processes

MDPI

Article

Predefined-Time Fuzzy Neural Network Control for Omnidirectional Mobile Robot

Peng Qin, Tao Zhao, Nian Liu *, Zhen Mei and Wen Yan

College of Electrical Engineering, Sichuan University, Chengdu 610065, China
* Correspondence: liu-nian@scu.edu.cn

Abstract: In this paper, a fuzzy neural network based predefined-time trajectory tracking control method is proposed for the tracking problem of omnidirectional mobile robots (FM-OMR) with uncertainties. Considering the requirement of tracking error convergence time, a position tracking controller based on predefined-time stability is proposed. Compared with the traditional position tracking control method, the minimum upper bound of the convergence time can be explicitly set. In order to obtain more accurate angular velocity tracking, the inner loop controller combines Type 1 fuzzy neural network (T1FNN) to estimate the uncertainty. In addition, considering the problem of feedback channel noise, a Kalman filter combining velocity and position information is proposed. Finally, the simulation results verify the effectiveness of this method.

Keywords: trajectory tracking; predefined-time stability; fuzzy neural network; omnidirectional mobile robot

check for updates

Citation: Qin, P.; Zhao, T.; Liu, N.; Mei, Z.; Yan, W. Predefined-Time Fuzzy Neural Network Control for Omnidirectional Mobile Robot. *Processes* **2023**, *11*, 23. https://doi.org/10.3390/pr11010023

Academic Editor: Jie Zhang

Received: 24 November 2022
Revised: 14 December 2022
Accepted: 19 December 2022
Published: 22 December 2022

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the development of technology, there is an increasing demand for autonomy and intelligence, and mobile robots can play a key role in challenging problems in numerous fields. For example, some ultraviolet-C disinfection robots have recently been developed to deal with the new coronavirus [1]. Moreover, many households need mobile robots to perform cleaning tasks [2]. In addition, mobile robots are widely used for exploration in hazardous environments such as space [3], mine clearance [4], search and rescue [5], and military operations. Other application areas of mobile robots are warehouse and industrial logistics and heavy-duty transportation in industrial environments [6], precision agriculture [7,8].

Omnidirectional wheeled mobile robots have been extensively studied due to their greater mobility compared to conventional mobile robots, and hence their ability to move easily in constrained spaces [9]. Mecanum wheeled mobile robots are a type of omnidirectional mobile robot, which consists of a number of passive rollers that are mounted at 45 degrees around a solid axis. Mecanum wheeled mobile machines can move without changing the initial position orientation and can be well adapted to narrow, unstructured environments [10].

In order to achieve accurate autonomous motion of wheeled robots, the trajectory tracking control method is a key part. In practical application scenarios, there are also external unknown disturbances and internal model uncertainties, which pose challenging problems for trajectory tracking control. To solve this problem, many scholars have proposed different control methods: pid control [11], fuzzy control [12], sliding mode control [13] and other methods have all been studied with certain results. In [14], trajectory tracking control is implemented using model predictive control algorithm with control and system constraints. In [15], a sliding mode control method based on an expansive state observer was used to achieve trajectory tracking control of a four-wheel tracked wheeled mobile platform considering perturbations and model uncertainty. However, if the uncertainty of model is too large, the appeal method is difficult to deal with. Neural network

has been widely used in network attack and defense [16–18], classification, prediction, and also in the control field. In [19], a data-driven algorithm based on a TS fuzzy quadratic numerical neural network combined with a controlled autoregressive integral moving average model is used to design a generalized predictive controller for trajectory tracking control of a four-wheeled omnidirectional mobile robot. In [20], for wheel legged robots tracking, FNN is applied to approximate the unknown disturbance in the dynamic model. In [21], a control algorithm based on a global adaptive neural network is presented for disturbed pure feedback nonlinear systems. In particular, the approximation area of the neural network is a subset of the entire state space and can be switched between RBF and robust control through a switching function.

Although all existing methods can achieve certain control results, the sliding mode algorithm can lead to jitter in the output of the control system. Moreover, the applications nowadays have high requirements on the real-time of the control effect. To solve these problems, some scholars have studied scheduled time stabilization. The trajectory planning and control of a two-armed free-floating space robot based on predefined time stabilization is implemented in [22]. The problem of predefined-time attitude stabilization of a receptive vehicle with uncertain inertia and unknown external disturbances is implemented in [23]. The predefined time control is used in [24] for DC microgrid regulation.

However, the problem of predefined time for mobile robots is still less studied. In this paper, this approach is introduced to improve the control of mobile robots to achieve high precision and high real-time control. Considering the disturbance of external conditions and the variation of internal models, this paper considers the use of fuzzy systems to achieve fuzzy compensation. The universal approximation property of the fuzzy system and its ability to handle uncertainty are used to deal with disturbances. Fuzzy logic systems have been successfully used in a variety of areas [25–30]. A fuzzy non-singular terminal sliding mode is used in [31] to achieve trajectory tracking control of an automatic guided vehicle with Mwcanum wheels. The problem of scheduled time suppression for TS fuzzy systems is studied in [32]. Fuzzy fault-tolerant control of a Mecanum omnidirectional mobile robot is implemented in [33].

Although the control method can handle some uncertainties, the feedback channel can be subject to many disturbances in practice and affect the control effect of the system. In this paper, a Kalman filter is used for the filtering process. In [34], a sensor fusion method based on the extended Kalman filter is used to achieve path tracking control of a complete mobile robot with four mechanical wheels. Indoor visual navigation of a mobile robot is implemented in [35] using a multi-model, multi-frequency based Kalman filter.

Based on the above discussion, it can be seen that the trajectory tracking of FM-OMR faces the problems of uncertainty, feedback channel noise and control time. Inspired by the shortcomings and advantages of the above research, the motivation of this paper is to propose a trajectory tracking control method to deal with the above three problems. The main contributions and advantages of this method are as follows:

(1) Considering the requirement of control time, the predefined-time stability is introduced into the design of trajectory tracking controller for FM-OMR. Compared with the traditional feedforward proportional controller, the parameters of the controller can not only explicitly specify the minimum upper bound of the convergence time, but also adjust the convergence process of the error within the upper bound, which improves the flexibility of the tracking control.

(2) There is still a gap in the research of the combination of predefined-time theory and fuzzy neural network, and there is no exploration of relevant methods in the field of FM-OMR trajectory tracking control. The predefined-time controller combined with fuzzy neural network is proposed. The approximation of fuzzy neural network is used to reduce the influence of uncertainty and improve the tracking performance of angular velocity. The method verifies the feasibility of the combination of the two and its application in the trajectory tracking control of FM-OMR.

(3) Considering the influence of feedback noise on the system, a Kalman filter is designed to improve the quality of feedback signal.

The structure of this paper is as follows. Preliminaries are introduced in Section 2. The design of the control scheme are shown in Section 3. Section 4 shows the stability analysis. Section 5 reveals the simulation results Finally, the conclusion is presented in Section 6.

2. Preliminaries

2.1. Model

The omnidirectional mobile robot in this paper is shown in Figure 1. The four Mecanum wheels of the chassis are arranged symmetrically, and the direction of the diagonal wheels is the same. Mecanum wheel was proposed by engineer Bengt Ilon in 1973. Mecanum wheel is different from other types of wheels. It consists of wheel hub and roller, as shown in the Figure 2. The included angle between wheel axle and roller axle is 45°. When the wheel rotates, it will drive the roller to rotate, generating a force of lateral sliding, so that the wheel can move on two degrees of freedom. By reasonably controlling the rotation of the four wheels, the robot can move in all directions. According to reference [36] and Figure 1 its kinematic model is (1).

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & 1 & -(l_x + l_y) \\ -1 & 1 & l_x + l_y \\ 1 & 1 & l_x + l_y \\ -1 & 1 & -(l_x + l_y) \end{bmatrix} \begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} \tag{1}$$

where ω_i is the angular velocity, R is the radius of wheel, l_x is the track width, l_y is the wheelbase, (V_X, V_Y, V_Z) is the velocity of the robot. The inverse solution is (2).

$$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} = \frac{R}{4} \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ \frac{-1}{l_x+l_y} & \frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} & \frac{-1}{l_x+l_y} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{2}$$

The kinematic model mainly reflects the relationship between the vehicle velocity and the angular velocity of each wheel. The output of the outer loop controller and the input of the inner loop controller depend on their conversion.

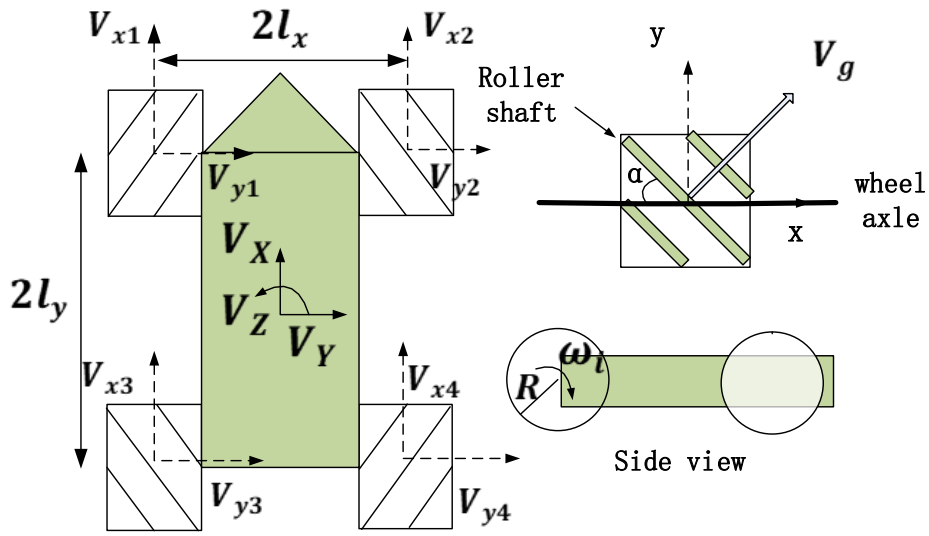


Figure 1. Structure diagram of FM-OMR.



Figure 2. Mecanum wheel.

According to [33], the ideal dynamic model is (3). Table 1 gives the definitions of variables in (3).

$$\begin{cases} M\dot{\omega} + D_\omega \omega = \tau \\ M = \begin{bmatrix} \Upsilon + \Xi + I_w & -\Upsilon & -\Upsilon + \Xi & \Upsilon \\ -\Upsilon & \Upsilon + \Xi + I_w & \Upsilon & -\Upsilon + \Xi \\ -\Upsilon + \Xi & \Upsilon & \Upsilon + \Xi + I_w & -\Upsilon \\ \Upsilon & -\Upsilon + \Xi & -\Upsilon & \Upsilon + \Xi + I_w \end{bmatrix} \\ D_\omega = diag(\mu_1, \mu_2, \mu_3, \mu_4), \mu_i \geq 0 \\ \tau = [\tau_1, \tau_2, \tau_3, \tau_4]^T \\ \Upsilon = \frac{I_\omega R^2}{16(l_x + l_y)^2} \ , \Xi = \frac{m_r R^2}{8} \end{cases} \tag{3}$$

Table 1. Definition of parameters.

Parameter	Definition
M	Inertial matrix
D_w	Viscous friction coefficient matrix
$\dot{\omega}$	The angular acceleration
μ_i	The coefficient of viscous friction
τ	The input torque
I_ω	The rotational inertia of the wheel
m_r	The quality of robot

Considering that there may be uncertainty in the system, a function $F(\omega)$ describing the uncertainty is introduced into the model, as shown in (4).

$$M\dot{\omega} + D_\omega \omega + F(\omega) = \tau \tag{4}$$

The dynamic model describes the relationship between control torque τ and angular velocity ω, and the inner loop controller will be designed based on the dynamic model.

2.2. Predefined Time Stability Theory

According to [22], for the following system (5).

$$\dot{x}(t) = f(x(t)), x(0) = x_0 \tag{5}$$

When the system meets the global fixed-time stable and the settling time $T(x_0)$ has the minimum upper bound T_{max}, the system is predefined-time stable.

When the system meets the following form, it is proved that the predefined-time stability can be obtained.

$$\dot{x} = -\frac{g(\infty) - g(0)}{T_c}\left(\frac{\partial g(|x|^m)}{\partial x}\right)^{-1} \tag{6}$$

where $0 < m < 1$, $g(x)$ is a settable function and satisfies the following conditions.

(a) $\lim_{\triangle \to 0} \frac{g(|x|^m + \triangle) - g(|x|^m)}{\triangle} > 0.$

(b) $g(x)$ is bounded.

(c) $\lim_{x \to 0}\left(\frac{\partial g(|x|^m)}{\partial x}\right)^{-1} = 0.$

3. Control Method Design

3.1. Overall Structure of Method

According to the robot model and the predefined time stability theory, a dual loop trajectory tracking controller is designed. The controller is mainly composed of three parts, including a predefined time position tracking controller in the outer loop, a fuzzy neural network compensation controller in the inner loop, and a filter in the feedback channel. The overall structure is shown in Figure 3. The origin of the control signal and the control process are as follows. First, the position and velocity of the robot are measured by sensors and filtered position information is obtained by Kalman filter. Secondly, the filtered position is compared with the reference trajectory to obtain the position error. Thirdly, the desired velocity is obtained according to the position error and the outer loop controller, and is converted into the desired angular velocity of the four wheels through the kinematics model. Finally, according to the expected angular velocity of the four wheels, the inner loop controller combines the output of T1FNN to obtain the control torque of the four wheels. Under the control torque, the wheels achieve angular velocity tracking and thus position tracking.

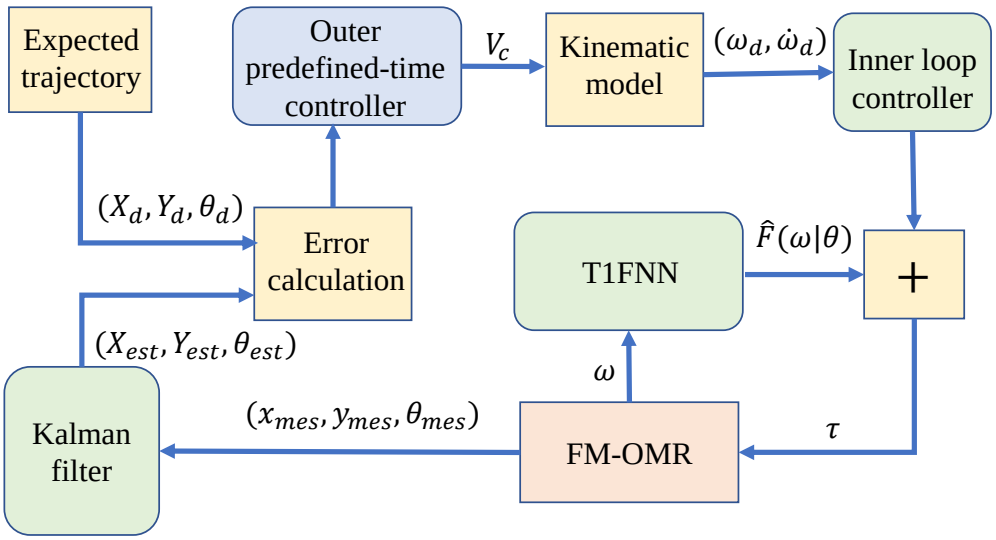


Figure 3. Control structure diagram.

3.2. Outer Loop Controller

Because the role of the outer loop controller is to achieve position tracking, the controller is designed according to the position error model and the predefined time stability theory. The tracking error of FM-OMR is shown in the Figure 4.

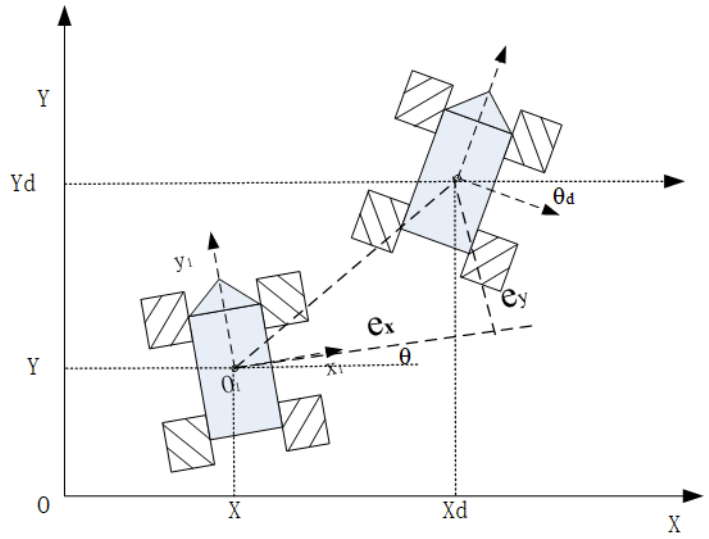


Figure 4. Tracking error.

Because FM-OMR has the ability to move in all directions at the same time, tracking error can be considered in the coordinate system of the robot body. From Figure 4, the tracking error based on the body coordinate system can be obtained as shown in (7).

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_d - X \\ Y_d - Y \\ \theta_d - \theta \end{bmatrix} \tag{7}$$

The derivative of (7) is (8).

$$\begin{cases} \dot{e}_x = \omega e_y - V_x + cos\dot{X}_d + sin\dot{Y}_d \\ \dot{e}_y = -\omega e_x - V_y + cos\dot{Y}_d - sin\dot{X}_d \\ \dot{e}_\theta = \omega_d - \omega \end{cases} \tag{8}$$

According to the predefined-time stability theory (6), it is necessary to build a controller to make the tracking error meet the following form.

$$\begin{cases} \dot{e}_x = -\dfrac{g(\infty) - g(0)}{T_x} \left(\dfrac{\partial g(|e_x|^{m_x})}{\partial e_x} \right)^{-1} \\ \dot{e}_y = -\dfrac{g_1(\infty) - g_1(0)}{T_y} \left(\dfrac{\partial g_1(|e_y|^{m_y})}{\partial e_y} \right)^{-1} \\ \dot{e}_\theta = -\dfrac{g_2(\infty) - g_2(0)}{T_\theta} \left(\dfrac{\partial g_2(|e_\theta|^{m_\theta})}{\partial e_\theta} \right)^{-1} \end{cases} \tag{9}$$

where T_x, T_y and T_θ respectively correspond to the predefined convergence time of x direction, y direction and yaw angle error. m_x, m_y, m_θ are the adjustable parameters of their corresponding function g.

According to the condition requirements of function g, the g function set in this paper is (10).

$$\begin{cases} g({e_x}^m) = 1 - exp\left[-\dfrac{1}{2} \left(\dfrac{{e_x}^m}{\sigma_x} \right)^2 \right] \\ g({e_y}^m) = 1 - exp\left[-\dfrac{1}{2} \left(\dfrac{{e_y}^m}{\sigma_y} \right)^2 \right] \\ g({e_\theta}^m) = 1 - exp\left[-\dfrac{1}{2} \left(\dfrac{{e_\theta}^m}{\sigma_\theta} \right)^2 \right] \end{cases} \tag{10}$$

where $\sigma_x, \sigma_y, \sigma_\theta$ are adjustable parameters.

Since the FM-OMR has the ability to move in all directions and its yaw angle can remain unchanged, the angular velocity can be set to 0. According to (8)–(10), the outer loop controller can be obtained as (11).

$$\begin{cases} V_x = \frac{1}{T_x}\left[exp\left[-\frac{1}{2}\left(\frac{{e_x}^{m_x}}{\sigma_x}\right)^2\right]\frac{m_x}{\sigma_x^2}{e_x}^{2m_x-1}\right]^{-1} + \dot{X}_d cos\theta + \dot{Y}_d sin\theta \\ V_y = \frac{1}{T_y}\left[exp\left[-\frac{1}{2}\left(\frac{{e_y}^{m_y}}{\sigma_y}\right)^2\right]\frac{m_y}{\sigma_y^2}{e_y}^{2m_y-1}\right]^{-1} + \dot{Y}_d cos\theta - \dot{X}_d sin\theta \\ \omega = \frac{1}{T_\theta}\left[exp\left[-\frac{1}{2}\left(\frac{{e_\theta}^{m_\theta}}{\sigma_\theta}\right)^2\right]\frac{m_\theta}{\sigma_\theta^2}{e_\theta}^{2m_\theta-1}\right]^{-1} \end{cases} \tag{11}$$

3.3. Inner Loop Controller

The reference speed can be obtained based on the above outer loop controller, and the function of the inner loop controller is to track the expected velocity. According to (4), the angular velocity error model can be obtained as (12), where ω_d is expected angular velocity.

$$\dot{\omega}_d - \dot{\omega} = \dot{\omega}_d - M^{-1}(\tau - F(\omega) - D_\omega \omega) \tag{12}$$

According to the predefined-time theory, (13) can be obtained.

$$\dot{\omega}_d - \dot{\omega} = -\frac{g_\omega(\infty) - g_\omega(0)}{T_\omega}\left(\frac{\partial g_\omega(|e_\omega|^{m_\omega})}{\partial e_\omega}\right)^{-1} \tag{13}$$

Using the same g function as (10), we can deduce that the controller is (14).

$$\tau = M\dot{\omega}_d + D_\omega \omega + F(\omega) + MG + k_d(\omega_d - \omega) \tag{14}$$

where

$$G = \begin{bmatrix} \frac{1}{T_{\omega_1}}\left[exp\left[-\frac{1}{2}\left(\frac{|e_{\omega_1}|^{m_{\omega_1}}}{\sigma_{\omega_1}}\right)^2\right]\frac{2m_{\omega_1}}{\sigma_{\omega_1}^2}|e_{\omega_1}|^{2m_{\omega_1}-2}e_{\omega_1}\right]^{-1} \\ \frac{1}{T_{\omega_2}}\left[exp\left[-\frac{1}{2}\left(\frac{|e_{\omega_2}|^{m_{\omega_2}}}{\sigma_{\omega_2}}\right)^2\right]\frac{2m_{\omega_2}}{\sigma_{\omega_2}^2}|e_{\omega_2}|^{2m_{\omega_2}-2}e_{\omega_2}\right]^{-1} \\ \frac{1}{T_{\omega_3}}\left[exp\left[-\frac{1}{2}\left(\frac{|e_{\omega_3}|^{m_{\omega_3}}}{\sigma_{\omega_3}}\right)^2\right]\frac{2m_{\omega_3}}{\sigma_{\omega_3}^2}|e_{\omega_3}|^{2m_{\omega_3}-2}e_{\omega_3}\right]^{-1} \\ \frac{1}{T_{\omega_4}}\left[exp\left[-\frac{1}{2}\left(\frac{|e_{\omega_4}|^{m_{\omega_4}}}{\sigma_{\omega_4}}\right)^2\right]\frac{2m_{\omega_4}}{\sigma_{\omega_4}^2}|e_{\omega_4}|^{2m_{\omega_4}-2}e_{\omega_4}\right]^{-1} \end{bmatrix} \tag{15}$$

The parameters in Equation (15) are similar to those in (9), $[e_{\omega_1}, e_{\omega_2}, e_{\omega_3}, e_{\omega_4}] = \omega_d - \omega$.

3.4. Design of T1FNN

Since the inner loop controller (14) contains uncertainty term $F(\omega)$, it is necessary to construct T1FNN to compensate. The structure of T1FNN is based on fuzzy rules, and the expression form of fuzzy rules is (16) [37].

$$\begin{gathered} \text{Rule } i \text{ :If } x_1 \text{ is } A_i^1, \ldots, \text{ and } x_n \text{ is } A_i^n \\ \text{Then } y \text{ is } \theta_i, i = 1, \cdots, K \end{gathered} \tag{16}$$

It includes input variables $x_j, j = 1 \ldots n$, fuzzy sets $A_i^j, i = 1 \ldots K, j = 1 \ldots n$ and adjustable parameters θ_i.

The form of fuzzy set is shown in (17).

$$\mu_j^i\left(x_j\right) = \exp\left\{\left(\frac{x_j - m_j^i}{\sigma_j^i}\right)^2\right\} \tag{17}$$

where m_j^i and σ_j^i are adjustable parameters that determine the shape of the fuzzy set.

The output of the network can be expressed as (18)

$$y = \frac{\sum_{k=1}^{N} f^k \theta_k}{\sum_{k=1}^{N} f^k} = \sum_{k=1}^{N} \theta_k \zeta^k = \boldsymbol{\theta}\boldsymbol{\zeta}(x) \tag{18}$$

The structure of T1FNN is shown in Figure 5.

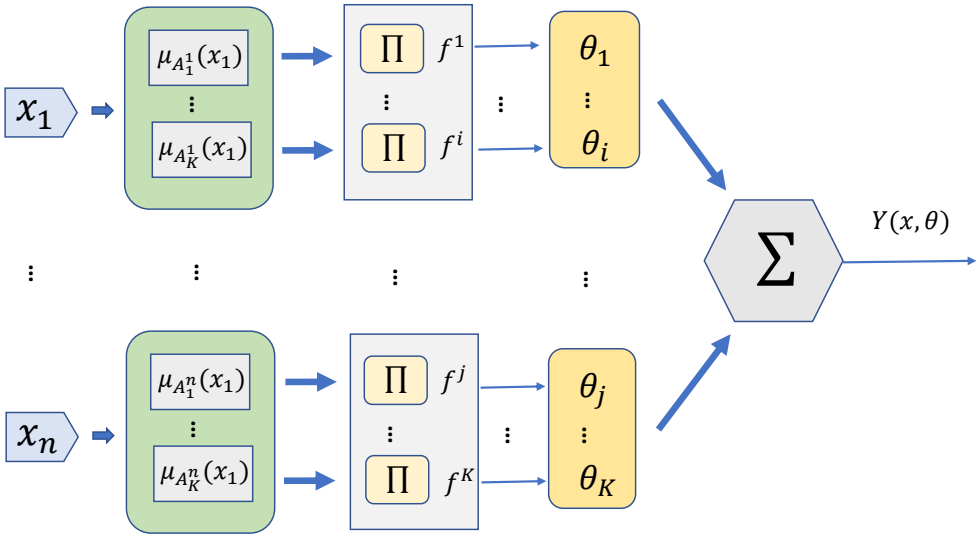


Figure 5. Structure of T1FNN.

Since the uncertainty term to be approximated is $F(\omega)$, a T1FNN with ω as input can be constructed. Compared with FNN as a controller, T1FNN in this paper is only used as an approximator to compensate the inner loop controller. Therefore, it is not necessary to train T1FNN in advance, but use the idea of adaptive control to update the network parameters in each control cycle. The adaptive law of rule parameters is (19).

$$\dot{\hat{\boldsymbol{\theta}}} = \frac{1}{\Gamma_r} s\boldsymbol{\zeta}(\omega) \tag{19}$$

To sum up, the final internal loop controller is (20)

$$\tau = M\dot{\omega}_d + D\omega + \hat{F}(\omega \mid \boldsymbol{\theta}) + MG + k_d(\omega_d - \omega) \tag{20}$$

3.5. Filter Design

In the actual system, the sensor has some noise, which will affect the performance of the control system. FM-OMR is a multi-sensor system, so real data can be estimated by fusing different sensor data. In this paper, based on the principle of the Kalman filter, the velocity information and the measured position information are fused to obtain the real position information.

Combined with the Kalman recursive formula, the following filter can be obtained. The predicted value is (21).

$$P_{pre}(k) = AP_{est}(k-1) + BV(k)T_S \tag{21}$$

where P_{pre} represents the predicted value, P_{est} represents the estimated value, V is velocity, B is the input matrix, A is the State matrix, and T_s is the sampling time.

The prior error covariance is (22).

$$H_{pre}(k) = AH_{est}(k-1)A^T + BQB^T \tag{22}$$

where H_{pre} represents the prior error covariance, H_{est} represents the estimated error covariance, and Q is covariance of noise.

The Kalman gain is (23).

$$Kar = \frac{H_{pre}(k)C}{CH_{pre}(k)C^T + R} \tag{23}$$

where C represents output matrix and R is process noise covariance.

The estimated value is (24).

$$P_{est}(k) = P_{pre}(k) + kar(P_{mes}(k) - CP_{pre}(k)) \tag{24}$$

where P_{mes} represents the measured value.

The estimated error covariance is (25).

$$H_{est}(k) = (I - karC)H_{pre}(k) \tag{25}$$

4. Stability Analysis

For outer loop control system, Consider candidate Lyapunov functions $V_1 = |e_x|$, $V_2 = |e_y|$,$V_3 = |e_\theta|$. Their derivatives are (26), (27) and (28), respectively.

$$\begin{aligned}
\dot{V}_1 &= \frac{e_x\dot{e}_x}{|e_x|} \\
&= -\frac{1}{T_x}\left(\frac{\partial g(|e_x|^{m_x})}{\partial e_x}\right)^{-1}\frac{e_x}{|e_x|} \\
&= -\frac{1}{T_x}\left[exp\left[-\frac{1}{2}\left(\frac{|e_x|^{m_x}}{\sigma_x}\right)^2\right]\frac{2m_x}{\sigma_x^2}|e_x|^{2m_x-1}\frac{e_x}{|e_x|}\right]^{-1}\frac{e_x}{|e_x|} \\
&= -\frac{1}{m_xT_x}\left[\frac{\partial g(V^{m_x})}{\partial V^{m_x}}\right]^{-1}V^{1-m_x}
\end{aligned} \tag{26}$$

$$\begin{aligned}
\dot{V}_2 &= \frac{e_y\dot{e}_y}{|e_y|} \\
&= -\frac{1}{T_y}\left(\frac{\partial g(|e_y|^{m_y})}{\partial e_y}\right)^{-1}\frac{e_y}{|e_y|} \\
&= -\frac{1}{T_y}\left[exp\left[-\frac{1}{2}\left(\frac{|e_y|^{m_y}}{\sigma_y}\right)^2\right]\frac{2m_y}{\sigma_y^2}|e_y|^{2m_y-1}\frac{e_y}{|e_y|}\right]^{-1}\frac{e_y}{|e_y|} \\
&= -\frac{1}{m_yT_y}\left[\frac{\partial g(V^{m_y})}{\partial V^{m_y}}\right]^{-1}V^{1-m_y}
\end{aligned} \tag{27}$$

$$\begin{aligned}\dot{V}_3 &= \frac{e_\theta \dot{e}_\theta}{|e_\theta|} \\ &= -\frac{1}{T_\theta}\left(\frac{\partial g(|e_\theta|^{m_\theta})}{\partial e_\theta}\right)^{-1}\frac{e_\theta}{|e_\theta|} \\ &= -\frac{1}{T_\theta}\left[exp\left[-\frac{1}{2}\left(\frac{|e_\theta|^{m_\theta}}{\sigma_\theta}\right)^2\right]\frac{2m_\theta}{\sigma_\theta^2}|e_\theta|^{2m_\theta-1}\frac{e_\theta}{|e_\theta|}\right]^{-1}\frac{e_\theta}{|e_\theta|} \\ &= -\frac{1}{m_\theta T_\theta}\left[\frac{\partial g(V^{m_\theta})}{\partial V^{m_\theta}}\right]^{-1}V^{1-m_\theta}\end{aligned} \tag{28}$$

According to [22], it can be determined that e_x, e_y, e_y are finite-time stable. Their convergence times are (29)–(31).

$$\begin{aligned}T_1 &= \int_0^{e_{x0}} \frac{T_x}{\left(\frac{\partial g(|e_x|^{m_x})}{\partial e_x}\right)^{-1}} de_x \\ &= T_x\left(1 - exp\left[-\frac{1}{2}\left(\frac{e_{x0}{}^{m_x}}{\sigma_x}\right)^2\right]\right)\end{aligned} \tag{29}$$

$$\begin{aligned}T_2 &= \int_0^{e_{y0}} \frac{T_y}{\left(\frac{\partial g(|e_y|^{m_y})}{\partial e_y}\right)^{-1}} de_y \\ &= T_y\left(1 - exp\left[-\frac{1}{2}\left(\frac{e_{y0}{}^{m_y}}{\sigma_y}\right)^2\right]\right)\end{aligned} \tag{30}$$

$$\begin{aligned}T_3 &= \int_0^{e_{\theta 0}} \frac{T_\theta}{\left(\frac{\partial g(|e_\theta|^{m_\theta})}{\partial e_\theta}\right)^{-1}} de_\theta \\ &= T_\theta\left(1 - exp\left[-\frac{1}{2}\left(\frac{e_{\theta 0}{}^{m_\theta}}{\sigma_\theta}\right)^2\right]\right)\end{aligned} \tag{31}$$

When e_{x0}, e_{y0} and $e_{\theta 0}$ are ∞, $T_1 = T_x$,$T_2 = T_y$,$T_3 = T_\theta$, so the origin is predefined-time stable.

For the inner loop controller, consider the following Lyapunov function.

$$V = \frac{1}{2}\boldsymbol{s}^T M\boldsymbol{s} + \frac{1}{2}\tilde{\boldsymbol{\theta}}^T \Gamma_r \tilde{\boldsymbol{\theta}} \tag{32}$$

where $s = \omega_d - \omega$, $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}$.

The derivative of (32) is

$$\dot{V} = \boldsymbol{s}^T M\dot{\boldsymbol{s}} - \tilde{\boldsymbol{\theta}}^T \Gamma_r \dot{\hat{\boldsymbol{\theta}}} \tag{33}$$

Substituting (12) into (33) gives (34).

$$\dot{V} = \boldsymbol{s}^T\left(\boldsymbol{F}(\boldsymbol{\omega}) - \hat{\boldsymbol{F}}(\boldsymbol{\omega} \mid \boldsymbol{\theta}) - MG - K_d\boldsymbol{s}\right) - \tilde{\boldsymbol{\theta}}^T \Gamma_r \dot{\hat{\boldsymbol{\theta}}} \tag{34}$$

According to (18), (35) can be obtained.

$$\tilde{\boldsymbol{F}} = \boldsymbol{F}(\boldsymbol{\omega}) - \hat{\boldsymbol{F}}(\boldsymbol{\omega} \mid \boldsymbol{\theta}) = \tilde{\boldsymbol{\theta}}\boldsymbol{\zeta}(\boldsymbol{\omega}) + \varphi \tag{35}$$

Substitute (19) and (35) into (34) to obtain (36).

$$\dot{V} = -\boldsymbol{s}^T k_d \boldsymbol{s} - sMG + \boldsymbol{s}^T\varphi \tag{36}$$

Since $sMG \geq 0$ and φ is bounded, selecting an appropriate parameter k_d can stabilize the system.

5. Simulation

In order to verify the function of the controller, a circular trajectory tracking simulation is carried out. The selected model parameters are shown in Table 2. The selection of controller parameters is shown in Table 3.

Table 2. Physical parameters.

Parameters	
$l_x = 0.16$ m	$l_y = 0.12$ m
$m_r = 5.5$ kg	$R = 0.05$ m
$I_\omega = 6.921$ kg$\cdot$ m^2	$D_\omega = diag(1,1,1,1)$

Table 3. Actual controller parameters.

	Parameters
Outerloop	$T_x = 1, T_y = 0.5, T_\theta = 1$ $m_x = m_y = m_\theta = 0.2$ $\sigma_x = \sigma_y = \sigma_\theta = 0.2$
Inner loop	$T_{\omega_1} = T_{\omega_2} = T_{\omega_3} = T_{\omega_4} = 0.2$ $m_{\omega_1} = m_{\omega_2} = m_{\omega_3} = m_{\omega_4} = 0.2$ $\sigma_{\omega_1} = \sigma_{\omega_2} = \sigma_{\omega_3} = \sigma_{\omega_4} = 0.2$ $k_d = 2, \Gamma_r = 0.01$
fuzzy set	c = $[-100, -50, 0, 50, 100]$ $\sigma_i = 5$
Kalman filter	$Q = diag(20,20,20)$, $R = diag(0.01,0.01,0.01)$

5.1. Tracking Effect

The reference trajectory and uncertainty function are shown in (37) and (38), respectively.

$$\begin{cases} x = sin(t) \\ y = cos(t) \\ \theta = 0 \end{cases} \tag{37}$$

$$F(\omega) = \begin{bmatrix} 15\sin(\omega_1) \\ 20\cos(\omega_2) + 0.6t \\ 3\omega_3 \\ 2\omega_4 \end{bmatrix} \tag{38}$$

Figure 6 shows the trajectory of three different methods. In the figure, "T1FNNPTC" represents the control scheme proposed in this paper. "FPFNNSMC" is the combined controller of feedforward proportional and fuzzy adaptive sliding mode. "SMC" is the sliding mode controller. It can be seen from the figure that the tracking accuracy of the sliding mode controller is worse than that of the other two because of the absence of T1FNN. The tracking accuracy of FPFNNSMC and T1FNNPTC is similar, but the effect of T1FNNPTC is better at the initial stage.

In addition, the tracking error curve of each component are respectively shown in Figures 7–9. These figures show the error convergence process more intuitively. The error curve of SMC has the largest fluctuation amplitude, while the error amplitude of T1FNNPTC and FPFNNSMC is relatively small. According to Figures 7–9 the following Table 4 can be obtained. It can be seen from the data in the table that in the convergence stage of the tracking error, the overshoot peak value and settling time of the proposed method are smaller than those of FPFNNSMC, indicating that the performance of T1FNNPTC in the initial stage is better than that of FPFNNSMC.

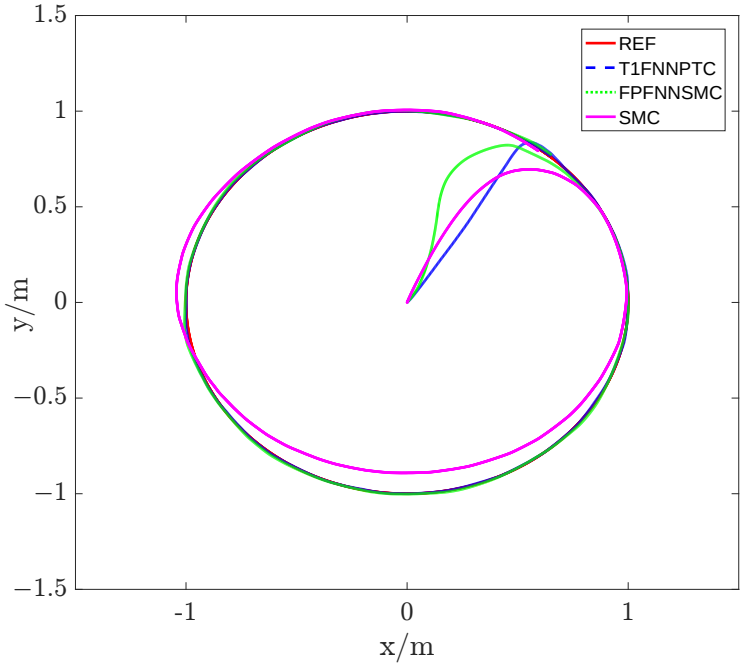


Figure 6. Circular trajectory tracking.

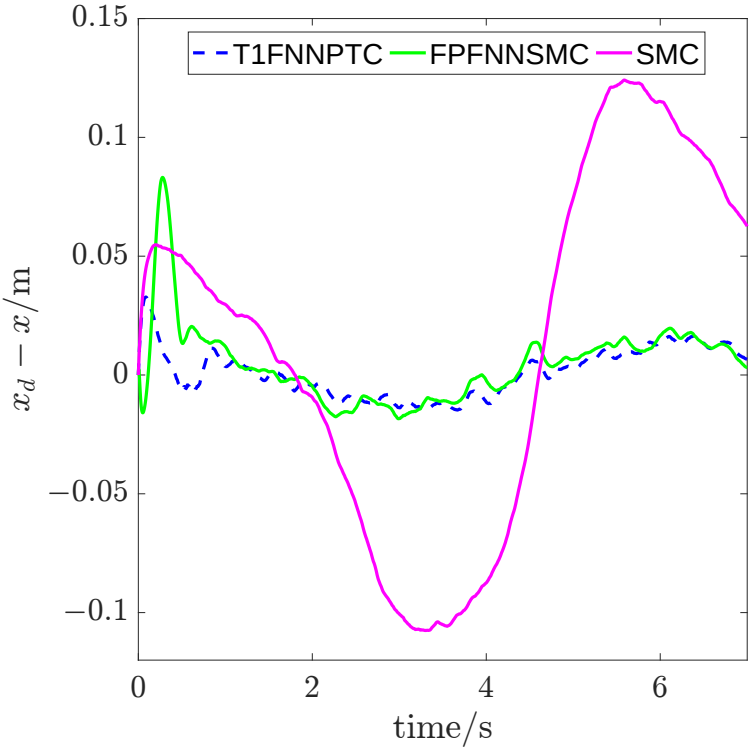


Figure 7. Tracking error of X component.

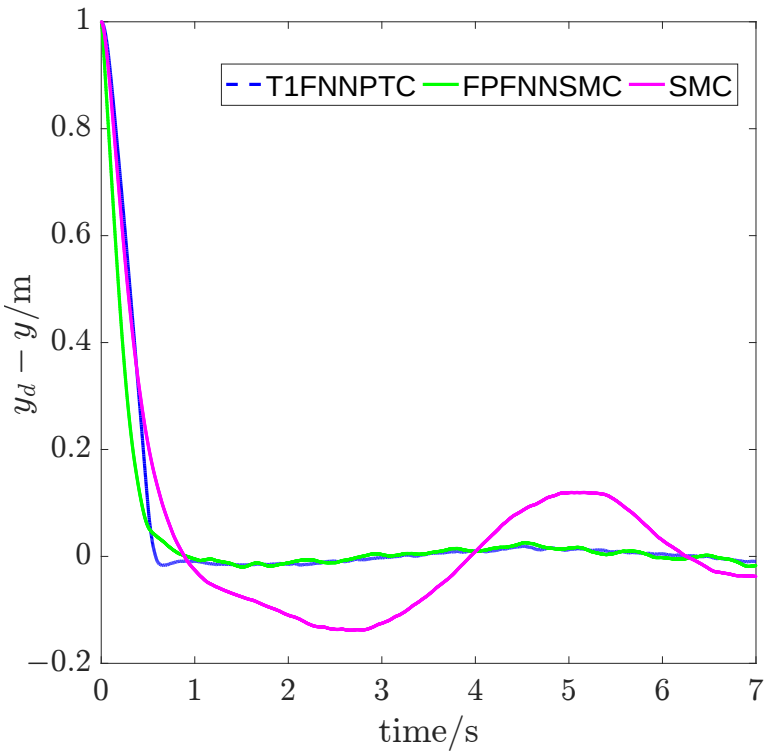


Figure 8. Tracking error of Y component.

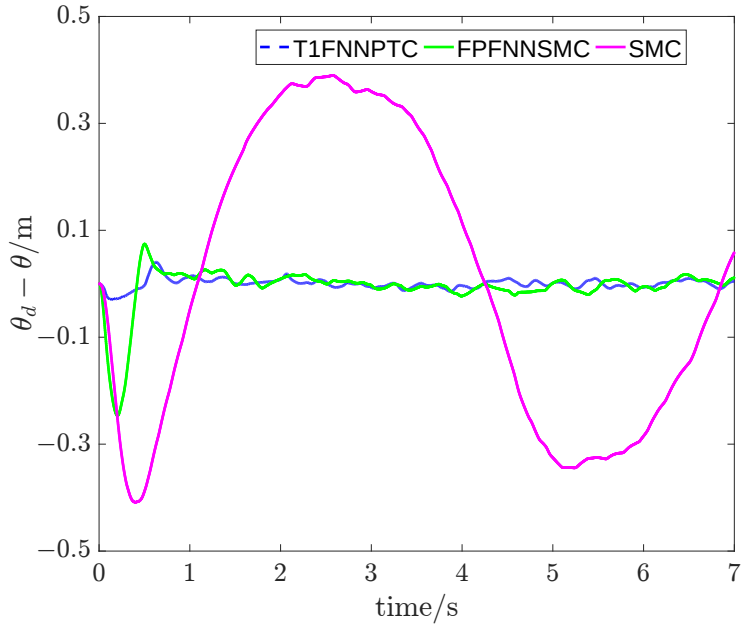


Figure 9. Tracking error of θ component.

In addition to position tracking, angular velocity (wheel 1) tracking is shown in Figure 10.

It can be seen from the figure that the angular velocity error curve of T1FNNPTC converges faster and has less overshoot than the other two methods.The above results show that the control method proposed in this paper has a certain effect on tracking.

Table 4. Performance parameters.

	T1FNNPTC	**FPFNNSMC**
$x_d - x$	Overshoot peak = 0.03279 Settling time = 0.2 s	Overshoot peak = 0.08314 Settling time = 0.4654 s
$y_d - x$	Overshoot peak = -0.01 Settling time = 0.554 s	Overshoot peak = 0 Settling time = 0.714 s
$\theta_d - x$	Overshoot peak = 0.03 Settling time = 0.14 s	Overshoot peak = -0.246 Settling time = 0.404 s

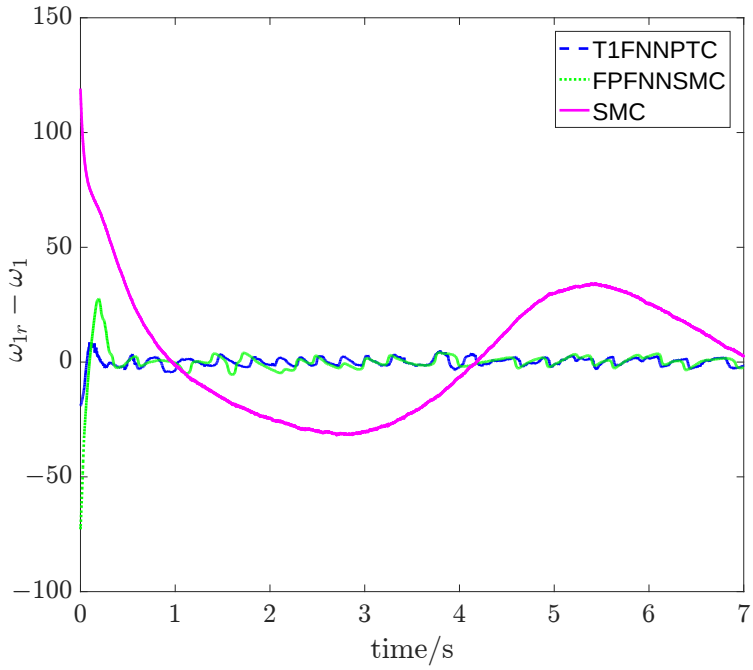


Figure 10. Tracking error of ω_1.

5.2. The Role of T1FNN

T1FNN is mainly used to approximate the uncertainty and compensate it. Because it acts on the inner loop, the effect of compensation is directly reflected in the tracking accuracy of angular velocity. To verify the effect of T1FNN, the method with network is compared with the method without network, and the angular velocity tracking error curve is shown in the Figure 11.

It can be seen from the figure that compared with the method without network compensation, the method with T1FNN has smaller tracking error of angular velocity, indicating that T1FNN has a certain compensation effect. In addition, in trajectory tracking control, tracking error related indicators are generally used to describe the tracking effect, mainly including Integrated Time weighted Absolute Error (ITAE), Integrated Absolute Error (IAE), and Integrated Square Error (ISE). In order to quantitatively analyze the steady-state tracking accuracy and reflect the impact of T1FNN on the tracking accuracy, the corresponding error indicators are calculated and compared, and the results are shown in Figure 12.

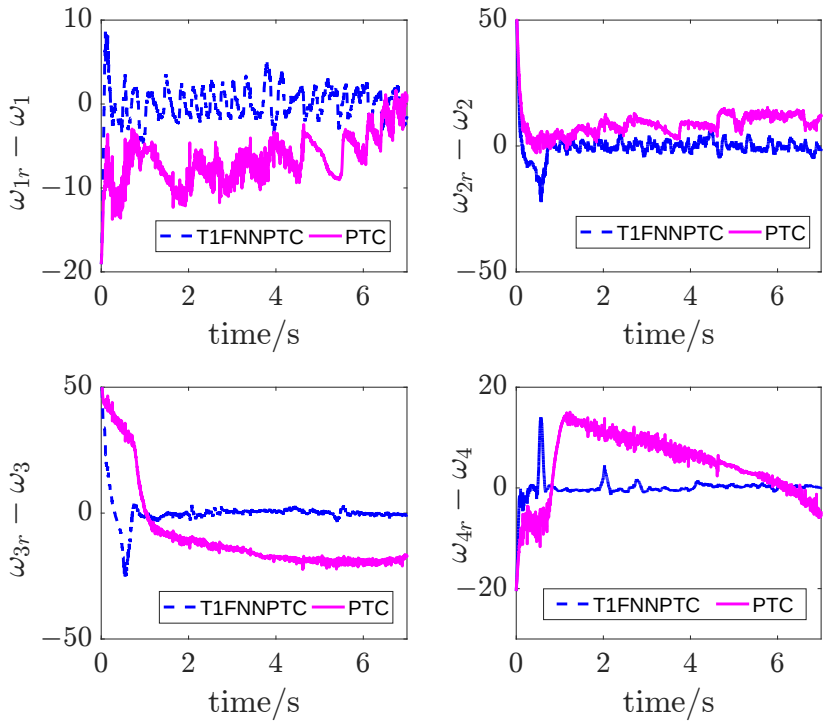


Figure 11. Tracking error of angular velocity.

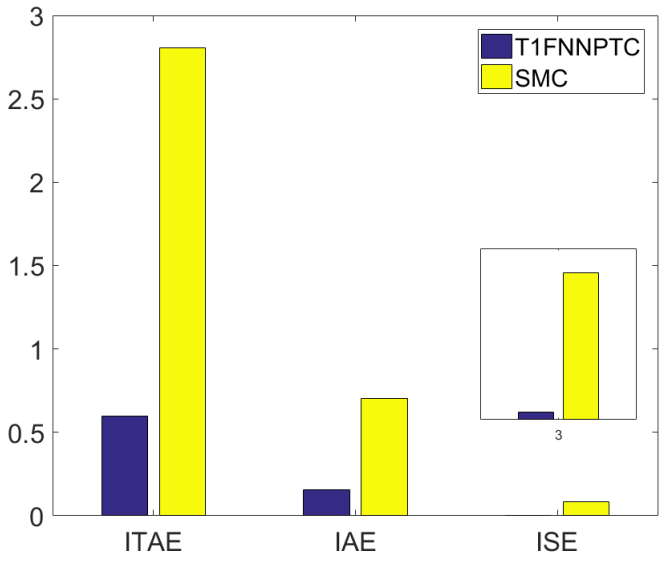


Figure 12. Error index.

5.3. Filtering Effect

In order to verify the function of the filter, the feedback signals in the filtered and unfiltered states are compared, as shown in Figures 13–15. The subscript "real" in the figure represents the true value, "mea" represents the measured value, and "eat" represents the estimated value. It can be clearly seen from the figure that the measured value fluctuates greatly after adding noise, and the estimated value is relatively stable under the effect of the filter. The estimated value still has some error, but it is better than the unfiltered signal in general.

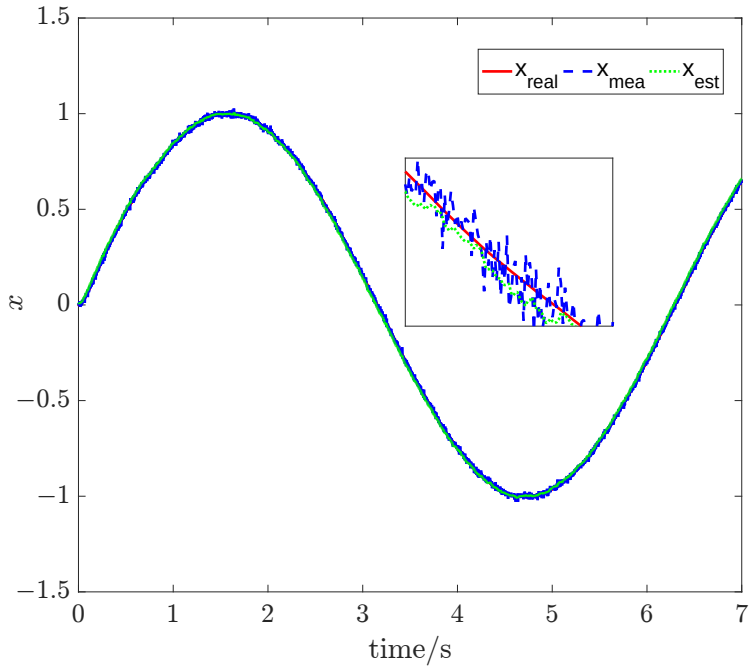


Figure 13. Signal x.

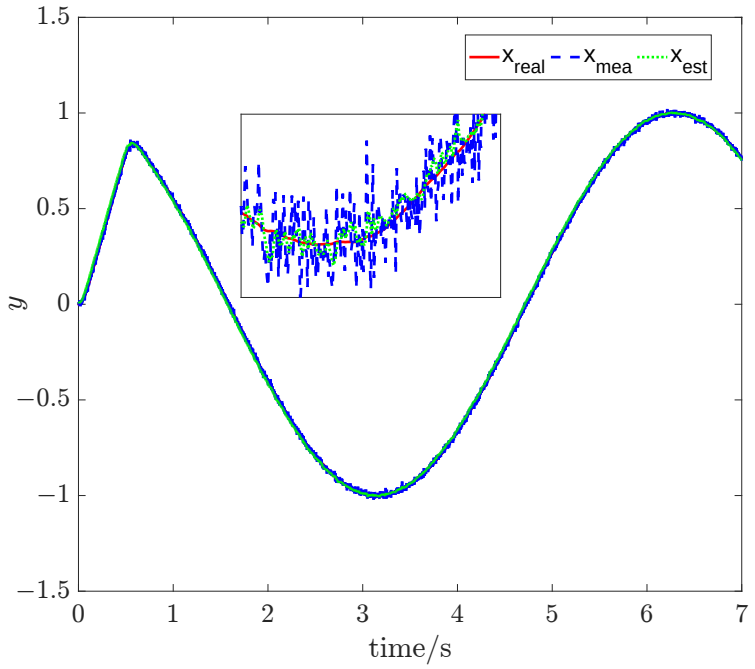


Figure 14. Signal y.

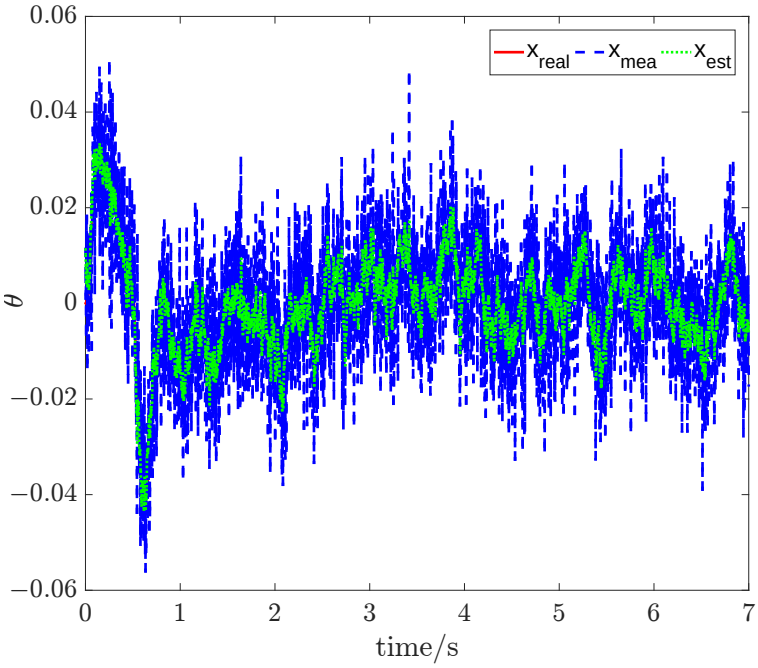


Figure 15. Signal θ.

6. Conclusions

Aiming at the problem of control time, model uncertainty and feedback channel noise of FM-OMR, this paper proposes a trajectory tracking control method based on fuzzy neural network with predefined-time stability. Based on the position error model and angular velocity error model, a position and angular velocity tracking controller with predefined-time is proposed to achieve the convergence of the position error within the specified time limit. A model uncertainty approximator based on fuzzy neural network is designed to compensate the output of the controller and improve the tracking accuracy of angular velocity. In addition, a Kalman filter is designed for the feedback channel noise to filter the noise and improve the signal quality. Finally, the simulation results verify the effect of this method.

Author Contributions: Conceptualization, P.Q., T.Z. and N.L.; methodology, P.Q., T.Z., N.L. and Z.M.; software, P.Q. and W.Y.; validation, P.Q. and N.L.; formal analysis, P.Q., Z.M. and W.Y.; investigation, P.Q. and N.L.; resources, N.L. and T.Z.; data curation, P.Q. and Z.M.; writing—original draft preparation, P.Q. and Z.M.; writing—review and editing, P.Q., T.Z. and Z.M.; visualization, W.Y. and Z.M.; supervision, T.Z. and W.Y.; project administration, P.Q. and T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Sichuan Science and Technology Program (2021ZYD0016).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guettari, M.; Gharbi, I.; Hamza, S. UVC disinfection robot. *Environ. Sci. Pollut. Res.* **2021**, *28*, 40394–40399. [CrossRef] [PubMed]
2. Murphy, R.R. By the time we build robots that care for us, will we be able to care for them? *Sci. Robot.* **2021**, *6*, eabj9789. [CrossRef] [PubMed]
3. Schilling, K.; Jungius, C. Mobile robots for planetary exploration. *Control Eng. Pract.* **1996**, *4*, 513–524. [CrossRef]

4. Baudoin, Y.; Habib, M.; Doroftei, I. Introduction: Mobile robotics systems for humanitarian de-mining and risky interventions. In *Using Robots in Hazardous Environments*; Elsevier: Amsterdam, The Netherlands, 2011; pp. 3–31.
5. Ruangpayoongsak, N.; Roth, H.; Chudoba, J. Mobile robots for search and rescue. In Proceedings of the IEEE International Safety, Security and Rescue Rototics, Workshop, Kobe, Japan, 6–9 June 2005; pp. 212–217.
6. Schneier, M.; Schneier, M.; Bostelman, R. *Literature Review of Mobile Robots for Manufacturing*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015.
7. Aravind, K.R.; Raja, P.; Pérez-Ruiz, M. Task-based agricultural mobile robots in arable farming: A review. *Span. J. Agric. Res.* **2017**, *15*, e02R01. [CrossRef]
8. Gao, X.; Li, J.; Fan, L.; Zhou, Q.; Yin, K.; Wang, J.; Song, C.; Huang, L.; Wang, Z. Review of wheeled mobile robots' navigation problems and application prospects in agriculture. *IEEE Access* **2018**, *6*, 49248–49268. [CrossRef]
9. Rotondo, D.; Puig, V.; Nejjari, F.; Romera, J. A fault-hiding approach for the switching quasi-LPV fault-tolerant control of a four-wheeled omnidirectional mobile robot. *IEEE Trans. Ind. Electron.* **2014**, *62*, 3932–3944. [CrossRef]
10. Lu, X.; Zhang, X.; Zhang, G.; Jia, S. Design of adaptive sliding mode controller for four-Mecanum wheel mobile robot. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 3983–3987.
11. 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]
12. Chen, C.P.; Ren, C.E.; Du, T. Fuzzy observed-based adaptive consensus tracking control for second-order multiagent systems with heterogeneous nonlinear dynamics. *IEEE Trans. Fuzzy Syst.* **2015**, *24*, 906–915. [CrossRef]
13. Alakshendra, V.; Chiddarwar, S.S. Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties. *Nonlinear Dyn.* **2017**, *87*, 2147–2169. [CrossRef]
14. Wang, C.; Liu, X.; Yang, X.; Hu, F.; Jiang, A.; Yang, C. Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy. *Appl. Sci.* **2018**, *8*, 231. [CrossRef]
15. Yuan, Z.; Tian, Y.; Yin, Y.; Wang, S.; Liu, J.; Wu, L. Trajectory tracking control of a four mecanum wheeled mobile platform: An extended state observer-based sliding mode approach. *IET Control Theory Appl.* **2020**, *14*, 415–426. [CrossRef]
16. Kwon, H.; Kim, Y. Toward Selective Membership Inference Attack against Deep Learning Model. *IEICE Trans. Inf. Syst.* **2022**, *105*, 1911–1915. [CrossRef]
17. Kwon, H.; Cho, C.; Lee, J. Priority Evasion Attack: An Adversarial Example That Considers the Priority of Attack on Each Classifier. *IEICE Trans. Inf. Syst.* **2022**, *105*, 1880–1889. [CrossRef]
18. Kwon, H. Defending Deep Neural Networks against Backdoor Attack by Using De-trigger Autoencoder. *IEEE Access* **2021**, *early access*. [CrossRef]
19. Ma, C.; Li, X.; Xiang, G.; Dian, S. A TS Fuzzy Quaternion-Value Neural Network-Based Data-Driven Generalized Predictive Control Scheme for Mecanum Mobile Robot. *Processes* **2022**, *10*, 1964. [CrossRef]
20. Li, J.; Wang, J.; Peng, H.; Zhang, L.; Su, H. Neural fuzzy approximation enhanced autonomous tracking control of the wheel-legged robot under uncertain physical interaction. *Neurocomputing* **2020**, *410*, 342–353. [CrossRef]
21. Zhang, Y.; Niu, B.; Zhao, X.; Duan, P.; Wang, H.; Gao, B. Global Predefined-Time Adaptive Neural Network Control for Disturbed Pure-Feedback Nonlinear Systems With Zero Tracking Error. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *early access*. [CrossRef]
22. Liu, Y.; Yan, W.; Zhang, T.; Yu, C.; Tu, H. Trajectory tracking for a dual-arm free-floating space robot with a class of general nonsingular predefined-time terminal sliding mode. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 3273–3286. [CrossRef]
23. Wu, C.; Yan, J.; Shen, J.; Wu, X.; Xiao, B. Predefined-time attitude stabilization of receiver aircraft in aerial refueling. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3321–3325. [CrossRef]
24. Zhang, Y.; Wang, Y.W.; Xiao, J.W.; Liu, X.K. Predefined-time secondary control for DC microgrid. *IEEE Trans. Ind. Electron.* **2021**, *69*, 13504–13513. [CrossRef]
25. Zhao, T.; Tong, W.; Mao, Y. Hybrid Non-singleton Fuzzy Strong Tracking Kalman Filtering for High Precision Photoelectric Tracking System. *IEEE Trans. Ind. Inform.* **2022**, *early access*. [CrossRef]
26. Zhao, T.; Chen, C.; Cao, H. Evolutionary self-organizing fuzzy system using fuzzy-classification-based social learning particle swarm optimization. *Inf. Sci.* **2022**, *606*, 92–111. [CrossRef]
27. Zhao, T.; Cao, H.; Dian, S. A Self-Organized Method for a Hierarchical Fuzzy Logic System based on a Fuzzy Autoencoder. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 5104–5115. [CrossRef]
28. Mei, Z.; Zhao, T.; Liu, N. Deep Hierarchical Interval Type 2 Self-Organizing Fuzzy System for Data-Driven Robot Control. *Processes* **2022**, *10*, 2091. [CrossRef]
29. Zhao, J.; Zhao, T.; Liu, N. Fractional-Order Active Disturbance Rejection Control with Fuzzy Self-Tuning for Precision Stabilized Platform. *Entropy* **2022**, *24*, 1681. [CrossRef]
30. Zhao, T.; Chen, C.; Cao, H.; Dian, S.; Xie, X. Multiobjective Optimization Design of Interpretable Evolutionary Fuzzy Systems With Type Self-Organizing Learning of Fuzzy Sets. *IEEE Trans. Fuzzy Syst.* **2022**, *early access*. [CrossRef]
31. Sun, Z.; Hu, S.; Li, N.; He, D. Trajectory-Following Control of Mecanum-Wheeled AGV Using Fuzzy Nonsingular Terminal Sliding Mode. In Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, 18–20 December 2020; pp. 342–347.
32. Liang, C.D.; Ge, M.F.; Liu, Z.W.; Zhan, X.S.; Park, J.H. Predefined-time stabilization of TS fuzzy systems: A novel integral sliding mode based approach. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 4423–4433. [CrossRef]

33. Alshorman, A.M.; Alshorman, O.; Irfan, M.; Glowacz, A.; Muhammad, F.; Caesarendra, W. Fuzzy-based fault-tolerant control for omnidirectional mobile robot. *Machines* **2020**, *8*, 55. [CrossRef]
34. Pizá, R.; Carbonell, R.; Casanova, V.; Cuenca, Á.; Salt Llobregat, J.J. Nonuniform Dual-Rate Extended Kalman-Filter-Based Sensor Fusion for Path-Following Control of a Holonomic Mobile Robot with Four Mecanum Wheels. *Appl. Sci.* **2022**, *12*, 3560. [CrossRef]
35. Xu, Y.; Liu, T.; Sun, B.; Zhang, Y.; Khatibi, S.; Sun, M. Indoor Vision/INS Integrated Mobile Robot Navigation Using Multimodel-Based Multifrequency Kalman Filter. *Math. Probl. Eng.* **2021**, *2021*, 6694084. [CrossRef]
36. Tsai, C.C.; Tai, F.C.; Lee, Y.R. Motion controller design and embedded realization for Mecanum wheeled omnidirectional robots. In Proceedings of the 2011 9th World Congress on Intelligent Control and Automation, Taipei, Taiwan, 21–25 June 2011; pp. 546–551. [CrossRef]
37. Razzaghian, A. A fuzzy neural network-based fractional-order Lyapunov-based robust control strategy for exoskeleton robots: Application in upper-limb rehabilitation. *Math. Comput. Simul.* **2022**, *193*, 567–583. [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.