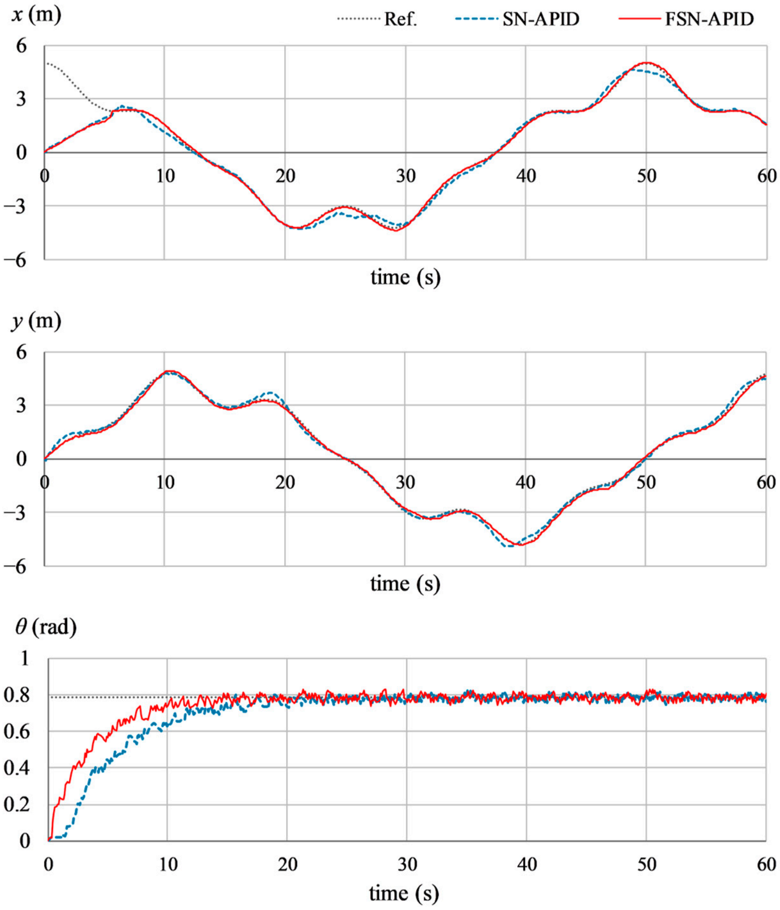

Figure 13. State variations under nominal conditions.

B. *Trajectory tracking under impulsive disturbances*: This experiment analyzes the controller's resilience and ability to recover from the impact of sudden collisions, sudden wheel skidding, motor voltage spikes, and uneven terrains or surface bumps. The said disturbance conditions are emulated by injecting software-generated displacement pulses, of magnitude 1.0 m and a period of 1.5 s, in x_{ref} and y_{ref}. These pulses are injected randomly. A pulse of magnitude 0.2 rad and a duration of 1.5 s is added in θ_{ref} at t ≈ 30 s. The experiment is performed in a grassy field. The state variations corresponding error profiles are displayed in Figures 16 and 17, respectively. The resulting rose-curve trajectory tracking profiles are shown in Figure 18.

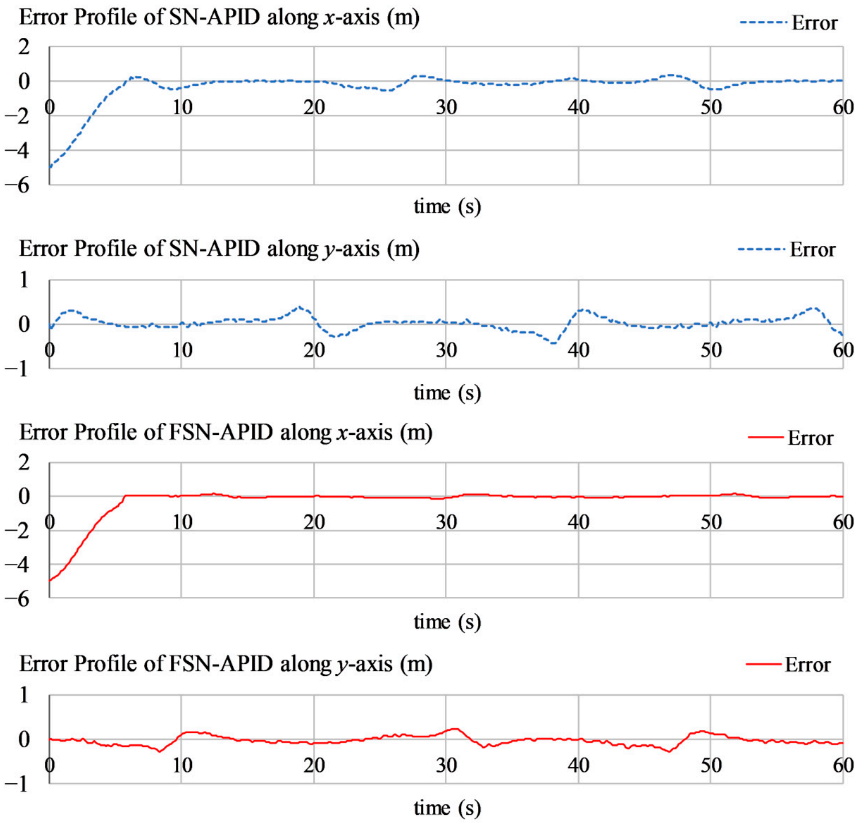

Figure 14. Error profile of each state variable under nominal conditions.

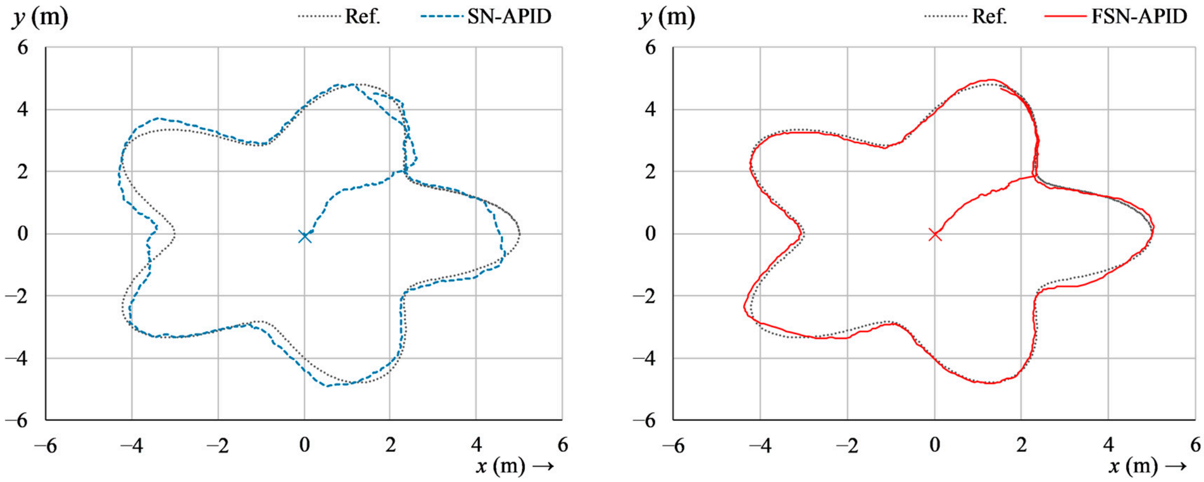

Figure 15. Rose-curve tracking profiles under nominal conditions.

C. *Trajectory tracking under step disturbances*: This test case assesses the controller's capacity to compensate for the disruptions contributed by the sudden changes in the robot's payload, transition of the robot from a flat surface to a ramped surface, changes in the mechanical friction encountered by the wheels, and battery voltage drop. The said experiment is performed by injecting software-generated displacement step signals of magnitude 0.75 m in x_{ref} and y_{ref}, at $t \approx 25.0$ s mark. The experiment is performed in a grassy field. The corresponding state variations, exhibited by each control procedure, are displayed in Figure 19. The error profiles are illustrated in Figure 20. The resulting rose-curve trajectory tracking plots are displayed in Figure 21.

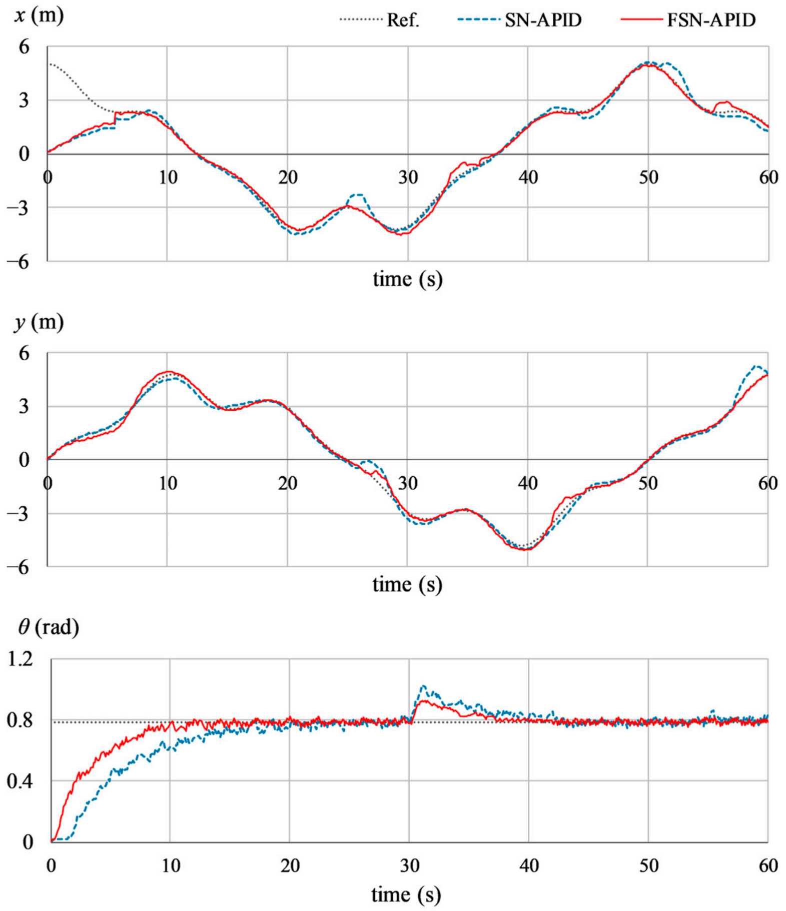

Figure 16. State variations under impulsive disturbances.

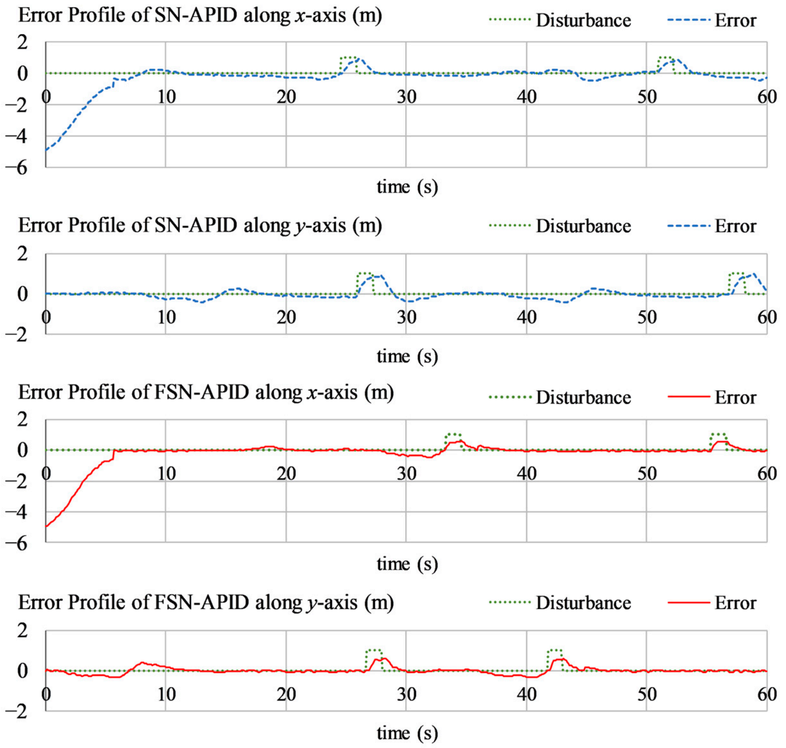

Figure 17. Error profile of each state variable under impulsive disturbances.

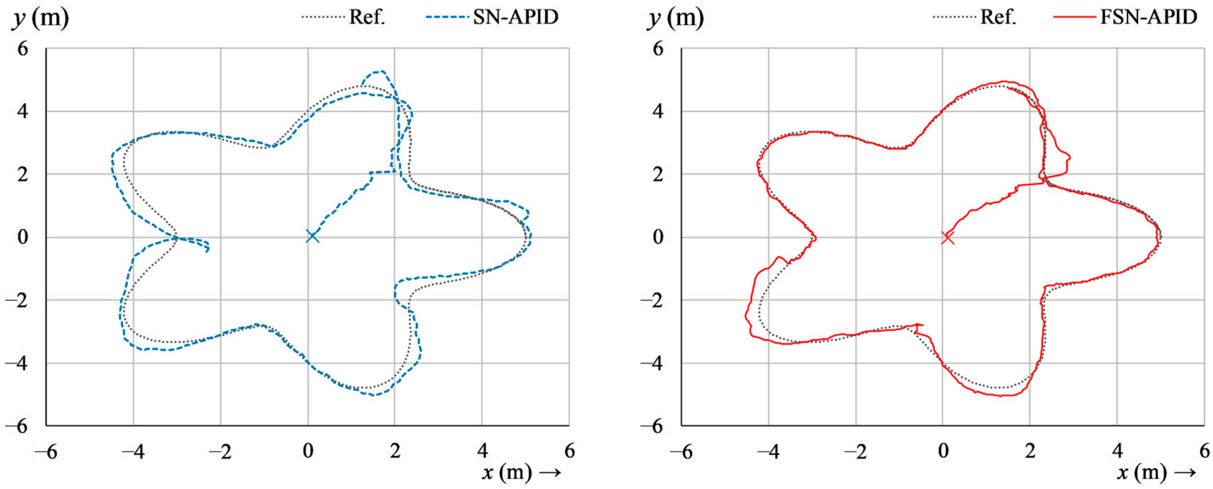

Figure 18. Rose-curve tracking profiles under impulsive disturbances.

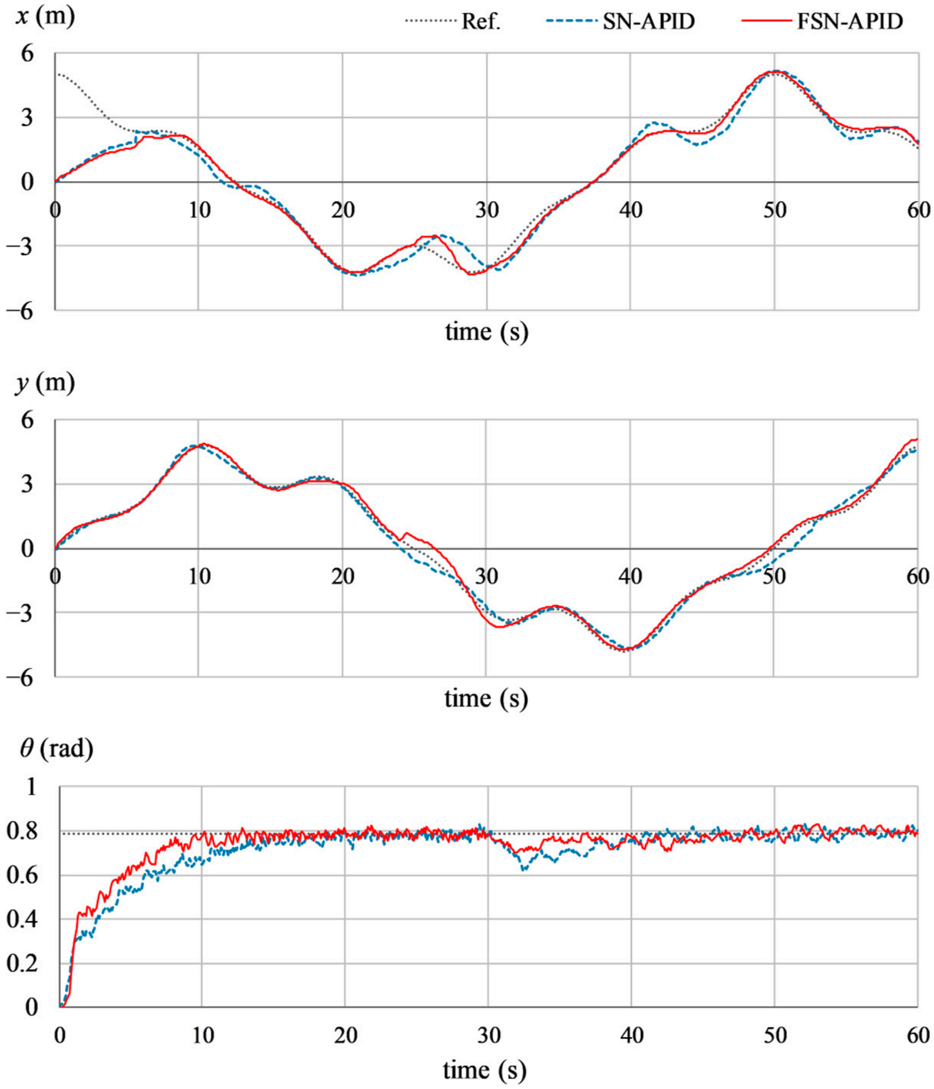

Figure 19. State variations under step disturbances.

D. *Trajectory tracking under randomized step disturbances*: This experiment assesses the controller's resilience against the perturbations caused by random changes in the robot's payload or unprecedented terrain irregularities. This experiment is performed by injecting software-generated displacement step signals of ±0.75 m in magnitude and 15 s in duration in x_{ref} and y_{ref} at random. The experiment is performed in a grassy field. The corresponding state variations, exhibited by each control procedure, are displayed in Figure 22. The disturbance signals and state error profiles are illustrated in Figure 23. The resulting rose-curve trajectory tracking plots are displayed in Figure 24.

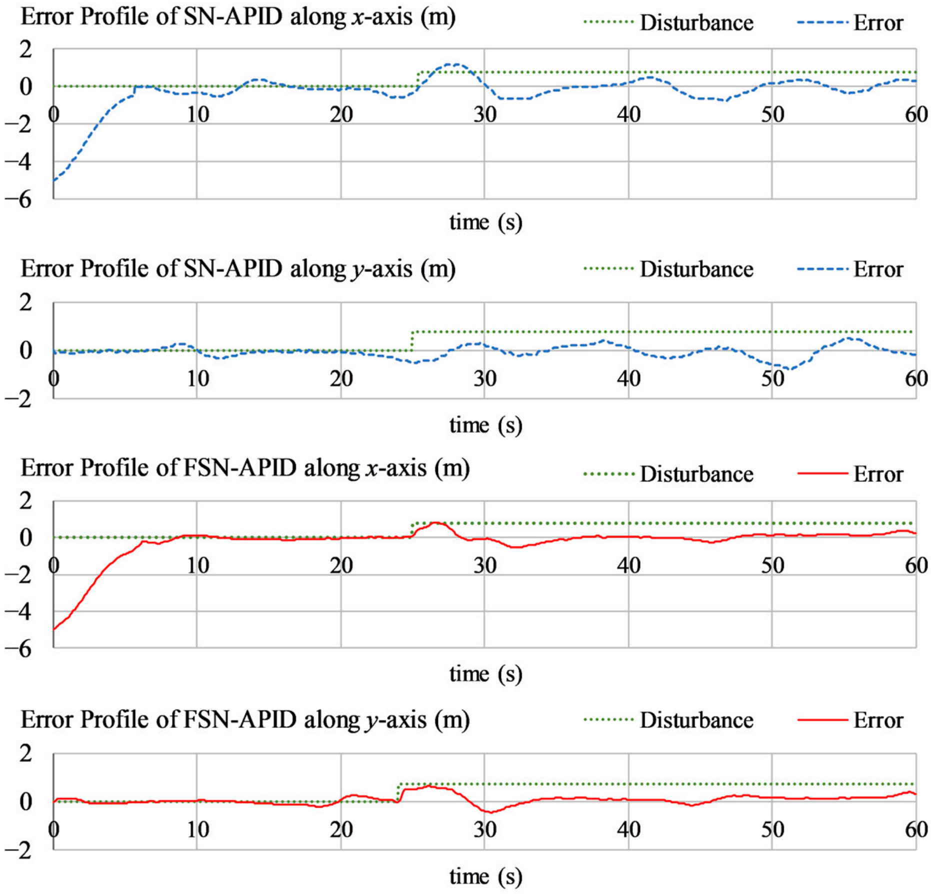

Figure 20. Error profile of each state variable under step disturbances.

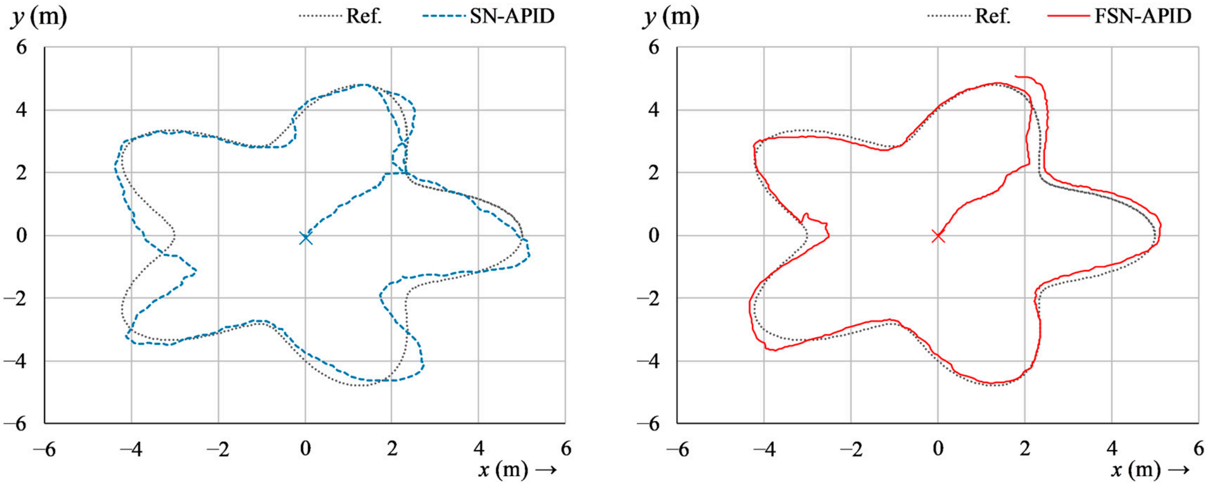

Figure 21. Rose-curve tracking profiles under step conditions.

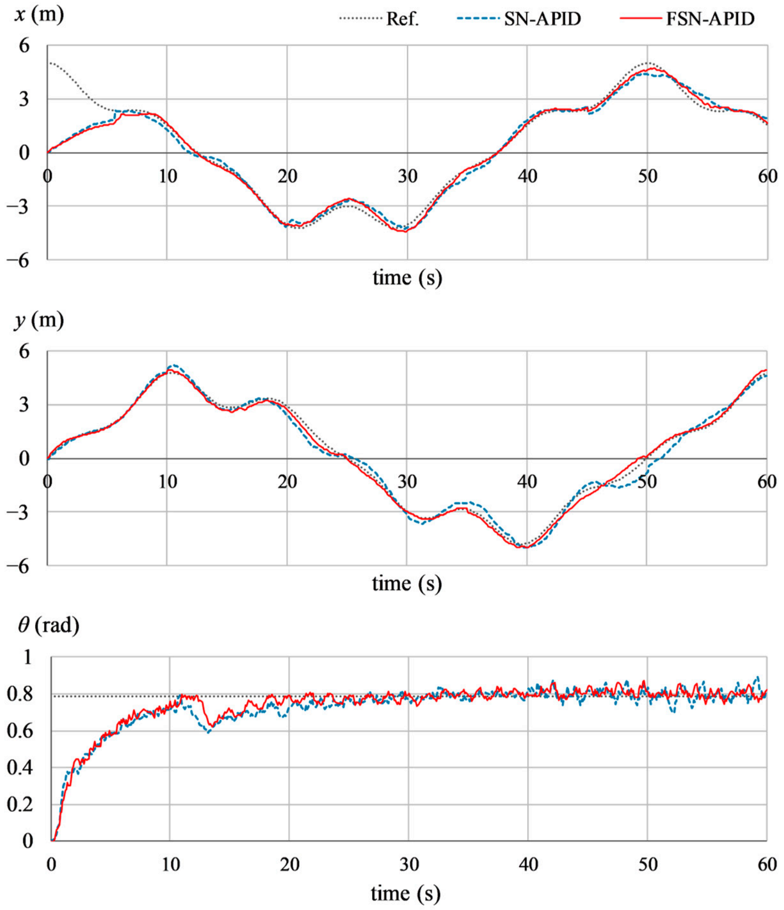

Figure 22. State variations under randomized step disturbances.

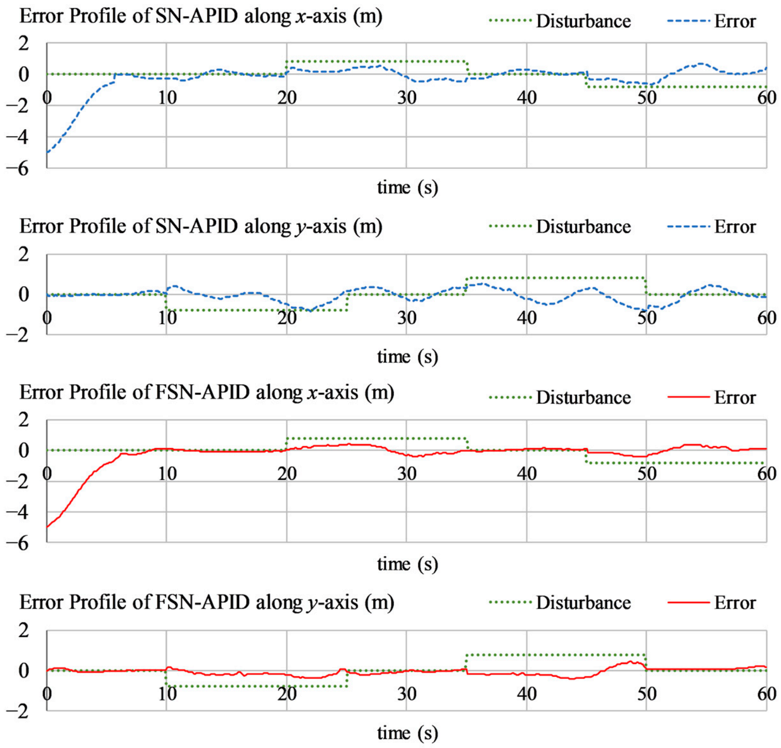

Figure 23. Error profile of each state variable under randomized step disturbances.

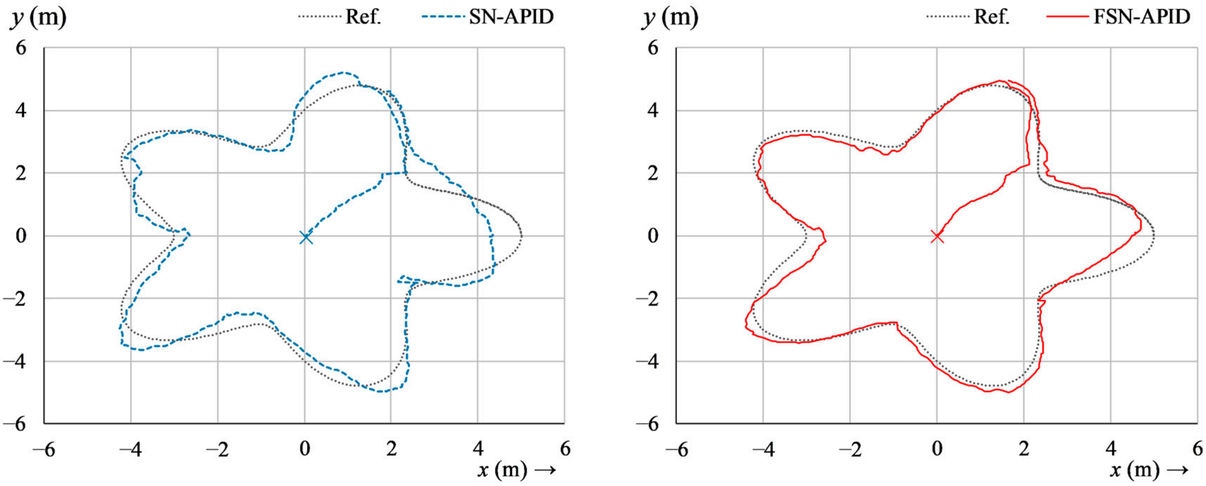

Figure 24. Rose-curve tracking profiles under randomized step conditions.

4.3. Discussion

The results of the aforementioned experiments are examined using the following key performance indicators (KPIs).

- RMSE: The root-mean-squared value of e_x, e_y, and e_θ. It is evaluated as $\sum\sqrt{\frac{(e_z(n))^2}{n}}$.
- $\mathrm{t_{set}}$: The time taken by the response along the x-, y-, and about the θ-axis to settle within ±2% of time-varying reference after the initial startup.
- MAD: The maximum absolute deviation in the response along the x-, y-, and about the θ-axis contributed by the external disturbances or environmental uncertainties.
- $\mathrm{t_{rec}}$: The amount of time it takes for the response to recover and stabilize within ±2% of the time-varying reference after being disturbed.

The results of the hardware experiments (A–D) verify that, in comparison to the baseline SN-APID controller, the proposed FSN-APID controller exhibits a strong disturbance rejection while maintaining an accurate trajectory tracking performance, validating the practical applicability of the proposed approach. The experimental results of the FSN-APID controller are quantitatively analyzed and compared to those of the baseline SN-APID controller for experiments A, B, and C in Table 2.

In Experiment A (Figures 13–15), the SN-APID control technique demonstrates moderate tracking accuracy despite the adaptive nature of the neural networks. While tracking the rose-curve trajectory, the SN-APID controller is observed to struggle with variations in the curvature of the reference path. In contrast, the proposed FSN-APID control exhibited relatively stronger damping against path deviation and a significantly improved tracking accuracy, despite the intricate geometric features of the reference trajectory. Moreover, the transient and settling times of the FSN-APID controller are relatively better than those of the baseline SN-APID controller.

In Experiment B (Figures 16–18), the SN-APID control technique manifests a moderate adaptability to abrupt dynamic changes. Although the single-neuronal model self-adjusts the critical controller parameters, the aforementioned controller exhibits a relatively slower error convergence rate. The proposed FSN-APID control scheme primarily leverages the single-neuron PID controller's flexibility to adaptively modify gains. Additionally, the fuzzy-immune regulator robustly reconfigures the control response on the fly to supplement the system's adaptability and strengthen its damping against impulse-like disturbances by quickening the controller's convergence speed and attenuating the peak overshoots in the state response(s).

Table 2. Overview of experimental findings.

Test	KPI		State	Control Technique		Improvement (%)
	Symbol	Unit		SN-APID	FSN-APID	
A	RMSE	m.	x	0.038	0.036	5.3
		m.	y	0.006	0.004	33.3
		rad.	θ	0.008	0.006	25.0
	MAD	m.	x	0.541	0.225	58.4
		m.	y	0.424	0.289	31.8
		rad.	θ	0.056	0.048	14.3
	t_{set}	s	x	9.8	6.6	32.6
		s	y	7.6	5.4	28.9
		s	θ	15.1	11.3	25.1
B	RMSE	m.	x	0.040	0.038	5.0
		m.	y	0.011	0.007	36.4
		rad.	θ	0.009	0.007	22.2
	MAD	m.	x	0.930	0.601	35.4
		m.	y	0.975	0.628	35.6
		rad.	θ	0.239	0.144	39.7
	t_{rec}	s	x	3.5	2.7	22.9
		s	y	5.4	3.1	42.6
		s	θ	9.7	7.2	25.8
C	RMSE	m.	x	0.041	0.039	4.9
		m.	y	0.010	0.008	20.0
		rad.	θ	0.006	0.005	16.7
	MAD	m.	x	1.201	0.835	30.5
		m.	y	0.794	0.648	18.4
		rad.	θ	0.162	0.092	43.2
	t_{rec}	s	x	8.6	5.4	37.2
		s	y	8.0	7.1	11.2
		s	θ	9.8	7.2	26.5
D	RMSE	m.	x	0.039	0.038	2.6
		m.	y	0.012	0.009	25.0
		rad.	θ	0.006	0.005	16.7
	MAD	m.	x	0.679	0.447	34.2
		m.	y	0.841	0.448	46.7
		rad.	θ	0.105	0.086	18.1
	t_{rec}	s	x	8.2	6.8	17.1
		s	y	7.7	7.2	6.5
		s	θ	8.8	7.5	14.8

In Experiment C (Figures 19–21), the SN-APID control technique demonstrates a reasonable disturbance rejection behavior. Although its online self-learning ability enables it to compensate for external disturbances, it still shows oscillations under environmental indeterminacies. The proposed FSN-APID controller, once again, demonstrates a significantly robust disturbance rejection capacity. It effectively rejects the parametric disruptions with minimal oscillations.

In Experiment D (Figures 19–21), the SN-APID control technique demonstrates a deficient disturbance compensation capacity. Despite its inherent self-learning capability, the SN-APID control law exhibits relatively large fluctuations in the trajectory tracking response, whereas, in the proposed FSN-APID control scheme, the fuzzy-immune regulator flexibly manipulates the control trajectory, enabling the system to demonstrate a relatively superior adaptability and stronger resilience to randomized step disturbances.

The proposed enhancement of the EKF-driven single-neuron PID controller with the immune regulation mechanism, allows the system to remain effective against disturbances by self-adjusting control parameters, thus reducing tracking error. The SN-APID control law adaptively modifies the PID coefficients as per the system's error variations and environmental perturbations. The weights of the SN-APID controller (which correspond to the PID coefficients) are reconfigured online by the EKF. This error-driven adaptive adjustment of weights aids in maintaining high tracking accuracy and reduces error under varying conditions. Additionally, the bio-inspired fuzzy-immune mechanism adaptively adjusts the scaling gain of the SINN, facilitating the system's quick response to disruptions. The adaptive self-learning introduced by the fuzzy-immunological computations flexibly inflates the scaling gain as the error enlarges, yielding prompt corrective actions, and decreases the said gain as the error reduces, which dampens the overshoots and oscillations.

The proposed FSN-APID control scheme makes a trade-off between performance and computational simplicity when benchmarked against a ubiquitous fixed-gain PID controller. A fixed-gain PID is significantly simpler to implement, which reduces the system's overall complexity. However, it inherently lacks the robustness to effectively handle nonlinear disruptions and time-variant behaviors linked with the system due to its static parameter setting. The proposed FSN-APID controller, on the other hand, adapts in real-time to system changes providing improved tracking accuracy, better compensation of disturbances, quicker convergence rate, and stronger damping against oscillations. However, it achieves this superior performance at the cost of increased computational demands and the implementation complexity required for real-time parameter adjustment and learning. Nevertheless, the computational requirements of the proposed controller can be effectively managed using modern digital computing resources, preventing any excessive computational load.

5. Statistical Analysis of Experimental Findings

The experimental results are statistically examined in this section to verify the performance enhancement of the proposed controller over the baseline scheme, as indicated in Section 4.3. Confidence interval analysis and hypothesis testing are used to statistically examine the quantitative results of experiments A, B, C, and D, which are contributed by each control scheme.

5.1. Confidence Interval Analysis

The confidence intervals (CIs) are essential for quantifying the dependability, degree of uncertainty, and practical importance of the proposed FSN-APID control procedure. As applied in this study, a 95% CI means that the (true) average absolute values of the tracking error variations (e_x, e_y, and e_θ) have a 95% chance of falling within the specified range. Table 3 displays the CI analysis of the state variations contributed by each unique control scheme under experiments A, B, C, and D. The CI analysis confirms the enhanced robustness of the FSN-APID controller by indicating the consistency of performance improvements across all testing scenarios.

Table 3. Confidence interval analysis.

Test	**Tool (m)**	$\lvert e_x \rvert$		$\lvert e_y \rvert$		$\lvert e_\theta \rvert$	
		SN-APID	**FSN-APID**	**SN-APID**	**FSN-APID**	**SN-APID**	**FSN-APID**
A	Mean	0.16	0.092	0.11	0.088	0.087	0.058
	Median	0.10	0.066	0.065	0.074	0.018	0.017
	Standard Dev.	0.17	0.14	0.099	0.062	0.17	0.12
	CI (95%)	0.014	0.011	0.008	0.005	0.014	0.010
	Upper CI (95%)	0.174	0.103	0.118	0.093	0.101	0.068
	Lower CI (95%)	0.146	0.081	0.102	0.083	0.073	0.048

Table 3. *Cont.*

Test	Tool (m)	$\lvert e_x \rvert$		$\lvert e_y \rvert$		$\lvert e_\theta \rvert$	
		SN-APID	**FSN-APID**	**SN-APID**	**FSN-APID**	**SN-APID**	**FSN-APID**
B	Mean	0.20	0.11	0.18	0.11	0.11	0.063
	Median	0.14	0.058	0.14	0.038	0.042	0.017
	Standard Dev.	0.18	0.14	0.19	0.14	0.18	0.13
	CI (95%)	0.016	0.012	0.016	0.011	0.014	0.010
	Upper CI (95%)	0.216	0.112	0.196	0.121	0.124	0.073
	Lower CI (95%)	0.184	0.098	0.164	0.099	0.096	0.053
C	Mean	0.33	0.16	0.19	0.15	0.079	0.057
	Median	0.29	0.11	0.15	0.12	0.029	0.022
	Standard Dev.	0.25	0.17	0.16	0.14	0.13	0.12
	CI (95%)	0.021	0.015	0.013	0.011	0.010	0.009
	Upper CI (95%)	0.351	0.175	0.203	0.161	0.089	0.066
	Lower CI (95%)	0.309	0.145	0.177	0.139	0.069	0.048
D	Mean	0.26	0.17	0.25	0.15	0.051	0.038
	Median	0.26	0.12	0.20	0.12	0.033	0.022
	Standard Dev.	0.17	0.14	0.20	0.11	0.052	0.046
	CI (95%)	0.014	0.011	0.016	0.008	0.0043	0.0038
	Upper CI (95%)	0.274	0.181	0.266	0.158	0.0553	0.0418
	Lower CI (95%)	0.246	0.159	0.234	0.142	0.0467	0.0342

5.2. Hypothesis Testing

Hypothesis testing is used to evaluate the degree to which the suggested FSN-APID controller optimizes the trajectory tracking accuracy in comparison to the baseline SN-APID controller. It involves the statistical analysis of an assumption (the hypothesis), as discussed below, about the state variations under the influence of the proposed controller.

- *Null Hypothesis* (H_0): The FSN-APID controller does not substantially reduce the trajectory tracking errors compared to the baseline SN-APID controller.
- *Alternative Hypothesis* (H_1): The FSN-APID controller substantially reduces the trajectory tracking errors compared to the baseline SN-APID controller.

The *t*-test examines the hypothesis for the experimental data sets of SN-APID and FSN-APID controllers. The significance level for this test is set at 0.05. Table 4 provides a summary of the *t*-test findings. It is observed that the t-stat value is greater than the corresponding t-critical value in each state variable. This observation concludes that the trajectory tracking accuracy of the FSN-APID controller is superior to the baseline controller, and hypothesis H_0 is thus rejected. Furthermore, in each instance, the *p*-value is smaller than the significance level (0.05), supporting the choice to reject H_0 in favor of H_1.

Table 4. Hypothesis testing results.

Test	Tool (m)	$\lvert e_x \rvert$	$\lvert e_y \rvert$	$\lvert e_\theta \rvert$
A	t-stat	7.57	5.38	3.53
	t-critical	1.96	1.96	1.96
	p-value	7.64×10^{-14}	9.15×10^{-8}	4.32×10^{-4}
B	t-stat	2.08	7.34	5.36
	t-critical	1.96	1.96	1.96
	p-value	3.74×10^{-2}	4.32×10^{-13}	9.94×10^{-8}
C	t-stat	2.96	4.77	3.18
	t-critical	1.96	1.96	1.96
	p-value	3.12×10^{-3}	2.09×10^{-6}	1.50×10^{-3}

Table 4. *Cont.*

Test	Tool (m)	$\lvert e_x \rvert$	$\lvert e_y \rvert$	$\lvert e_\theta \rvert$
D	t-stat	4.75	11.20	4.22
	t-critical	1.96	1.96	1.96
	p-value	2.28×10^{-6}	2.10×10^{-27}	2.69×10^{-5}

6. Ablation Study of the Proposed Controller

This section presents the following three ablation studies of the proposed FSN-APID controller by systematically varying its key parameters (or components) to analyze their influence on the controller's time-domain performance, particularly in terms of tracking precision and robustness.

6.1. Ablation of the Fuzzy Inference Parameters

To assess the effectiveness of the selected fuzzy inference parameters as well as their impact on the behavior of the FSN-APID controller, an ablation study was carried out by introducing a 10.0% decrement and a 10.0% increment in the optimized values of $\delta_{x,o}$, $\delta_{y,o}$, $\delta_{z,o}$, λ_x, λ_y, and λ_z as prescribed in Sections 2.4.1 and 3, respectively. For each altered parameter set, experiment A is conducted to track the rose-curve trajectory of (49) under nominal conditions. The state variations, exhibited by the FSN-APID controller under the influence of each parameter set, are displayed in Figure 25. The error profiles corresponding to each state variable are shown in Figure 26. The rose-curve trajectory tracking profiles yielded by each parameter set are displayed in Figure 27. The experimental findings are summarized in Table 5.

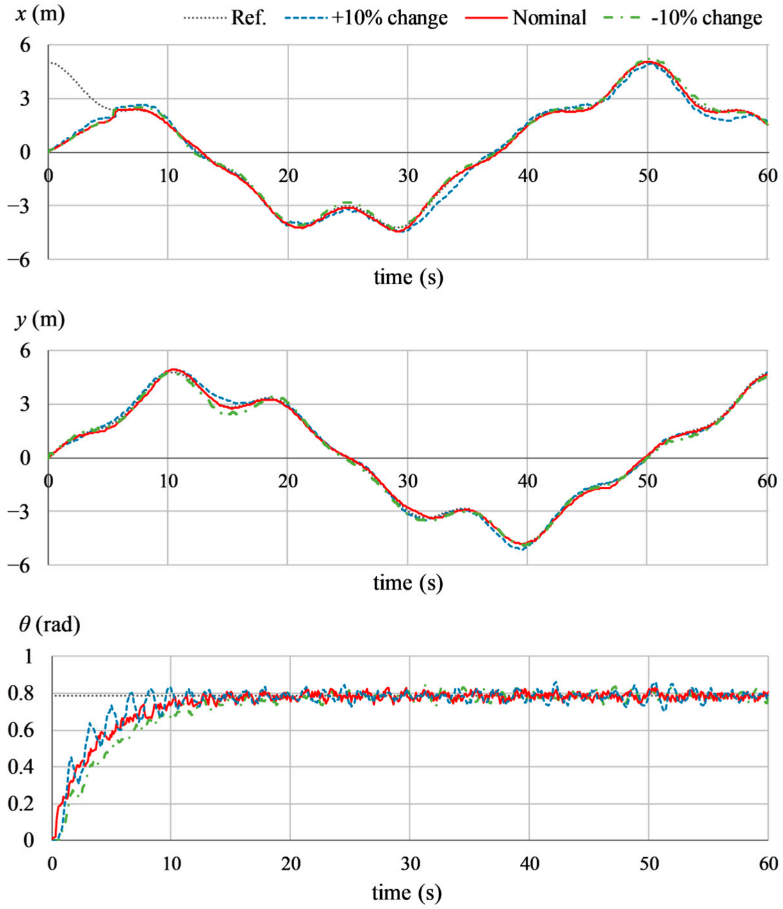

Figure 25. State variations under different fuzzy parameter settings of FSN-APID controller.

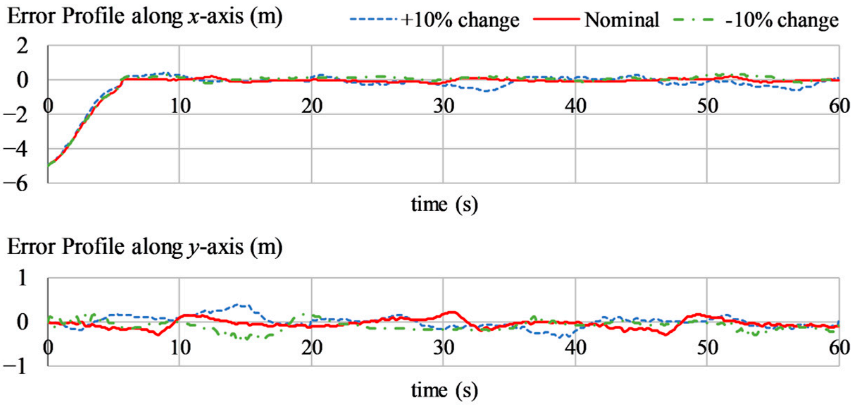

Figure 26. Error profile of each state variable's different fuzzy parameter settings of FSN-APID controller.

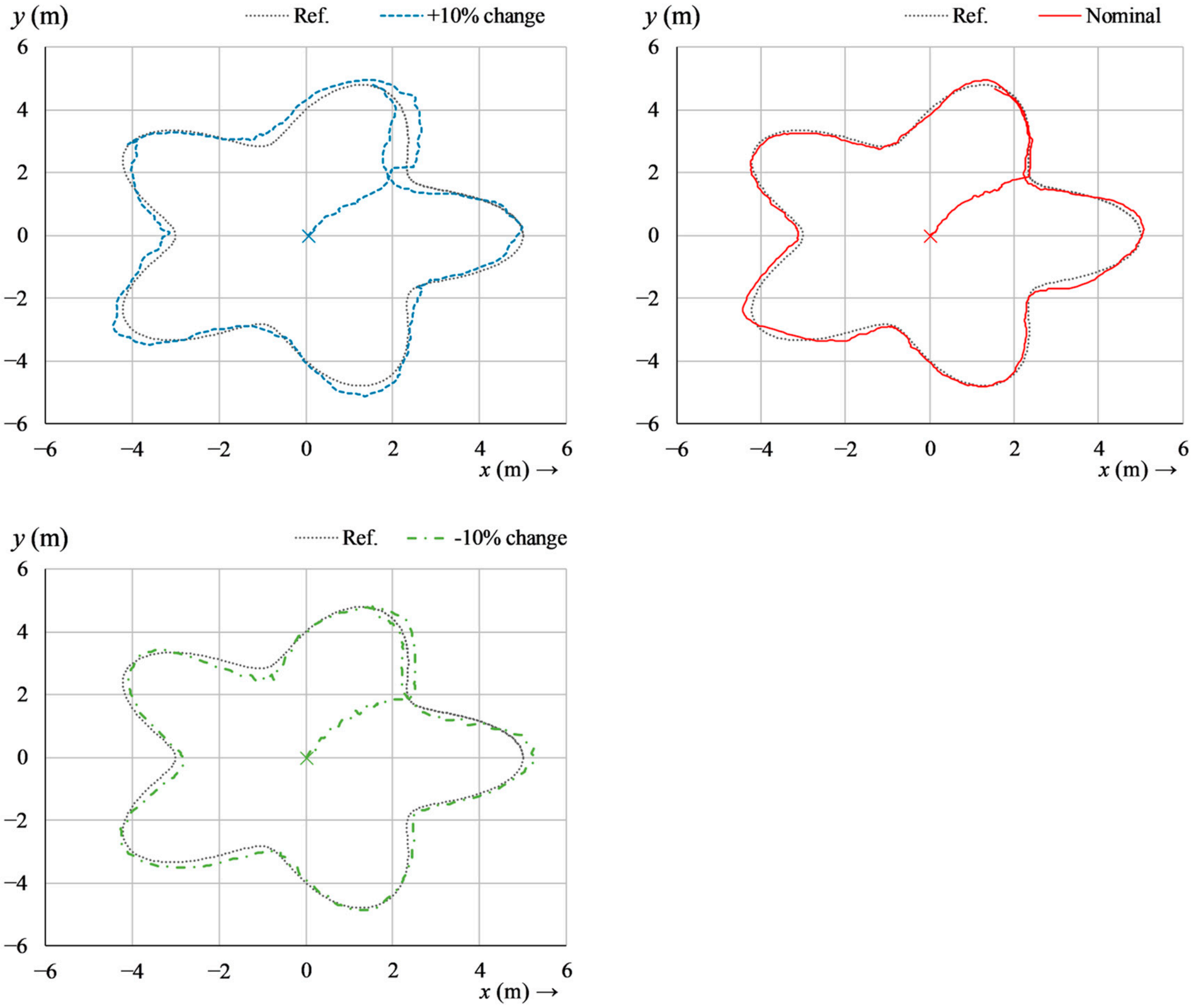

Figure 27. Rose-curve tracking profiles under different fuzzy parameter settings of FSN-APID controller.

The results validate that the nominal parameter settings yield the optimum tracking accuracy with reasonable response speed. The parameter settings with a 10.0% increment offer an aggressive control behavior with a slightly faster response speed at the cost of persistent oscillations in the response. The parameter settings with a 10.0% decrement deliver a slower error convergence speed.

Table 5. Summary of results under different fuzzy parameter settings of FSN-APID controller.

Test	KPI		State	Fuzzy Parameter Set		
	Symbol	**Unit**		**+10% Change**	**Nominal**	**−10% Change**
A	RMSE	m.	x	0.037	0.036	0.038
		m.	y	0.005	0.004	0.006
		rad.	θ	0.007	0.006	0.006
	MAD	m.	x	0.317	0.225	0.361
		m.	y	0.352	0.289	0.319
		rad.	θ	0.058	0.048	0.085
	t_{set}	s	x	6.3	6.6	6.8
		s	y	5.2	5.4	5.5
		s	θ	9.8	11.3	13.6

6.2. Evaluation of Different Activation Functions

This test is used to investigate the inclusion of various activation functions in the single-neuron model of the FSN-APID controller to minimize oscillations. Three commonly used odd-symmetric activation functions are tested; namely, hard hyperbolic tangent function (hard-Tanh) [56], typical Tanh (originally used in this study), and soft-sign function [57]. These functions naturally squash their inputs into the range $[-1, 1]$. These three functions are formulated in (53).

$$p_{1,z}(t) = \begin{cases} +1\ if\ h_z(t) > a \\ h_z(t) \quad otherwise \\ -1\ if\ h_z(t) < b \end{cases},\ p_{2,z}(t) = \tanh(h_z(t)),\ p_{3,z}(t) = \frac{h_z(t)}{1 + |h_z(t)|} \tag{53}$$

The values of $a = 1.18$ and $b = -1.18$ are empirically selected by trial and error during the preliminary test runs. The waveforms of the chosen activation functions are illustrated in Figure 28. To examine the efficacy of each activation function, experiment A is performed to track the rose curve trajectory of (49) under nominal conditions. The state variations, exhibited by the FSN-APID controller under the influence of each activation function, are displayed in Figure 29. The error profiles corresponding to each state variable are shown in Figure 30. The rose-curve trajectory tracking profiles with respect to each activation function are illustrated in Figure 31. The experimental findings are summarized in Table 6. The hard-Tanh function is simplest to compute; however, its abrupt clipping inevitably introduces oscillations in the response. The soft sign function has a slower saturation rate as compared to Tanh, which typically prevents oscillations; however, it also inevitably reduces the system's responsiveness, which leads to offsets in the response. These behaviors are also validated in the results. The typical Tanh exhibits optimum tracking accuracy with minimal oscillations and manageable computational complexity.

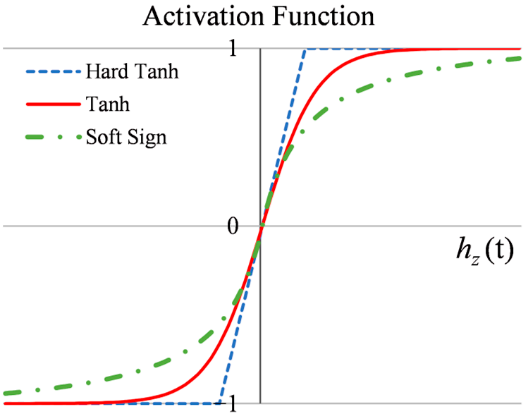

Figure 28. Waveforms of the activation functions used for the ablation study.

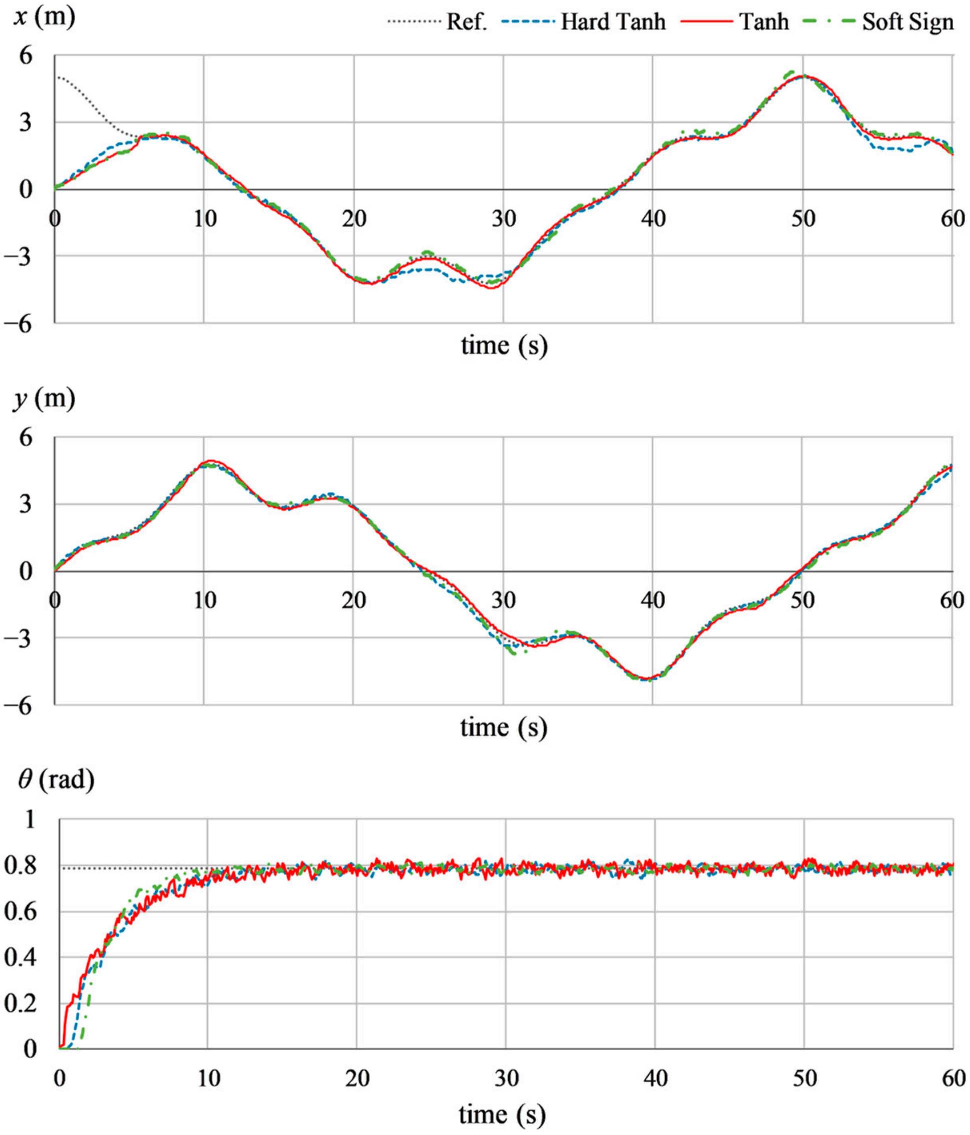

Figure 29. State variations under different activation functions of FSN-APID controller.

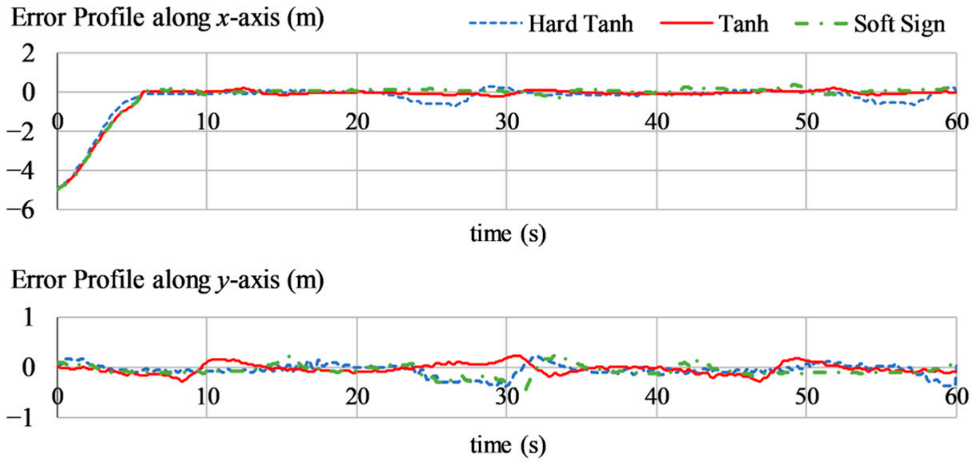

Figure 30. Error profile of each state variable under different activation functions of FSN-APID controller.

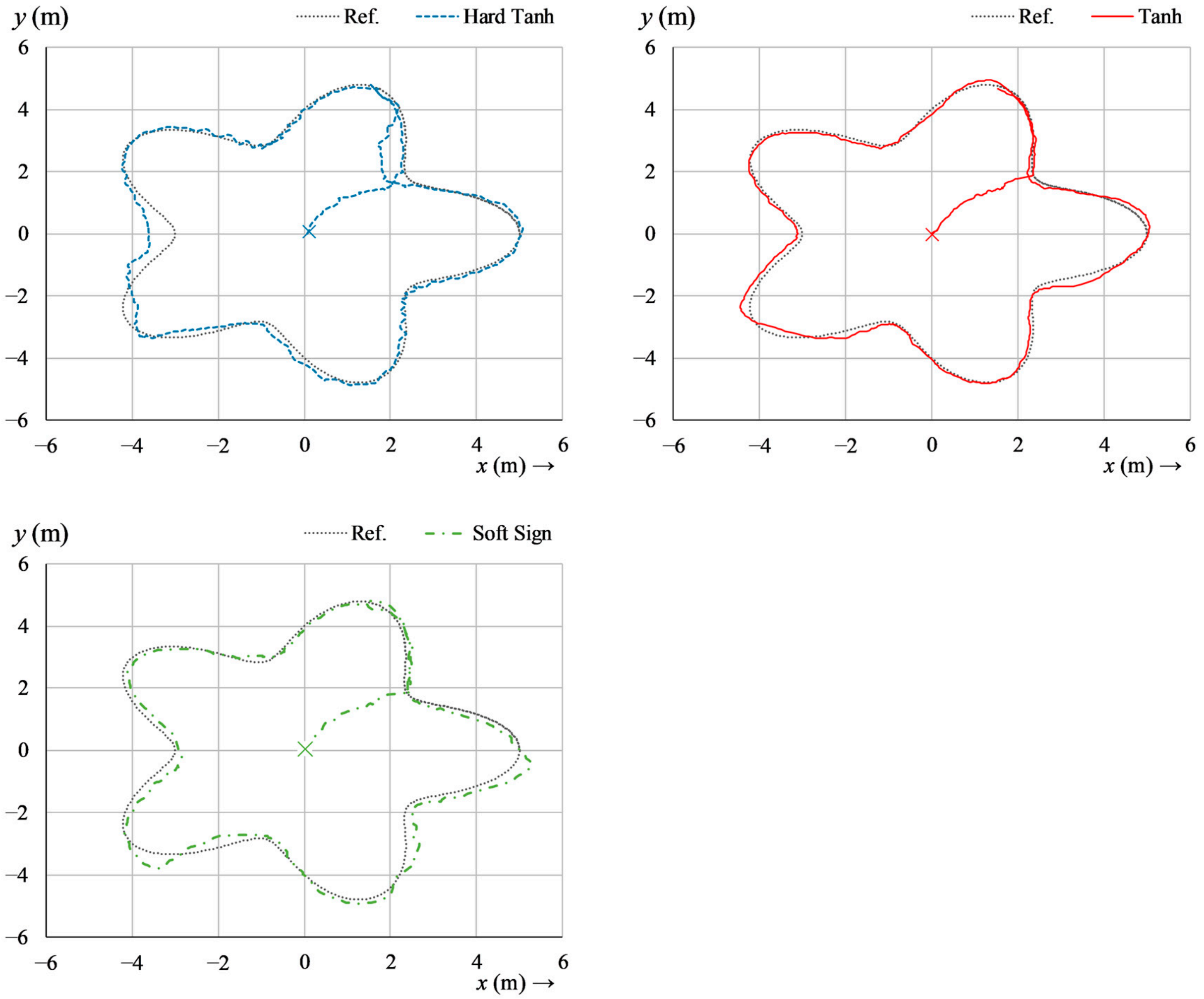

Figure 31. Rose-curve tracking profiles under different activation functions of FSN-APID controller.

Table 6. Summary of results under different activation functions of FSN-APID controller.

Test	KPI		State	Activation Function		
	Symbol	Unit		Hard Tanh	Tanh	Soft Sign
A	RMSE	m.	x	0.037	0.036	0.038
		m.	y	0.005	0.004	0.006
		rad.	θ	0.006	0.006	0.007
	MAD	m.	x	0.603	0.225	0.377
		m.	y	0.401	0.289	0.480
		rad.	θ	0.043	0.048	0.039
	t_{set}	s	x	6.4	6.6	6.6
		s	y	5.3	5.4	5.4
		s	θ	11.3	11.3	10.2

6.3. Robustness on Different Terrain Types

This test is performed to analyze the proposed FSN-APID controller's trajectory-tracking robustness on different terrain types. For this analysis, experiment A is performed separately on three different types of terrains; namely, concrete pavements to simulate urban environments, grassy fields to simulate uneven grassy terrain, and loose soil to simulate off-road, sandy, and dirty conditions. The state variations, exhibited by FSN-APID controller under the influence of each activation function, are displayed in Figure 32. The error profiles corresponding to each state variable are shown in Figure 33. The rose-curve trajectory tracking profiles with respect to each terrain type are shown in Figure 34. The experimental findings are summarized in Table 7. The concrete pavement yields the best trajectory-tracking behavior with minimal oscillations. This behavior is credited to the high friction surface of the concrete pavement, which improves traction and enables smoother (and more precise) trajectory tracking of the robot. The grassy field yields a mediocre trajectory tracking behavior. The loss of accuracy is caused by the variable friction and high resistance of the uneven (and possibly wet) field, which affects the tracking control response. Finally, the loose soil delivers the poorest tracking accuracy owing to its loose texture, which causes the robot's wheels to slip, stick, and sink. This phenomenon affects the sensor readings and degrades the tracking behavior.

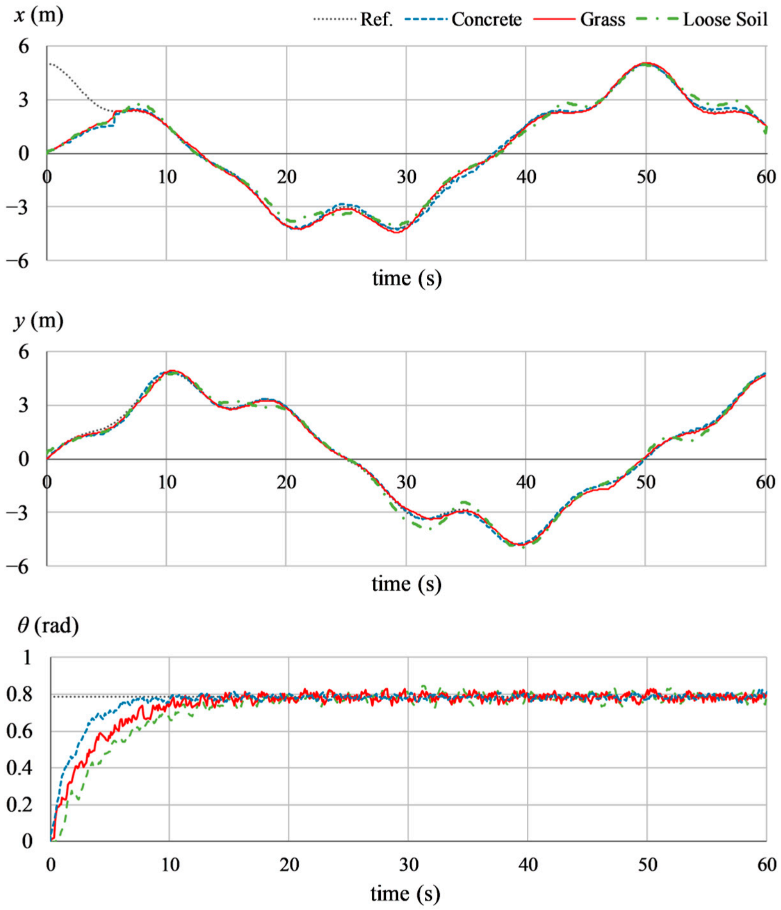

Figure 32. State variations under different terrain types.

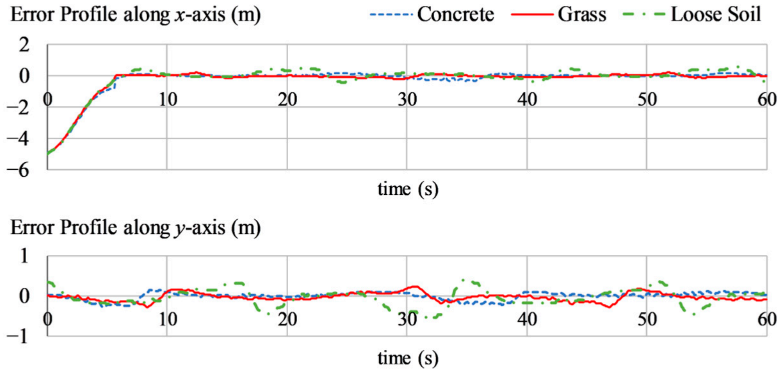

Figure 33. Error profile of each state variable under different terrain types.

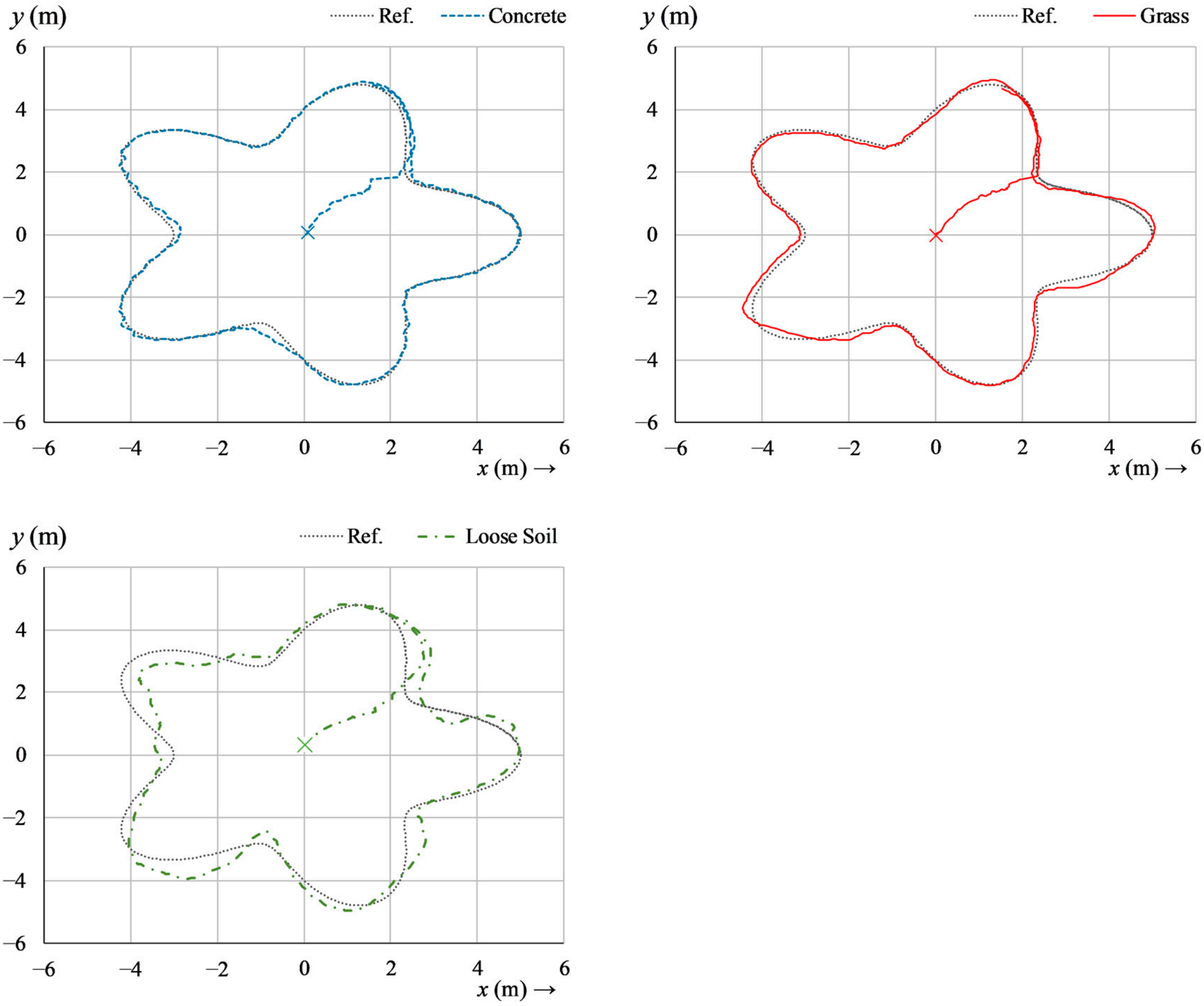

Figure 34. Rose-curve tracking profiles under different terrain types.

Table 7. Summary of results under different terrain types.

Test	**KPI**		**State**	**Terrain Type**		
	Symbol	**Unit**		**Concrete**	**Grass**	**Loose Dirt**
A	RMSE	m.	x	0.035	0.036	0.041
		m.	y	0.004	0.004	0.009
		rad.	θ	0.005	0.006	0.006
	MAD	m.	x	0.214	0.225	0.575
		m.	y	0.232	0.289	0.609
		rad.	θ	0.039	0.048	0.056
	t_{set}	s	x	6.4	6.6	6.9
		s	y	5.1	5.4	5.6
		s	θ	8.2	11.3	13.2

7. Conclusions

This article presents the constitution of an intelligent-adaptive steering control system for a differentially driven lawn-mowing robot, which effectively manages the environmental uncertainties, nonholonomic dynamics, and the system's intrinsic nonlinearities. The baseline SN-APID control procedure manifested improved convergence learning capabilities by leveraging the EKF for adaptive training, which yielded notable enhancement in trajectory tracking accuracy. The augmentation of the baseline controller with the proposed fuzzy-immune regulator further supplemented the controller's robustness, allowing it to emulate the self-tuning characteristics of biological immune responses.

The proposed hybrid approach offers a pragmatic solution by integrating the strengths of two different innovative control techniques. The proposed control scheme ensures that the robot maintains precise and stable trajectory tracking, even in challenging conditions like uneven terrain and varying surface conditions. These propositions are validated by conducting customized real-time experiments. The experimental findings justify that the proposed FSN-APID control scheme facilitates good functioning by effectively rejecting the impact of bounded external disturbances, guaranteeing dependable performance in unpredictable operating environments. As per the experimental results, the neural PID controller enhances tracking accuracy and disturbance rejection due to its inherent adaptive learning ability. However, its resilience deteriorates in extremely dynamic environment settings. By fusing the advantages of neural learning with the flexibility of the fuzzy-immune mechanism, the proposed FSN-APID controller provides the best overall performance. This capability guarantees quick convergence, excellent disturbance rejection, and real-time adaptation, which makes the proposed controller better suited for highly dynamic environments. To further improve the flexibility and resilience in dynamic environments, future research may investigate the extended application of this framework to other mobile robot configurations and explore other soft computing techniques such as a multilayer neural network instead of a single-neuron model, reinforcement learning, and fuzzy Q-learning. Finally, the implications of integrating a well-postulated SINN with a fractional PID control method, to adaptively modify its gains, can be investigated in future phases of this research work.

Author Contributions: Conceptualization, O.S.; methodology, O.S.; software, O.S. and A.H.; validation, J.I.; formal analysis, O.S.; investigation, O.S. and A.H.; resources, J.I.; data curation, A.H.; writing—original draft preparation, O.S.; writing—review and editing, J.I.; visualization, A.H.; supervision, J.I.; project administration, J.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The authors declare that the data analyzed during this study to support the findings will be made available on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest in preparing this article.

References

1. Khan, R.; Malik, F.M.; Raza, A.; Mazhar, N. Comprehensive study of skid-steer wheeled mobile robots: Development and challenges. *Ind. Robot.* **2021**, *48*, 142–156. [CrossRef]
2. Zohaib, M.; Pasha, S.M.; Javaid, N.; Salaam, A.; Iqbal, J. An improved algorithm for collision avoidance in environments having U and H shaped obstacles. *Stud. Inform. Control* **2014**, *23*, 97–106. [CrossRef]
3. Martins, N.A.; Bertol, D.W. *Wheeled Mobile Robot Control: Theory, Simulation, and Experimentation*; Springer Nature: Berlin/Heidelberg, Germany, 2021; Volume 380.
4. Raj, R.; Kos, A. A comprehensive study of mobile robot: History, developments, applications, and future research perspectives. *Appl. Sci.* **2022**, *12*, 6951. [CrossRef]
5. Song, L.; Huang, J.; Liang, Q.; Nie, L.; Liang, X.; Zhu, J. Trajectory tracking strategy for sliding mode control with double closed-loop for lawn mowing robot based on ESO. *IEEE Access* **2022**, *11*, 1867–1882. [CrossRef]
6. Shafaei, S.M.; Mousazadeh, H. Experimental comparison of locomotion system performance of ground mobile robots in agricultural drawbar works. *Smart Agric. Technol.* **2023**, *3*, 100131. [CrossRef]
7. Liao, J.C.; Chen, S.H.; Zhuang, Z.Y.; Wu, B.W.; Chen, Y.J. Designing and manufacturing of automatic robotic lawn mower. *Processes* **2021**, *9*, 358. [CrossRef]
8. Daniyan, I.; Balogun, V.; Adeodu, A.; Oladapo, B.; Peter, J.K.; Mpofu, K. Development and performance evaluation of a robot for lawn mowing. *Proced. Manuf.* **2020**, *49*, 42–48. [CrossRef]
9. Samson, C.; Morin, P.; Lenain, R. *Modeling and Control of Wheeled Mobile Robots*; Springer: Cham, Switzerland, 2016.
10. De Luca, A.; Oriolo, G.; Vendittelli, M. *Control of Wheeled Mobile Robots: An Experimental Overview*; Springer: Berlin/Heidelberg, Germany, 2001.

11. Tagliavini, L.; Colucci, G.; Botta, A.; Cavallone, P.; Baglieri, L.; Quaglia, G. Wheeled mobile robots: State of the art overview and kinematic comparison among three omnidirectional locomotion strategies. *J. Intell. Robot. Syst.* **2022**, *106*, 57. [CrossRef]
12. Tran, A.M.D.; Vu, T.V. A study on general state model of differential drive wheeled mobile robots. *J. Adv. Eng. Comput.* **2023**, *7*, 174–186. [CrossRef]
13. Felix-Rendon, J.; Bello-Robles, J.C.; Fuentes-Aguilar, R.Q. Control of differential-drive mobile robots for soft object deformation. *ISA Trans.* **2021**, *117*, 221–233. [CrossRef]
14. Hassan, N.; Saleem, A. Analysis of trajectory tracking control algorithms for wheeled mobile robots. In Proceedings of the 2021 IEEE Industrial Electronics and Applications Conference (IEACon), Penang, Malaysia, 22–23 November 2021.
15. Martins, O.; Adekunle, A.; Adejuyigbe, S.; Adeyemi, O.; Arowolo, M. Wheeled Mobile Robot Path Planning and Path Tracking Controller Algorithms: A Review. *J. Eng. Sci. Technol. Rev.* **2020**, *13*, 152–164. [CrossRef]
16. Khan, H.; Khatoon, S.; Gaur, P.; Abbas, M.; Saleel, C.A.; Khan, S.A. Speed control of wheeled mobile robot by nature-inspired social spider algorithm-based PID controller. *Processes* **2023**, *11*, 1202. [CrossRef]
17. Wang, J.; Liu, Z.; Chen, H.; Zhang, Y.; Zhang, D.; Peng, C. Trajectory Tracking Control of a Skid-Steer Mobile Robot Based on Nonlinear Model Predictive Control with a Hydraulic Motor Velocity Mapping. *Appl. Sci.* **2024**, *14*, 122. [CrossRef]
18. Zhang, L.; Liu, L.; Zhang, S. Design, Implementation, and Validation of Robust Fractional-Order PD Controller for Wheeled Mobile Robot Trajectory Tracking. *Complexity* **2020**, *2020*, 9523549. [CrossRef]
19. Xu, L.; Du, J.; Song, B.; Cao, M. A combined backstepping and fractional-order PID controller to trajectory tracking of mobile robots. *Syst. Sci. Control Eng.* **2022**, *10*, 134–141. [CrossRef]
20. Thai, N.H.; Ly, T.T.K.; Thien, H.; Dzung, L.Q. Trajectory tracking control for differential-drive mobile robot by a variable parameter PID controller. *Int. J. Mech. Eng. Robot. Res.* **2022**, *11*, 614–621. [CrossRef]
21. Zhai, J.; Song, Z. Adaptive sliding mode trajectory tracking control for wheeled mobile robots. *Int. J. Control* **2018**, *92*, 2255–2262. [CrossRef]
22. Ayyıldız, M.; Tilki, U. Adaptive sliding mode based fault tolerant control of wheeled mobile robots. *Automatika* **2023**, *64*, 467–483. [CrossRef]
23. Mechali, O.; Xu, L.; Xie, X.; Iqbal, J. Fixed-time nonlinear homogeneous sliding mode approach for robust tracking control of multirotor aircraft: Experimental validation. *J. Franklin Inst.* **2022**, *359*, 1971–2029. [CrossRef]
24. Wei, J.; Zhu, B. Model predictive control for trajectory-tracking and formation of wheeled mobile robots. *Neural Comput. Appl.* **2022**, *34*, 16351–16365. [CrossRef]
25. Li, P.; Yang, H.; Wang, S. Model predictive tracking control with disturbance compensation for wheeled mobile robots in an environment with obstacles. *J. Franklin Inst.* **2023**, *360*, 6669–6692. [CrossRef]
26. Wang, J.; Fader, M.T.; Marshall, J.A. Learning-based model predictive control for improved mobile robot path following using Gaussian processes and feedback linearization. *J. Field Robot.* **2023**, *40*, 1014–1033. [CrossRef]
27. Szuster, M.; Hendzel, Z. *Intelligent Optimal Adaptive Control for Mechatronic Systems*; Springer: Berlin/Heidelberg, Germany, 2018.
28. Khan, H.; Khatoon, S.; Gaur, P.; Khan, S.A. Speed control comparison of wheeled mobile robot by ANFIS, Fuzzy and PID controllers. *Int. J. Informat Technol.* **2022**, *14*, 1893–1899. [CrossRef]
29. Štefek, A.; Pham, V.T.; Krivanek, V.; Pham, K.L. Optimization of fuzzy logic controller used for a differential drive wheeled mobile robot. *Appl. Sci.* **2021**, *11*, 6023. [CrossRef]
30. Mondal, S.; Ray, R.; Reddy, S.; Nandy, S. Intelligent controller for nonholonomic wheeled mobile robot: A fuzzy path following combination. *Math. Comput. Simul.* **2022**, *193*, 533–555. [CrossRef]
31. Cao, G.; Zhao, X.; Ye, C.; Yu, S.; Li, B.; Jiang, C. Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot. *J. Mech. Sci. Technol.* **2022**, *36*, 2019–2029. [CrossRef]
32. Bozek, P.; Karavaev, Y.L.; Ardentov, A.A.; Yefremov, K.S. Neural network control of a wheeled mobile robot based on optimal trajectories. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420916077. [CrossRef]
33. Hassan, N.; Saleem, A. Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots. *IEEE Access* **2022**, *10*, 13582–13597. [CrossRef]
34. Chen, Z.; Liu, Y.; He, W.; Qiao, H.; Ji, H. Adaptive-neural-network-based trajectory tracking control for a nonholonomic wheeled mobile robot with velocity constraints. *IEEE Trans. Ind. Electron.* **2020**, *68*, 5057–5067. [CrossRef]
35. Malekabadi, M.; Haghparast, M.; Nasiri, F. Air Condition's PID Controller Fine-Tuning Using Artificial Neural Networks and Genetic Algorithms. *Computers* **2018**, *7*, 32. [CrossRef]
36. Hernandez-Barragan, J.; Rios, J.D.; Alanis, A.Y.; Lopez-Franco, C.; Gomez-Avila, J.; Arana-Daniel, N. Adaptive single neuron anti-windup pid controller based on the extended kalman filter algorithm. *Electronics* **2020**, *9*, 636. [CrossRef]
37. Tang, W.; Wang, L.; Gu, J.; Gu, Y. Single neural adaptive PID control for small UAV micro-turbojet engine. *Sensors* **2020**, *20*, 345. [CrossRef] [PubMed]
38. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Real-time learning capability of neural networks. *IEEE Trans. Neural Netw.* **2024**, *17*, 863–878. [CrossRef] [PubMed]
39. Tabbussum, R.; Dar, A.Q. Comparative analysis of neural network training algorithms for the flood forecast modelling of an alluvial Himalayan river. *J. Flood Risk Manag.* **2020**, *13*, e12656. [CrossRef]
40. Saleem, O.; Alsuwian, T.; Amin, A.A.; Ali, S.; Alqarni, Z.A. Stabilization control of rotary inverted pendulum using a novel EKF-based fuzzy adaptive sliding-mode controller: Design and experimental validation. *Automatika* **2024**, *65*, 538–558. [CrossRef]

41. Li, L.; Wang, T.; Xia, Y.; Zhou, N. Trajectory tracking control for wheeled mobile robots based on nonlinear disturbance observer with extended Kalman filter. *J. Franklin Inst.* **2020**, *357*, 8491–8507. [CrossRef]
42. Taghavifar, H.; Hu, C.; Qin, Y.; Wei, C. EKF-neural network observer based type-2 fuzzy control of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4788–4800. [CrossRef]
43. Cho, S.; Shrestha, B.; Jang, W.; Seo, C. Trajectory tracking optimization of mobile robot using artificial immune system. *Multimed. Tools Appl.* **2019**, *78*, 3203–3220. [CrossRef]
44. Goto, W.J.N.; Wildgrube Bertol, D.; Martins, N.A. Adaptive immune fuzzy quasi-sliding mode control for leader–follower formation of wheeled mobile robots under uncertainties and disturbances with obstacle avoidance. *Eng. Comput.* **2024**, *41*, 1697–1739. [CrossRef]
45. Huang, H.C. Intelligent motion control for four-wheeled holonomic mobile robots using FPGA-based artificial immune system algorithm. *Adv. Mech. Eng.* **2013**, *5*, 589510. [CrossRef]
46. Khai, T.Q.; Ryoo, Y.J.; Gill, W.R.; Im, D.Y. Design of kinematic controller based on parameter tuning by fuzzy inference system for trajectory tracking of differential-drive mobile robot. *Int. J. Fuzzy Syst.* **2020**, *22*, 1972–1978. [CrossRef]
47. Bhatti, O.S. Adaptive fuzzy-pd tracking controller for optimal visual-servoing of wheeled mobile robots. *Control Eng. Appl. Inform.* **2017**, *19*, 58–68.
48. Suarez-Rivera, G.; Muñoz-Ceballos, N.D.; Vásquez-Carvajal, H.M. Development of an adaptive trajectory tracking control of wheeled mobile robot. *Revista Facultad Ingeniería* **2021**, *30*, 1–15. [CrossRef]
49. Toroslu, I.; Doğan, M. Effective sensor fusion of a mobile robot for SLAM implementation. In Proceedings of the 2018 4th International Conference on Control, Automation and Robotics (ICCAR), Auckland, New Zealand, 20–23 April 2018; pp. 76–81.
50. Murua, A.M.; Bistue, G.; Rubio, A.; Gonzalez, J. A direct yaw-moment control logic for an electric 2wd formula sae using an error-cube proportional derivative controller. *SAE Int. J. Connect. Autom. Veh.* **2020**, *3*, 139–148. [CrossRef]
51. Alagoz, B.B.; Ates, A.; Yeroglu, C.; Senol, B. An experimental investigation for error-cube PID control. *Trans. Inst. Meas. Control* **2015**, *37*, 652–660. [CrossRef]
52. Su, D.; Yao, W.; Yu, F.; Liu, Y.; Zheng, Z.; Wang, Y.; Xu, T.; Chen, C. Single-neuron PID UAV variable fertilizer application control system based on a weighted coefficient learning correction. *Agriculture* **2022**, *12*, 1019. [CrossRef]
53. Saleem, O.; Omer, U. EKF-based self-regulation of an adaptive nonlinear PI speed controller for a DC motor. *Turk. J. Electr. Eng. Comput. Sci.* **2017**, *25*, 4131–4141. [CrossRef]
54. Saleem, O.; Iqbal, J. Fuzzy-immune-regulated adaptive degree-of-stability LQR for a self-balancing robotic mechanism: Design and HIL realization. *IEEE Robot. Automat. Lett.* **2023**, *8*, 4577–4584. [CrossRef]
55. Al Mamun, M.A.; Nasir, M.T.; Khayyat, A. Embedded system for motion control of an omnidirectional mobile robot. *IEEE Access* **2018**, *6*, 6722–6739. [CrossRef]
56. Lupu, D.; Necoara, I. Exact representation and efficient approximations of linear model predictive control laws via HardTanh type deep neural networks. *Syst. Control Lett.* **2024**, *186*, 105742. [CrossRef]
57. Wang, Y.; Li, Y.; Song, Y.; Rong, X. The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition. *Appl. Sci.* **2020**, *10*, 1897. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.