

Article

# Path Tracking Control of an Omni-Directional Service Robot Based on Model Predictive Control of Adaptive Neural-Fuzzy Inference System

Fenglin Pang<sup>1,2</sup>, Minzhou Luo<sup>1,2,\*</sup>, Xiaobin Xu<sup>1,2</sup>  and Zhiying Tan<sup>1,2</sup>

<sup>1</sup> College of Mechanical & Electrical Engineering, Hohai University, Changzhou 213022, China; pfl9491@163.com (F.P.); xxbtc@hhu.edu.cn (X.X.); zytan@hhu.edu.cn (Z.T.)

<sup>2</sup> Jiangsu Key Laboratory of Special Robot Technology, Hohai University, Changzhou 213022, China

\* Correspondence: lmz@hhuc.edu.cn; Tel.: +86-1843-924-9491

**Abstract:** In this paper, model predictive control (MPC) based on an adaptive neural-fuzzy inference system (ANFIS) is proposed to realize control of an omni-directional service robot in path tracking. The weight of the cost function in a traditional MPC needs to be manually adjusted, and it is difficult to adjust to a satisfactory value. In order to improve the performance and the control accuracy of MPC, a fuzzy system trained by ANFIS is used to adaptively adjust the weight of MPC's cost function to reduce the error in the process of path tracking. The different simulation experiments are conducted to verify the performance of the proposed algorithm. The experimental results show that the distance error of MPC based on ANFIS is reduced by more than 50% under different paths compared with a traditional MPC, and the angle error is reduced by more than 70%. Meanwhile, the stability is increased by around 60%. The results show the feasibility and superiority of MPC based on ANFIS.

**Keywords:** omni-directional chassis; path tracking; model predictive control; adaptive neural-fuzzy inference system; weight of cost function



**Citation:** Pang, F.; Luo, M.; Xu, X.; Tan, Z. Path Tracking Control of an Omni-Directional Service Robot Based on Model Predictive Control of Adaptive Neural-Fuzzy Inference System. *Appl. Sci.* **2021**, *11*, 838. <https://doi.org/10.3390/app11020838>

Received: 15 November 2020

Accepted: 13 January 2021

Published: 18 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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

In recent years, the technology of service robots has been developed widely, and it has been a research hotspot [1,2]. Among these, path tracing is crucial.

Path tracing methods mainly include proportion integration differentiation (PID), fuzzy control (FC), proportion integration differentiation fuzzy (FPID), model predictive control (MPC), fuzzy model predictive control (FMPC), etc. In [2], a simple and effective PID robust tuning method is proposed and applied to a simple mobile robot model with uncertain delay estimation. Although experiments show that it has good path tracking performance, it cannot cope with multiple controlled quantities simultaneously. Moreover, it is not suitable for complex nonlinear time-varying systems. In [3], a path tracking and heading adjustment system for a Mecanum wheel robot based on a fuzzy logic controller (FLC) is proposed. It can achieve better path tracking control without external references such as beacons or visual marks. However, it simplifies a Mecanum wheel chassis to a differential wheel chassis during the control process. Additionally, the system cannot control the rotation angle of the chassis, which makes the chassis lose the flexibility unique to the Mecanum wheel chassis. In [4], an equivalence relationship between fuzzy proportional-integral-derivative (PID) controllers and conventional PID controllers is put forward, the designer can further tune the membership functions of fuzzy variables in the control rules to acquire more superior performance. In [5], a self-tuning fuzzy PID controller, which is easy to be realized on a single-chip microcomputer, is presented. Even though it can control irregular lines precisely, the speed response of a single motor is insufficient in a short time. In [6], a fuzzy controller is designed to minimize the path tracking error, and a cuckoo

optimization algorithm (COA) is used to optimize the fuzzy controller. Experiments show that it can achieve accurate path tracking.

Predictive control is a novel type of control algorithm in the field of industrial process control. It is not only a representative advanced control algorithm but also a theoretical system of stability and robust design of uncertain systems with rolling optimization characteristics [7]. In [8], a MPC strategy for non-holonomic constrained wheeled mobile robot (WMR) is put forward. The kinematic model of the WMR is linearized. According to the constraints of the status and input, WMR control law can achieve optimal control. In [9–12], MPC applied to an omni-directional wheeled chassis is introduced. MPC can realize accurate control of complex paths. Nevertheless, some parameters of MPC need to be manually adjusted, which is time-consuming and labor-intensive. If the tuning is inaccurate, it is difficult to achieve optimal path tracking. Thus, there is more research and development required in the direction of adaptive tuning. In [13], an adaptive fuzzy nonlinear model predictive control (FNMP) strategy for trajectory tracking of the WMR is proposed. An adaptive fuzzy system is obtained through training and applied to the approximation of the WMR model. The parameters of the fuzzy model are adjusted online by using the gradient descent algorithm, which can minimize the tracking error of a given trajectory. In [14], an improved MPC strategy is designed. The improved MPC also can adaptively adjust the weight of the cost function through fuzzy control, and it is superior to a traditional MPC in terms of tracking accuracy and steering stability. However, operators need to summarize a complete set of fuzzy control rules through a large amount of practical experience. If the fuzzy system is simple or wrong, it may lead to the reduction of control accuracy, even the deterioration of dynamic quality. Therefore, it is more difficult to build complete fuzzy rules.

In [15], an adaptive network fuzzy inference system (ANFIS) architecture and learning process are explored. This system can realize the establishment of a fuzzy inference system in the framework of an adaptive network. It uses a hybrid learning algorithm or other learning algorithms to train and learn data set. A superior fuzzy system can also be obtained by optimizing the premise and consequence parameters. In [16], a control of a pilot-scale conversion hydrogen production unit is studied. ANFIS is applied to MPC to realize an optimal control strategy. The process is trained and learned by ANFIS, and the membership function is optimized using empirical training data.

In order to ensure high tracking accuracy and stability, and without the need for operators to summarize a complete set of fuzzy control rules, this paper puts forward a strategy of MPC based on ANFIS. This strategy can adaptively adjust the weight of the cost function in MPC to achieve accurate path tracking.

The rest of this paper is organized as follows: Section 2 describes the kinematic model of the omni-directional wheel chassis. In Section 3, the design of the control system is proposed. Section 4 is the simulation analysis. The conclusion of this paper is presented in Section 5.

## 2. Kinematic Model

As shown in Figure 1, the local coordinate frame of the service robot chassis is  $X_R P Y_R$ , the global coordinate frame is  $X_I O Y_I$ , and the point  $P$  is the reference point on the chassis.

In the global coordinate frame, the position  $P$  is determined by  $X_I$  and  $Y_I$  [17–20], and the position of the chassis in the local coordinate frame is:

$$\zeta_R = R(\theta) \bullet \zeta_I \quad (1)$$

where  $\theta$  is the angle between the global coordinate frame and the local coordinate frame,  $\xi_I = [x_I \ y_I \ \theta_I]^T$  is the position of the robot in the global coordinate frame, and  $\xi_R$  is  $\xi_R = [x_R \ y_R \ \theta_R]^T$ .  $R(\theta)$  is the rotation matrix.  $R(\theta)$  is:

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

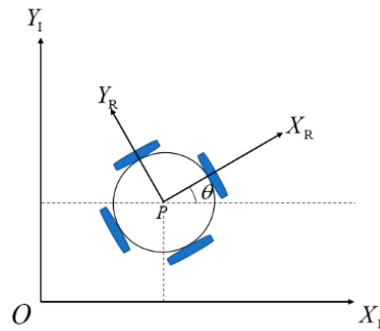


Figure 1. Global coordinate and local coordinate frames.

The rolling constraint of the Mecanum wheel forces all rollers to move along the ground direction. There must be a certain number of rollers rotating so that there is pure rolling at the contact point [10,21]. The velocity direction of the wheels is decomposed and deduced:

$$jR(\theta)\dot{\xi}_I = r\dot{\varphi} \cos \gamma \tag{3}$$

where  $j = [ \sin(\alpha + \beta + \gamma) \quad -\cos(\alpha + \beta + \gamma) \quad -l \cos(\beta + \gamma) ]$ , as shown in Figure 2,  $l$  is the distance between every Mecanum wheel and the point  $P$ , and  $r$  is the radius of the Mecanum wheel.

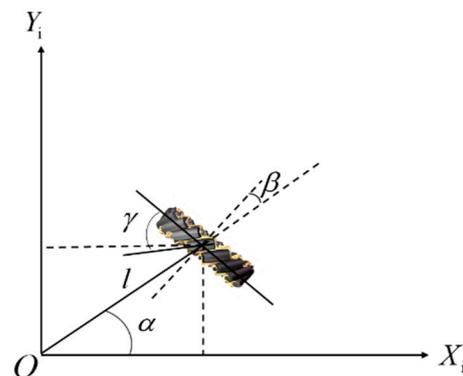


Figure 2. A schematic diagram of the motion of a single wheel.

The rolling constraints of an individual Mecanum wheel can be grouped into the single Equations (4) and (5).

$$J_1R(\theta)\dot{\xi}_I - J_2\dot{\varphi} = 0 \tag{4}$$

$$\dot{\varphi} = J_2^{-1}J_1R(\theta)\dot{\xi}_I \tag{5}$$

where  $\dot{\varphi}(t)$  is the velocity function of the horizontal axis of the Mecanum wheel.

$$J_1 = \begin{bmatrix} \sin(\alpha_1 + \beta_1 + \gamma_1) & -\cos(\alpha_1 + \beta_1 + \gamma_1) & -l \cos(\beta_1 + \gamma_1) \\ \sin(\alpha_2 + \beta_2 + \gamma_2) & -\cos(\alpha_2 + \beta_2 + \gamma_2) & -l \cos(\beta_2 + \gamma_2) \\ \sin(\alpha_3 + \beta_3 + \gamma_3) & -\cos(\alpha_3 + \beta_3 + \gamma_3) & -l \cos(\beta_3 + \gamma_3) \\ \sin(\alpha_4 + \beta_4 + \gamma_4) & -\cos(\alpha_4 + \beta_4 + \gamma_4) & -l \cos(\beta_4 + \gamma_4) \end{bmatrix} \quad (6)$$

$$J_2 = \begin{bmatrix} r \cos \gamma_1 & & & \\ & r \cos \gamma_2 & & \\ & & r \cos \gamma_3 & \\ & & & r \cos \gamma_4 \end{bmatrix} \quad (7)$$

Assuming  $J = J_2^{-1} J_1 R(\theta)$ , the velocity is:

$$\dot{\zeta}_I = (J^T J)^{-1} J^T \dot{\varphi} \quad (8)$$

### 3. Design of Control System

The control system schematic diagram is shown in Figure 3. The schematic diagram can be described as a MPC design and ANFIS design.

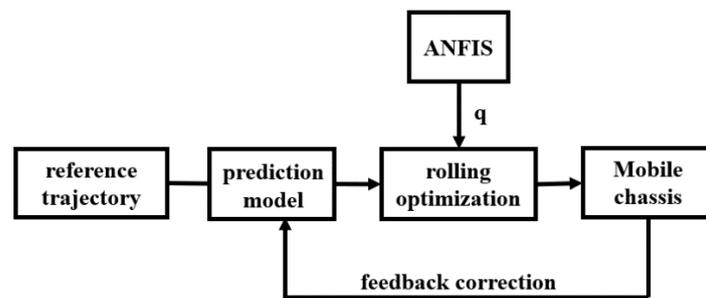


Figure 3. The control system schematic diagram.

#### 3.1. MPC

MPC can solve complex control problems of multi-variable inputs [22]. MPC consists of the following three parts: linearization and approximate discretization of a kinematic model, design of objective function, and design of constraint conditions [23,24].

##### 3.1.1. Linearization and Approximate Discretization

Define  $\dot{\varphi}$  as  $u$ , Equation (8) can be simplified as:

$$\dot{\zeta} = f(\zeta, u) \quad (9)$$

First-order Taylor expansion is performed at a reference point in Equation (9), which was rewritten as:

$$\dot{\hat{\zeta}} = A \hat{\zeta} + B \hat{u} \quad (10)$$

where  $\dot{\zeta} - \dot{\zeta}_r = \dot{\hat{\zeta}}$ ,  $\zeta - \zeta_r = \hat{\zeta}$ ,  $u - u_r = \hat{u}$ ,  $\frac{\partial f}{\partial \zeta} \Big|_{\zeta = \zeta_r, u = u_r} = A$ ,  $\frac{\partial f}{\partial u} \Big|_{\zeta = \zeta_r, u = u_r} = B$ ,

$\dot{\zeta}_r = f(\zeta_r, u_r)$ ,  $\zeta_r$  and  $u_r$  are the reference pose and velocity which are the closest to the latest pose (Euclidean Distance).

Equation (10) can be approximately discretized as:

$$\hat{\zeta}((k+1)T) \approx \tilde{A} \hat{\zeta}(kT) + \tilde{B} \hat{u}(kT) \quad (11)$$

where  $\tilde{A} = I + TA, \tilde{B} = TB$ .

### 3.1.2. Design of Objective Function

Define  $kT$  as  $k$ , The state functions are:

$$\begin{cases} \chi(k+1) = \hat{A}_k\chi(k) + \hat{B}_k\Delta u(k) \\ \eta(k) = \hat{C}_k\chi(k) \end{cases} \tag{12}$$

where  $\chi(k) = \begin{bmatrix} \hat{\xi}(k) \\ u(k-1) \end{bmatrix}_{(m+n)\times 1}$ ,  $\hat{A}_k = \begin{bmatrix} \tilde{A}_{n\times n} & \tilde{B}_{n\times m} \\ 0_{m\times n} & I_m \end{bmatrix}_{(m+n)\times(m+n)}$ ,  $\hat{B}_k = \begin{bmatrix} \tilde{B}_{n\times m} \\ I_m \end{bmatrix}_{(m+n)\times m}$ ,  $\hat{C}_k = [ I_{n\times n} \quad 0_{n\times m} ]_{n\times(m+n)}$ ,  $m$  is the number of control quantity, and  $n$  is the number of status quantity.

At time  $k + 2$ , Equation (12) can be written as:

$$\begin{cases} \chi(k+2) = \hat{A}_{k+1}\chi(k+1) + \hat{B}_{k+1}\Delta u(k+1) \\ \eta(k+1) = \hat{C}_{k+1}\chi(k+1) \end{cases} \tag{13}$$

Substituting Equation (12) into Equation (13), the state functions can be expressed as:

$$\begin{cases} \chi(k+2) = \hat{A}_{k+1}\hat{A}_k\chi(k+1) + \hat{A}_{k+1}\hat{B}_k\Delta u(k) + \hat{B}_{k+1}\Delta u(k+1) \\ \eta(k+1) = \hat{C}_{k+1}\hat{A}_k\chi(k) + \hat{C}_{k+1}\hat{B}_k\Delta u(k) \end{cases} \tag{14}$$

By recursion in the above way, Equation (15) can be solved.

$$\left\{ \begin{aligned} \eta(k+1) &= \hat{C}_{k+1}\hat{A}_k\chi(k) + \hat{C}_{k+1}\hat{B}_k\Delta u(k) \\ \eta(k+2) &= \hat{C}_{k+2}\hat{A}_{k+1}\hat{A}_k\chi(k) + \hat{C}_{k+2}\hat{A}_{k+1}\hat{B}_k\Delta u(k) + \hat{C}_{k+2}\hat{B}_{k+1}\Delta u(k) \\ \eta(k+3) &= \hat{C}_{k+3}\hat{A}_{k+2}\hat{A}_{k+1}\hat{A}_k\chi(k) + \hat{C}_{k+3}\hat{A}_{k+2}\hat{A}_{k+1}\hat{B}_k\Delta u(k) + \hat{C}_{k+3}\hat{A}_{k+2}\hat{B}_{k+1}\Delta u(k) + \hat{C}_{k+3}\hat{B}_{k+2}\Delta u(k) \\ &\vdots \\ \eta(k+N_c) &= \hat{C}_{k+N_c} \prod_{i=0}^{N_c-1} \hat{A}_{k+i}\chi(k) + \hat{C}_{k+N_c} \prod_{i=2}^{N_c-1} \hat{A}_{k+i}\Delta u(k) + \dots + \hat{C}_{k+2}\hat{B}_{k+N_c-1}\Delta u(k) \\ &\vdots \\ \eta(k+N_p) &= \hat{C}_{k+N_p} \prod_{i=0}^{N_p-1} \hat{A}_{k+i}\chi(k) + \hat{C}_{k+N_p} \prod_{i=1}^{N_p-1} \hat{A}_{k+i}\Delta u(k) + \dots + \hat{C}_{k+N_p}\hat{B}_{k+N_p-1} \prod_{i=N_c}^{N_p-1} \hat{A}_{k+i}\Delta u(k) \end{aligned} \right. \tag{15}$$

Equation (16) can be abbreviated as:

$$Y(k) = \psi_k\chi(k) + \Theta_k\Delta U_k \tag{16}$$

$$\text{where } Y(k) = \begin{bmatrix} \eta(k+1) \\ \eta(k+2) \\ \dots \\ \eta(k+N_c) \\ \dots \\ \eta(k+N_p) \end{bmatrix}_{N_p \times 1}, \hat{C}_k = \hat{C}_{k+1} = \dots = \hat{C}_{k+N_p}, \psi_k = \begin{bmatrix} \hat{C}_k \hat{A}_k \\ \hat{C}_k \hat{A}_{k+1} \hat{A}_k \\ \dots \\ \hat{C}_k \prod_{i=0}^{N_c-1} \hat{A}_{k+i} \\ \dots \\ \hat{C}_k \prod_{i=0}^{N_p-1} \hat{A}_{k+i} \end{bmatrix}_{N_p \times 1},$$

$$\Theta_k = \begin{bmatrix} \hat{C}_k \hat{B}_k & 0 & \dots & 0 \\ \hat{C}_k \hat{A}_k \hat{B}_k & \hat{C}_k \hat{B}_k & 0 & 0 \\ \dots & \dots & \ddots & 0 \\ \hat{C}_k \hat{B}_k \prod_{i=1}^{N_c-1} \hat{A}_{k+i} & \hat{C}_k \hat{B}_{k+1} \prod_{i=2}^{N_c-2} \hat{A}_{k+i} & \dots & \hat{C}_k \hat{B}_{k+N_c-1} \\ \hat{C}_k \hat{B}_k \prod_{i=1}^{N_c} \hat{A}_{k+i} & \hat{C}_k \hat{B}_{k+1} \prod_{i=2}^{N_c} \hat{A}_{k+i} & \dots & \hat{C}_k \hat{A}_k \hat{B}_{k+N_c-1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_k \hat{B}_k \prod_{i=1}^{N_p-1} \hat{A}_{k+i} & \hat{C}_k \hat{B}_{k+1} \prod_{i=2}^{N_p-1} \hat{A}_{k+i} & \dots & \hat{C}_k \hat{B}_{k+N_c-1} \prod_{i=N_c}^{N_p-1} \hat{A}_{k+i} \end{bmatrix}_{N_p \times N_c},$$

$$\Delta U_k = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \dots \\ \Delta u(k+N_c-1) \end{bmatrix}_{N_c \times 1}, N_p \text{ is the prediction horizon, } N_c \text{ is the control horizon,}$$

and  $N_p > N_c$ .

The MPC in this paper was a relatively simplistic but standard non-linear MPC algorithm, in which all the  $\hat{A}_{k+i}$  were the same in the processing of the prediction model and all the  $\hat{B}_{k+i}$  were the also same [23].

According to the MPC strategy, the optimization problem should be converted into a quadratic programming (QP) problem [7]. The objective function was the quadratic function of the spatial state of the omni-directional wheel chassis and the control inputs. Additionally, the relaxation factor was introduced, which could not only directly limit the control increment, but also avoid the emergence of non-feasible solutions in the execution process [25]. The objective function is:

$$J(k) = (Y(k) - Y_{ref}(k))^T Q (Y(k) - Y_{ref}(k)) + \Delta U(k)^T R \Delta U(k) + \rho \varepsilon^2 \tag{17}$$

where  $Q$  is the weight matrix of  $Y$ ,  $Q = \text{diag}(Q_1, Q_2, \dots, Q_{N_c}, \dots, Q_{N_p})$ ,  $Q_i = \text{diag}(q_x, q_y, q_\varphi)$ , ( $i = 1, 2, \dots, N_c, \dots, N_p$ ),  $R$  is the weight matrix of  $\Delta U$ ,  $R = \text{diag}(R_0, R_1, \dots, R_{N_c-1})$ ,  $R_i = \text{diag}(r_{\Delta 1}, r_{\Delta 2}, r_{\Delta 3}, r_{\Delta 4})$ , ( $i = 0, 1, 2, \dots, N_c-1$ ),  $\rho$  is weight coefficient, and  $\varepsilon$  is the relaxation factor.

### 3.1.3. Design of Constraints

QP is a typical mathematical optimization problem. The reference trajectory tracked by the omni-directional wheel chassis was quicker and smoother by adding linear or non-linear constraints. Hence, the motion performance of the omni-directional wheel chassis was improved. The linear constraints were:

$$U_{\min} \leq \Gamma \Delta U(k) + U(k-1) \leq U_{\max} \tag{18}$$

where  $\Gamma = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \ddots & 0 \\ 1 & \dots & 1 \end{bmatrix}_{N_c \times N_c} \otimes I_m = \begin{bmatrix} I_m & 0_m & 0_m \\ \vdots & \ddots & 0_m \\ I_m & \dots & I_m \end{bmatrix}$  ( $\otimes$  is Kronecker product),

$U_{\min} = [u_{\min}, u_{\min}, \dots, u_{\min}]_{1 \times N_c}^T$ ,  $U_{\max} = [u_{\max}, u_{\max}, \dots, u_{\max}]_{1 \times N_c}^T$ ,  $U(k-1) = [u(k-1), u(k-1), \dots, u(k-1)]_{1 \times N_c}^T$ .

Equation (17) could be rewritten as:

$$J(k) = \begin{bmatrix} \Delta U(k)^T & \varepsilon \end{bmatrix} H(k) \begin{bmatrix} \Delta U(k)^T \\ \varepsilon \end{bmatrix} + G(k) \begin{bmatrix} \Delta U(k)^T \\ \varepsilon \end{bmatrix} + P(k) \tag{19}$$

where  $E(k) = \psi(k) - Y_{ref}(k)$ ,  $H(k) = \begin{bmatrix} \Theta(k)^T Q \Theta(k) & 0 \\ 0 & \rho \end{bmatrix}$ ,  $G(k) = \begin{bmatrix} 2E(k)^T Q \Theta(k) & 0 \end{bmatrix}$ ,

$P(k) = E(k)^T Q E(k)$ .

Equation (20) is:

$$\begin{cases} \begin{bmatrix} \Gamma(k) & 0 \\ -\Gamma(k) & 0 \end{bmatrix} \begin{bmatrix} \Delta U(k) \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} U_{\min} - U(k-1) \\ U_{\max} - U(k-1) \end{bmatrix} \\ \begin{bmatrix} \Delta U_{\min} \\ 0 \end{bmatrix} \leq \begin{bmatrix} \Delta U(k) \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} \Delta U_{\max} \\ M \end{bmatrix} \end{cases} \tag{20}$$

Therefore, the quadratic programming problem is:

$$\begin{cases} \min J(k) \rightarrow (J(k) = \begin{bmatrix} \Delta U(k)^T & \varepsilon \end{bmatrix} H(k) \begin{bmatrix} \Delta U(k)^T \\ \varepsilon \end{bmatrix} + G(k) \begin{bmatrix} \Delta U(k)^T \\ \varepsilon \end{bmatrix} + P(k)) \\ S.T \rightarrow \begin{cases} \begin{bmatrix} \Gamma(k) & 0 \\ -\Gamma(k) & 0 \end{bmatrix} \begin{bmatrix} \Delta U(k) \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} U_{\min} - U(k-1) \\ U_{\max} - U(k-1) \end{bmatrix} \\ \begin{bmatrix} \Delta U_{\min} \\ 0 \end{bmatrix} \leq \begin{bmatrix} \Delta U(k) \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} \Delta U_{\max} \\ M \end{bmatrix} \end{cases} \end{cases} \tag{21}$$

$\varepsilon$  is used as an independent variable, and is computed by Equation (21). A series of optimally controlled increments  $\Delta U(k)^*$  could be calculated in the control horizon by optimizing Equation (21) in each control cycle.

$$\Delta U(k)^* = [ \Delta u(k)^* \quad \Delta u(k+1)^* \quad \dots \quad \Delta u(k+N_c-1)^* ]^T \tag{22}$$

$$u(k) = u(k-1) + \Delta u(k)^* \tag{23}$$

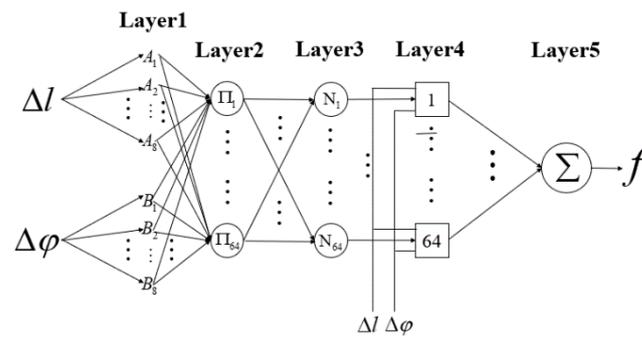
The first element acted as the actual controlled increment of  $\Delta U(k)^*$  in Equations (22) and (23).

### 3.2. Design of ANFI

ANFIS is a multi-inputs and single-output model. Based on experience and knowledge, a neural network was employed to optimize the premise and consequence parameters to adaptively establish a fuzzy inference system [15,26–29].

#### 3.2.1. Architecture of ANFIS

Assuming that ANFIS had two inputs  $\Delta l$  and  $\Delta \varphi$ , and one output  $q$ . As shown in Figure 4, ANFIS covered the input variables with 8 fuzzy subsets. The rules of the rule base are:



**Figure 4.** The architecture diagram of an adaptive neural-fuzzy inference system (ANFIS).

If  $\Delta l$  is  $A_i$  and  $\Delta \varphi$  is  $B_i$ , then  $f_i = p_i \Delta l + q_i \Delta \varphi + s_i, (i = 1, 2 \dots, 8)$ .

Where  $A_i(B_i)$  is the fuzzy subset of  $\Delta l(\Delta \varphi)$ ,  $p_i, q_i, s_i$  called the consequence parameters are the linear parameters of the function  $f_i$ .

The structure of each layer of ANFIS was as follows.

**Layer 1**

This layer is called the “fuzzification layer”. The nodes are adaptive.

$$O_i^1 = \mu_{A_i}(\Delta l) \text{ or } O_i^1 = \mu_{B_i}(\Delta \varphi), i = 1, 2 \dots, 8 \tag{24}$$

where  $\Delta l$  or  $\Delta \varphi$  is the input of the node  $i$ .  $A_i$  or  $B_i$  is the language tags associated with this node function,  $O_i^1$  is the membership function of  $A_i$  or  $B_i$ , which specifies how much given  $\Delta l(\Delta \varphi)$  satisfies  $A_i (B_i)$ . The superscript 1 of  $O_i^1$  indicates the first layer, and the subscript  $i$  indicates the  $i$ th node.

**Layer 2**

This layer is called the “rule layer”. The nodes were fixed and labeled as  $\Pi_i, (i = 1, 2 \dots, 64)$ . The nodes realized the process of fuzzy inference. The output, which represented the firing strength of a rule, was the product of all the incoming signals.

$$O_i^2 = \omega_i = \mu_{A_i}(\Delta l) \bullet \mu_{B_i}(\Delta \varphi), i = 1, 2 \dots, 8 \tag{25}$$

**Layer 3**

This layer is called the “normalization layer”. The nodes were fixed and labeled as  $N_i (i = 1, 2 \dots, 64)$ . Every node in this layer received all outputs from layer 2 and the normalized firing strength of the  $i$ th rule.

$$O_i^3 = \bar{\omega}_i = \frac{\omega_i}{\sum_{i=1}^{64} \omega_i}, i = 1, 2 \dots, 64 \tag{26}$$

**Layer 4**

This layer is called the “defuzzification layer”. The nodes were adaptive.

$$O_i^4 = \bar{\omega}_i f_i = \bar{\omega}_i (p_i \Delta l + q_i \Delta \varphi + s_i), i = 1, 2 \dots, 64 \tag{27}$$

where  $\bar{\omega}_i$  is a normalized firing strength value received from layer 3,  $\Delta l$  and  $\Delta \varphi$  are the input of the node  $i$ , and  $p_i, q_i, s_i$  are the linear parameters of the function  $f_i$ .

**Layer 5**

This layer is called the “summation layer”. The single node was fixed, and labeled as  $\Sigma$ . This layer calculated the sum of all outputs from layer 4 and provided an output of ANFIS.

$$O_1^5 = \sum_{i=1}^{64} \bar{\omega}_i f_i = \frac{\sum_{i=1}^{64} \omega_i f_i}{\sum_{i=1}^{64} \omega_i} \tag{28}$$

### 3.2.2. Collection of Data Set

It is a crucial process to collect a training data set for ANFIS. This paper used an ANFIS architecture of two-inputs and single-output. The three parameters of a collected data set are:

$$\begin{cases} \Delta l = \left( \sqrt{(l_{refX} - l_{realX})^2 + (l_{refY} - l_{realY})^2} \right) * 100 \\ \Delta \varphi = (\varphi_{ref} - \varphi_{real}) * 100 \\ q = \frac{q_{lX}}{q_\varphi} = \frac{q_{lY}}{q_\varphi} \end{cases} \tag{29}$$

where  $\Delta l$  represents the error between the reference XY distance and the actual XY distance,  $\Delta \varphi$  represents the error between the reference  $\varphi_{ref}$  and the actual  $\varphi_{real}$ , and the output  $q$  represents the weight ratio.

A traditional MPC was used to track a reference trajectory. At the sampling period  $T$ ,  $\Delta l$  and  $\Delta \varphi$  were collected, and the weight ratio  $q$  was adjusted to reduce the error significantly. The  $\Delta l$ ,  $\Delta \varphi$  and adjusted  $q$  were collected as the training data set of ANFIS.

### 3.2.3. Training of Data Set

Training ANFIS means using the optimization algorithm to optimize the premise and the consequence parameters in its structure. The premise parameters were  $\mu_{A_i}(\Delta l)$  or  $\mu_{B_i}(\Delta \varphi)$  ( $i = 1, 2 \dots, 8$ ) in the second membership function, and the consequence parameters were  $p_i, q_i, s_i$  ( $i = 1, 2 \dots, 64$ ) in the fourth membership function  $f_i$ . In MATLAB, ANFIS toolbox and graphic editing tools were used to train the data set.

### 3.3. Control System

A fuzzy system can be obtained through ANFIS training. As shown in Figure 5, the weight  $q$  can be adaptively adjusted by the fuzzy system according to the input of  $\Delta l$  and  $\Delta \varphi$ . Assuming  $q_\varphi$  is 1, Equation (30) can be solved according to Equation (29), and  $Q_i$  is used at Equation (21).

$$Q_i = \text{diag}(q, q, 1), (i = 1, 2, \dots, N_c, \dots, N_p) \tag{30}$$

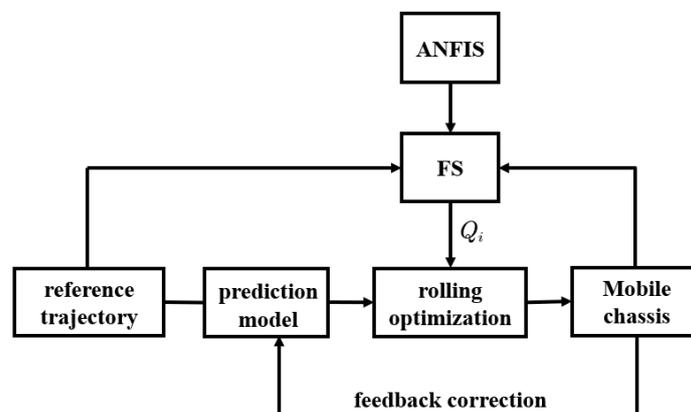


Figure 5. The control system schematic diagram.

### 4. Simulation and Analysis

#### 4.1. Experiment Parameters

##### 4.1.1. The Parameters of Omni-Directional Wheel Chassis

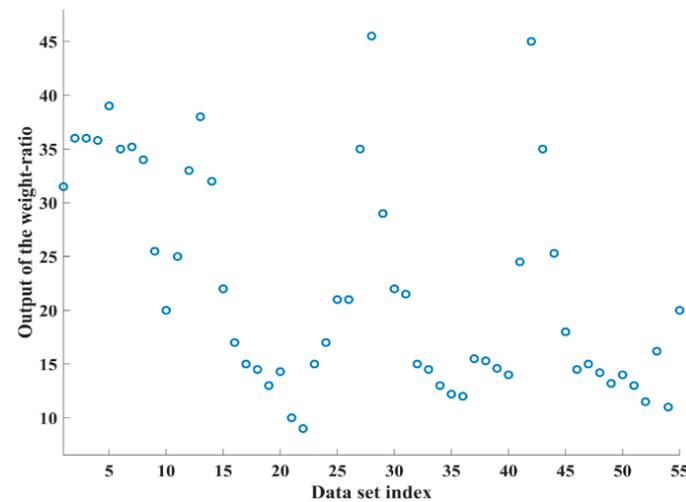
The distance  $l$  between every omni-directional wheel and the robot center point is 0.291 m, and the other parameters are shown in Table 1.

**Table 1.** Calculation parameters.

Wheel	$\alpha$	$\beta$	$\gamma$	$\theta$
1	45°	0°	0°	0°
2	135°	0°	0°	0°
3	-135°	0°	0°	0°
4	-45°	0°	0°	0°

##### 4.1.2. Training of Data Set

MPC was used to track the path of a circle. The sampling period  $T$  was 0.5 s. The data set for training ANFIS is shown in Figure 6. In SIMULINK, ANFIS toolbox and graphic editing tools were used to train the data set, and the parameters were set as shown in Table 2. The membership functions of using 7 and 8 fuzzy sets are shown in Table 3, the training error of using 7 fuzzy sets was 137.63, and the training error of using 8 fuzzy sets was 0.14. Accordingly, this paper used 8 fuzzy sets, and the membership functions after training ANFIS are shown in Figures 7 and 8.



**Figure 6.** Data set for training ANFIS controller.

**Table 2.** Part of the design parameters for the ANFIS toolbox in MATLAB.

Item	Method	Error Tolerance	Epochs
Parameter	Back-Propagation	0	150

**Table 3.** The membership functions of using 7 and 8 fuzzy sets.

Num of MFs	MFs
8	NB NM NS NZ PZ PS PM PB
7	NB NM NS Z PS PM PB

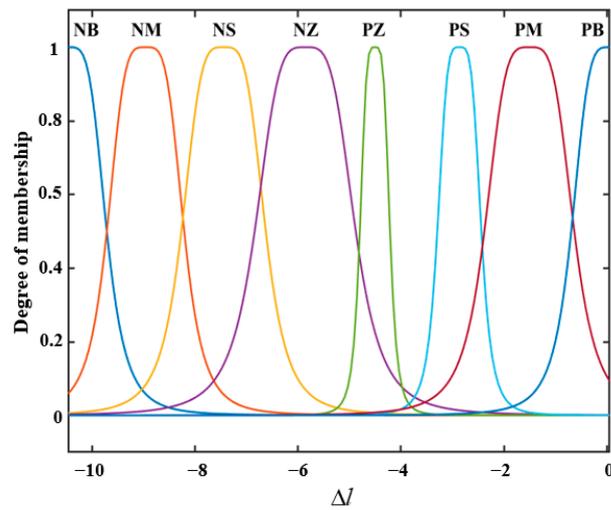


Figure 7. The membership functions of  $\Delta l$ .

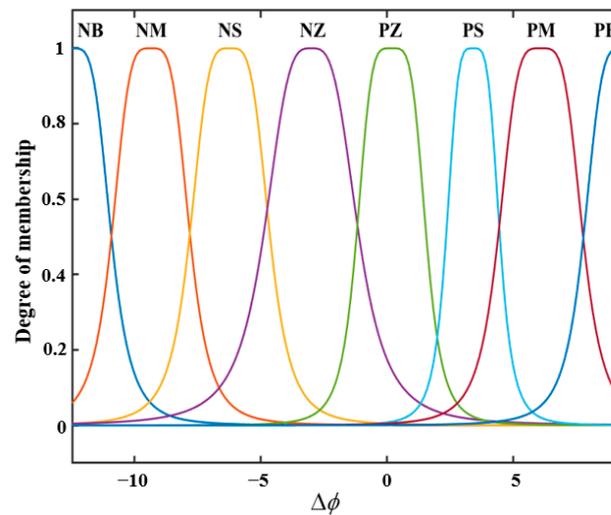


Figure 8. The membership functions of  $\Delta\phi$ .

#### 4.2. Simulation Results

MPC based on ANFIS was adopted to simulate and analyze two different paths. The simulation time  $t$  was 30 s, and the sampling period  $T$  was 0.5 s. The speed, acceleration, and relaxation factor of every motor of the omni-directional wheel chassis were:

$$\begin{cases} [u_{\min} \quad u_{\max}] = [-7 \quad 7] \\ [\Delta u_{\min} \quad \Delta u_{\max}] = [-3 \quad 3] \\ [\varepsilon_{\min} \quad \varepsilon_{\max}] = [0 \quad 10] \end{cases} \quad (31)$$

when the error curve fluctuated uniformly, the error peak value was the maximum fluctuation difference. The optimization ratio of reducing the maximum fluctuation difference can be solved by Equation (32).

$$P_{or} = \frac{e_t - e_o}{e_t} \quad (32)$$

where  $P_{or}$  is the optimization ratio,  $e_t$  is the maximum fluctuation difference of a traditional MPC, and  $e_o$  is the maximum fluctuation difference of MPC based on ANFIS.

4.2.1. Path I

The reference trajectory of a circle was Equation (33), the initial status  $\zeta_{start}$  was  $[0, 0, 0]^T$ , and the range of weight ratio  $q$  was  $q \in ( 13 \ 50 )$ . The simulation comparison was conducted between a traditional MPC (the weight ratio  $q$  is 25) and MPC based on ANFIS.

$$\begin{cases} X = 2.5 * \sin(\frac{2 * \pi * t}{30}) \\ Y = 3.5 - 2.5 * \cos(\frac{2 * \pi * t}{30}) \\ \varphi = 0.5 \end{cases} \quad (33)$$

As shown in Figures 9–14, after stabilization, the results are shown in Tables 4 and 5.

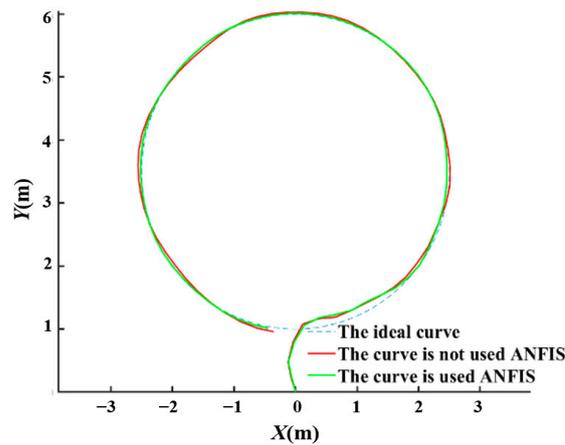


Figure 9. Trajectory comparison without disturbance.

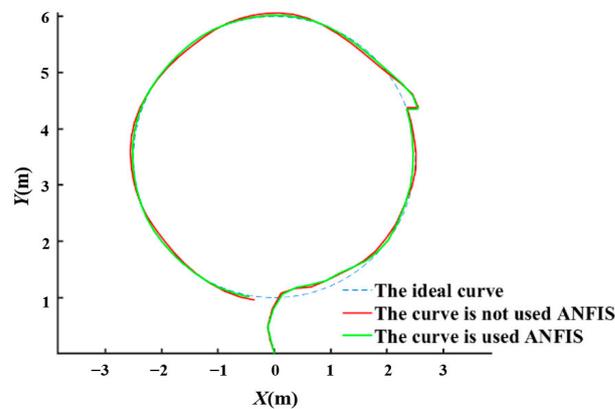


Figure 10. Trajectory comparison with disturbance.

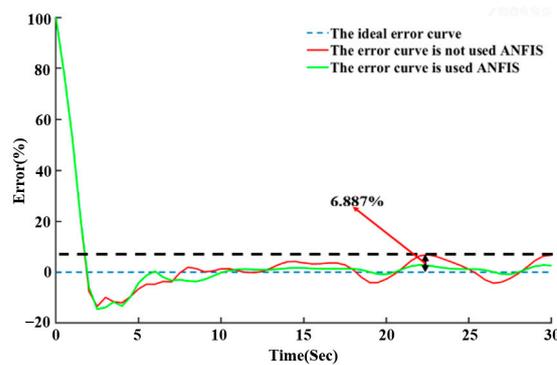


Figure 11.  $\Delta l$  without disturbance.

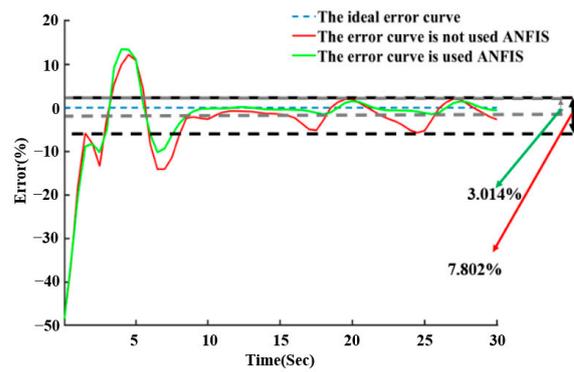


Figure 12.  $\Delta\varphi$  without disturbance.

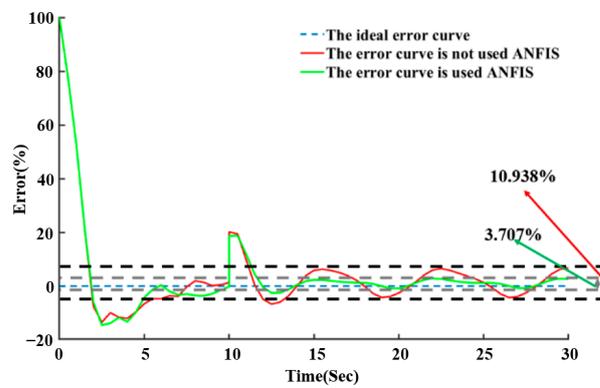


Figure 13.  $\Delta l$  with disturbance.

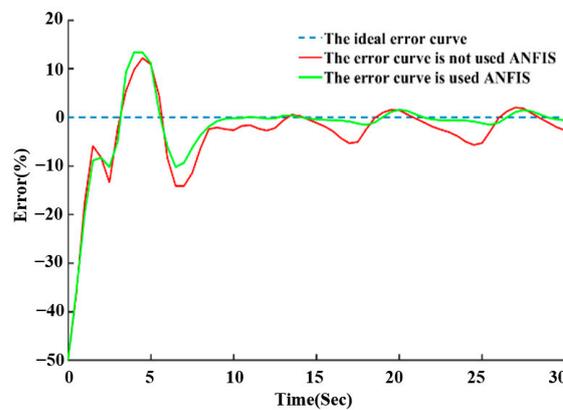


Figure 14.  $\Delta\varphi$  with disturbance.

Table 4. Maximum fluctuation difference without disturbance.

	$\Delta l$	$\Delta\varphi$
traditional MPC	6.887%	5.737%
MPC based on ANFIS	2.826%	1.555%

Table 5. The peak difference between the up and down fluctuations without disturbance.

	$\Delta l$	$\Delta\varphi$
traditional MPC	11.199%	7.802%
MPC based on ANFIS	3.6357%	3.014%

Compared with the traditional MPC without disturbance, MPC based on ANFIS reduced the maximum fluctuation difference of  $\Delta l$  by 58.966%, and the maximum fluctuation difference of  $\Delta\varphi$  was reduced by 72.895%. The stability of  $\Delta l$  increased by 67.668% and the stability of  $\Delta\varphi$  was increased by 61.369%. All the data showed that the error curve of the proposed algorithm was flatter and more stable compared with the traditional MPC without disturbance.

The external disturbance applied to the X direction at  $t$  was 10 s, and the disturbance was 0.2 m. After stabilization, the results are shown in Tables 6 and 7.

**Table 6.** Maximum fluctuation difference with disturbance.

	$\Delta l$	$\Delta\varphi$
traditional MPC	6.712%	5.670%
MPC based on ANFIS	2.841%	1.569%

**Table 7.** The peak difference between the up and down fluctuations with disturbance.

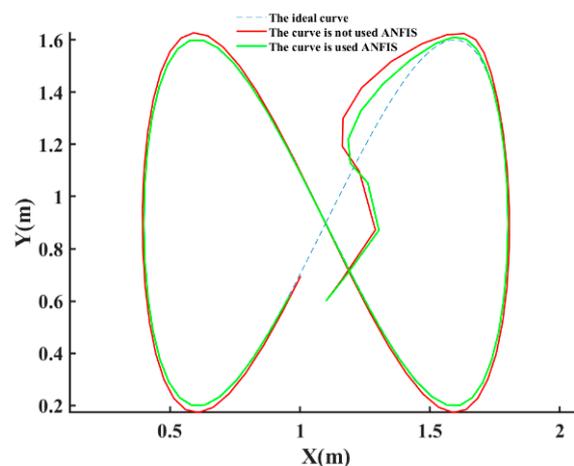
	$\Delta l$	$\Delta\varphi$
traditional MPC	10.938%	7.693%
MPC based on ANFIS	3.707%	3.085%

Compared with the traditional MPC, MPC based on ANFIS reduced the maximum fluctuation difference of  $\Delta l$  by 57.673%, and the maximum fluctuation difference of  $\Delta\varphi$  was reduced by 72.328%. The stability of  $\Delta l$  was increased by 66.109%, and the stability of  $\Delta\varphi$  increased by 59.899%. The error curve was much flatter and stabilized faster compared to the traditional MPC with disturbance. The experiment showed the effectiveness and superiority of the MPC based on ANFIS.

#### 4.2.2. Path II

The reference trajectory of  $\infty$  was Equation (34), the initial status  $\xi_{start}$  was  $[1.1, 0.6, 0]^T$ , and the range of weight ratio  $q$  was  $q \in [9, 15]$ . The weight ratio  $q$  of a traditional MPC was 15. The simulation results are shown in Figures 15–20.

$$\begin{cases} X = 1.1 + 0.7 \sin\left(\frac{2\pi t}{30}\right) \\ Y = 0.9 + 0.7 \sin\left(\frac{4\pi t}{30}\right) \\ \varphi = 0.5 \end{cases} \quad (34)$$



**Figure 15.** Trajectory comparison without disturbance.

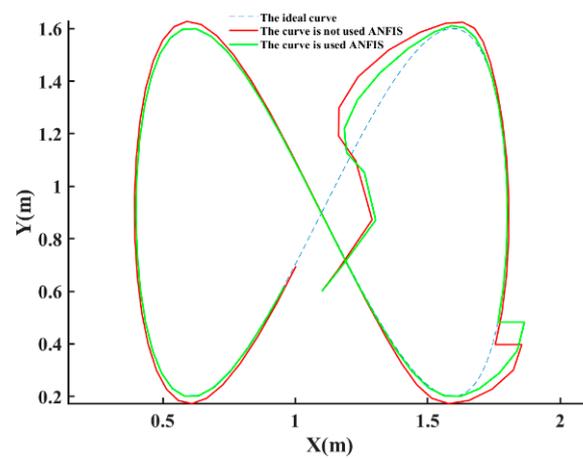


Figure 16. Trajectory comparison with disturbance.

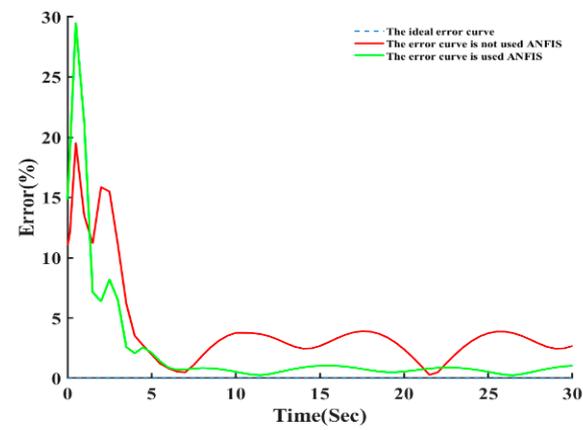


Figure 17.  $\Delta l$  without disturbance.

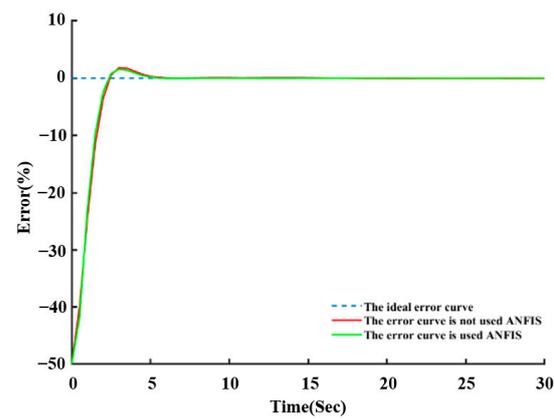


Figure 18.  $\Delta\phi$  without disturbance.

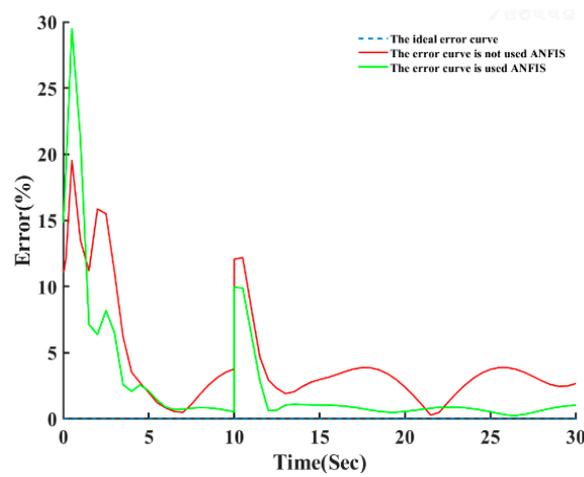


Figure 19.  $\Delta l$  with disturbance.

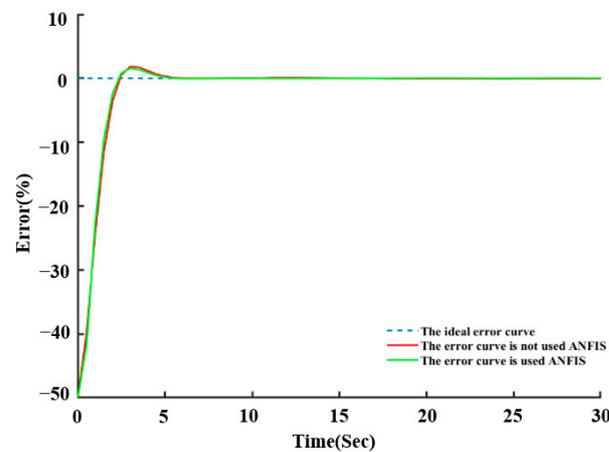


Figure 20.  $\Delta\varphi$  with disturbance.

As shown in Figures 15–20, after stabilization, the results are shown in Tables 8 and 9.

Table 8. Maximum fluctuation difference without disturbance.

	$\Delta l$	$\Delta\varphi$
traditional MPC	3.869%	$\approx 0$
MPC based on ANFIS	1.045%	$\approx 0$

Table 9. The peak difference between the up and down fluctuations without disturbance.

	$\Delta l$	$\Delta\varphi$
traditional MPC	3.5915%	–
MPC based on ANFIS	0.798%	–

Compared with the traditional MPC without disturbance, MPC based on ANFIS reduced the maximum fluctuation difference of  $\Delta l$  by 72.990%, and all the  $\Delta\varphi$  approached 0. The stability of  $\Delta l$  was increased by 77.780%. The error curve tended to be flatter and more stable compared to the traditional MPC without disturbance.

An external disturbance applied to the X direction at  $t$  was 10 s, and the disturbance was 0.1 m. After stabilization, the results are shown in Tables 10 and 11.

**Table 10.** Maximum fluctuation difference with disturbance.

	$\Delta l$	$\Delta\varphi$
traditional MPC	3.869%	$\approx 0$
MPC based on ANFIS	1.032%	$\approx 0$

**Table 11.** The peak difference between the up and down fluctuations with disturbance.

	$\Delta l$	$\Delta\varphi$
traditional MPC	3.5923%	-
MPC based on ANFIS	0.785%	-

Compared with the traditional MPC with disturbance, MPC based on ANFIS reduced the maximum fluctuation difference of  $\Delta l$  by 73.326%, and all the  $\Delta\varphi$  approached 0. The stability of  $\Delta l$  was increased by 78.148%. The error curve was much flatter and stabilized faster compared with the traditional MPC with disturbance. Moreover, the robot could respond more quickly when there was disturbance.

Experiments showed that MPC based on ANFIS had superior performance compared with a traditional MPC.

## 5. Conclusions

By adjusting the weight of the cost function in MPC, the path tracking error can be reduced prominently. However, the weight of the cost function in a traditional MPC needs to be manually adjusted which is time-consuming and labor-intensive, and it is difficult to adjust to a satisfactory value. Therefore, in order to adaptively adjust the weight, MPC based on ANFIS is proposed. The ANFIS toolbox is used for data set training, and the training error is 0.14. In SIMULINK, MPC based on ANFIS and a traditional MPC are compared to verify the effectiveness and superiority. Experiments show that when the path is circular and  $\infty$ , compared with a traditional MPC, the maximum fluctuation difference effect of  $\Delta l$  is reduced by around 50%, the maximum fluctuation difference effect of  $\Delta\varphi$  is reduced by around 70% (all the  $\Delta\varphi$  approach 0 when the path is  $\infty$ ), and the stability is increased by around 60%. The error curve is much flatter and tends to stabilize faster compared with a traditional MPC. All the experiments show the effectiveness and superiority of the MPC based on ANFIS.

**Author Contributions:** F.P., M.L. and Z.T. developed the methodology and drafted the manuscript; F.P. performed the numerical simulations and experiments; F.P. and X.X. revised the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Jiangsu Key R&D Program (Grant Nos. BE2018004-1, BE2018004) and the Fundamental Research Funds for the Central Universities (Grant No. B200202221).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** This paper is no data.

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

## References

1. Qian, J.; Zi, B.; Wang, D.; Ma, Y.; Zhang, D. The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system. *Sensors* **2017**, *17*, 2073. [[CrossRef](#)] [[PubMed](#)]
2. Normey-Rico, J.E.; Alcalá, I.; Gómez-Ortega, J.; Camacho, E.F. Mobile robot path tracking using a robust PID controller. *Control Eng. Pract.* **2001**, *9*, 1209–1214. [[CrossRef](#)]
3. Kuo, C.H. Trajectory and heading tracking of a mecanum wheeled robot using fuzzy logic control. In Proceedings of the 2016 International Conference on Instrumentation, Control, and Automation, Bandung, Indonesia, 29–31 August 2016.

4. Chao, C.-T.; Sutarna, N.; Chiou, J.-S.; Wang, C.-J. Equivalence between fuzzy PID controllers and conventional PID controllers. *Appl. Sci.* **2017**, *7*, 513. [[CrossRef](#)]
5. Malayjerdi, E.; Kalani, H.; Malayjerdi, M. Self-Tuning Fuzzy PID Control of a Four-Mecanum Wheel Omni-directional Mobile Platform. In Proceedings of the Iranian Conference on Electrical Engineering (ICEE), Mashhad, Iran, 8–10 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 816–820.
6. Hossein, F.M.; Khalil, A.; Bahram, T. Tracking-Error Fuzzy-Based Control for Nonholonomic Wheeled Robots. *Arab. J. Sci. Eng.* **2019**, *44*, 881–892.
7. Camacho, E.F.; Alba, C.B. *Model Predictive Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
8. Kuhne, F.; Lages, W.F.; Da Silva, J.G., Jr. Model predictive control of a mobile robot using linearization. In Proceedings of the Mechatronics and Robotics, Aachen, Germany, 13–15 September 2004; pp. 525–530.
9. Caireta, I.M. *Model Predictive Control for a Mecanum-Wheeled Robot in Dynamical Environments*; Universitat Politècnica de Catalunya Facultat de Matemàtiques i Estadística, Universitat Politècnica de Barcelona: Barcelona, Spain, 2019.
10. Fuad, A.; Mahmood, I.; Ahmad, S.; Norsahperi, N.; Toha, S.; Akmeliawati, R.; Darsivan, F. Modeling and simulation for heavy-duty mecanum wheel platform using model predictive control. *IOP Conf. Ser. Mater. Sci. Eng.* **2017**, *184*, 012050. [[CrossRef](#)]
11. Han, Y.; Zhu, Q. Robust Optimal Control of Omni-directional Mobile Robot using Model Predictive Control Method. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 4679–4684.
12. 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](#)]
13. Sinaeefar, Z.; Farrokhi, M. Adaptive Fuzzy Model-Based Predictive Control of Nonholonomic Wheeled Mobile Robots including Actuator Dynamics. *Int. J. Sci. Eng. Res.* **2012**, *3*, 305–311.
14. Wang, H.; Liu, B.; Ping, X.; An, Q. Path Tracking Control for Autonomous Vehicles Based on an Improved MPC. *IEEE Access* **2019**, *7*, 161064–161073. [[CrossRef](#)]
15. Jang, J.S. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [[CrossRef](#)]
16. Essien, E.; Ibrahim, H.; Mehrazdeh, M.; Idem, R. Adaptive neuro-fuzzy inference system (ANFIS)—Based model predictive control (MPC) for carbon dioxide reforming of methane (CDRM) in a plug flow tubular reactor for hydrogen production. *Therm. Sci. Eng. Prog.* **2019**, *9*, 148–161. [[CrossRef](#)]
17. Morales, S.; Magallanes, J.; Delgado, C.; Canahuire, R. LQR Trajectory Tracking Control of an Omnidirectional Wheeled Mobile Robot. In Proceedings of the 2018 IEEE 2nd Colombian Conference on Robotics and Automation, Barranquilla, Colombia, 1–3 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5.
18. Shimada, A.; Yajima, S.; Viboonchaicheep, P.; Samura, K. Mecanum-wheel vehicle systems based on position corrective control. In Proceedings of the 31st Annual Conference of IEEE Industrial Electronics Society (IECON 2005), Raleigh, NC, USA, 6–10 November 2005; IEEE: Piscataway, NJ, USA, 2005; p. 6.
19. De Tlale, N.; Villiers, M. Kinematics and dynamics modelling of a mecanum wheeled mobile platform. In Proceedings of the 2008 15th International Conference on Mechatronics and Machine Vision in Practice, Auckland, New Zealand, 2–4 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 657–662.
20. Mishra, S.; Sharma, M.; Mohan, S. Behavioural Fault tolerant control of an Omni directional Mobile Robot with Four mecanum Wheels. *Def. Sci. J.* **2019**, *69*, 353. [[CrossRef](#)]
21. Siegwart, R.; Nourbakhsh, I.R.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots*; MIT Press: Cambridge, MA, USA, 2011.
22. Kurdi, M.; Elzein, I.A.; Dadykin, A.K. Model Predictive Control for Positioning and Navigation of Mobile Robot with Cooperation of UAV. *Commun. Appl. Electron.* **2017**, *6*, 17–25.
23. Liang, H.; Li, G.; Rong, T.; Deng, L.; Amp, A. Unmanned vehicle trajectory tracking based on model predictive control. *Automob. Appl. Technol.* **2017**, *20*, 53–55.
24. Klančar, G.; Škrjanc, I. Tracking-error model-based predictive control for mobile robots in real time. *Robot. Auton. Syst.* **2007**, *55*, 460–469. [[CrossRef](#)]
25. Li, S.; Wang, J.; Li, K. Stabilization of linear predictive control systems with softening constraints. *J. TsingHua Univ. Sci Technol.* **2010**, *10*, 94–98.
26. Karaboga, D.; Kaya, E. Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey. *Artif. Intell. Rev.* **2019**, *52*, 2263–2293. [[CrossRef](#)]
27. Subbash, P.; Chong, K.T. Adaptive network fuzzy inference system based navigation controller for mobile robot. *Front. Inf. Technol. Electron. Eng.* **2019**, *20*, 141–151. [[CrossRef](#)]
28. Eesaño, J.M.; Bordons, C.; Vilas, C.; García, M.R.; Alonso, A. Neurofuzzy model based predictive control for thermal batch processes. *J. Process. Control.* **2009**, *19*, 1566–1575. [[CrossRef](#)]
29. Bakircioğlu, V.; Şen, M.A.; Kalyoncu, M. Adaptive Neural-Network Based Fuzzy Logic (ANFIS) Based Trajectory Controller Design for One Leg of a Quadruped Robot. In Proceedings of the 5th International Conference on Mechatronics and Control Engineering, Venice, Italy, 14–17 December 2016; pp. 82–85.