

基于模糊免疫神经网络 PID 算法的全向底盘控制方法

孙浩水, 王小平[✉], 王晓光, 林秦颖

(空军工程大学航空工程学院, 西安, 710038)

摘要 针对全向底盘控制的实际需求,提出了基于模糊免疫神经网络 PID 算法的智能控制方法。首先根据神经网络算法和模糊算法的结构特点建立了模糊神经网络模型,并使用误差反向传播的方法对模型进行训练;然后使用免疫算法确定学习率,实现了对 PID 参数的动态整定,并对底盘路径跟踪控制器参数进行整定,以实现对底盘的精确运动控制;最后建立了底盘的运动学模型,基于 Matlab 平台进行了相关算法的仿真,并基于 Linux Ubuntu 系统下 Tensorflow 框架搭建并训练了神经网络模型,进而实现了整体算法。轨迹跟踪试验表明:当底盘沿不同方向以 5 m/s 的速度进行轨迹跟踪时,最大误差为 4.88 cm,平均误差为 0.25 cm,该算法能够有效地对底盘进行控制,满足全向底盘控制的要求。

关键词 全向底盘;模糊神经网络;PID 算法;免疫算法

DOI 10.3969/j.issn.1009-3516.2018.04.011

中图分类号 TP249 **文献标志码** A **文章编号** 1009-3516(2018)04-0059-07

Control Method for Omni-Directional Chassis Based on Fuzzy Immune Neural Network PID Algorithm

SUN Haoshui, WANG Xiaoping[✉], WANG Xiaoguang, LIN Qinying

(Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China)

Abstract: Aiming at the actual demand of omni-directional chassis control, an intelligent control method based on fuzzy immune neural network PID algorithm is proposed to set parameters of the path tracking controller of the chassis to realize the precise motion control of the chassis. According to the structural characteristics of the neural network algorithm and the fuzzy algorithm, the fuzzy neural network model is established and the model is trained by the method of error back propagation. The immune algorithm is used to determine the learning rate and then, the dynamic tuning of the PID parameters is realized. After that, the kinematic model of the chassis is established. Then, based on Matlab platform, the relevant algorithm is simulated. Finally, the neural network model is established and trained via the Tensorflow structure under the Linux Ubuntu system and the algorithm is then implemented. When the chassis carries out trajectory tracking in different directions at a 5 m/s speed, the maximum error of the process is 4.88 cm, the average error is 0.25 cm. Simulations and experiments both show that the algorithm can effectively control the chassis to meet requirements of the control of the omni-directional chassis.

Key words: omni-directional chassis; fuzzy neural network; PID algorithm; immune algorithm

收稿日期: 2017-11-02

基金项目: 航空科学基金(20145196023)

作者简介: 孙浩水(1994—),男,江苏徐州人,硕士生,主要从事移动机器人研究. E-mail:sunhaoshui@163.com

通信作者: 王小平(1972—),男,江苏常州人,教授,主要从事控制算法与理论研究. E-mail:15829781618@yeah.net

引用格式: 孙浩水,王小平,王晓光,等. 基于模糊免疫神经网络 PID 算法的全向底盘控制方法[J]. 空军工程大学学报(自然科学版), 2018, 19(4): 59-65. SUN Haoshui, WANG Xiaoping, WANG Xiaoguang, et al. Control Method for Omni-Directional Chassis Based on Fuzzy Immune Neural Network PID Algorithm[J]. Journal of Air Force Engineering University (Natural Science Edition), 2018, 19(4): 59-65.

全向运动平台具有平面上的3个自由度,能够实现纵向平移、侧向平移和自身的旋转运动,有很大的灵活性,因而在狭小、复杂的空间中被大量使用^[1]。但是全向运动平台很容易受到不平整地面等外界因素的干扰,对于运动控制的精度要求很高^[2]。为了实现更好的控制效果,在硬件传感器方面,多传感器融合的方式成为主流。现阶段较为常用的本体感知类的导航传感器有激光雷达、摄像头、里程计、IMU(惯性测量单元)等,其中里程计和IMU相融合的方案已经较为成熟。在工程控制算法方面,PID算法具有简单、可靠、易于工程实现等优点。但是,对于非线性系统来说,常规的PID算法的稳定性、动态性等性能并没有得到完整的数学证明^[3];对于宽输入系统,也无法在全输入范围内进行很好的跟踪。模糊自适应PID算法^[4-5]、遗传PID算法^[6-7]、神经网络PID算法^[8-9]等等,都在解决这一问题上取得了不错的效果。但由于计算量及部署条件等原因,这些算法中的绝大部分并没有被应用于小型低成本移动平台的控制中。近几年,随着处理器的飞速发展,x86平台的微型化,以及GPU的普及,在小型移动平台上运行大计算量算法已成为可能。同时,随着深度神经网络框架重新崛起,TensorFlow、Caffe、MXNet、Torch等深度学习框架的快速发展,深度神经网络算法的部署也越来越方便^[10]。

从成本角度出发,本文选择了里程计和IMU相融合的传感器方案来实现全向底盘的控制。同时,为了更好地应对不平整路面环境等外界干扰因素,底盘加装了悬挂减震系统以更好地适应外界复杂环境。

1 全向底盘运动模型

目前主流的全向运动底盘主要采用麦克纳姆轮。麦克纳姆轮由瑞典麦克纳姆公司发明,轮缘上分布着许多小滚子,可以实现横向移动,故而大量运用在全向运动平台上。麦克纳姆轮主要有45°滚子和90°滚子,其中45°滚子的麦克纳姆轮有较强的负载能力而被广泛运用于工业实践中^[11]。

如图1所示,分别建立世界坐标系 xoy 与车身坐标系 $x'o'y'$ 坐标系。设4个轮子的角速度分别为 $\omega_1, \omega_2, \omega_3, \omega_4$,麦克纳姆轮半径为 R ,车身坐标系 $o'x'$ 轴与世界坐标系 oy 轴夹角为 θ ,底盘长为 $2L_1$,宽为 $2L_2$,滚子接触地面时与 x 轴夹角为 $\alpha, \alpha=45^\circ$ 。

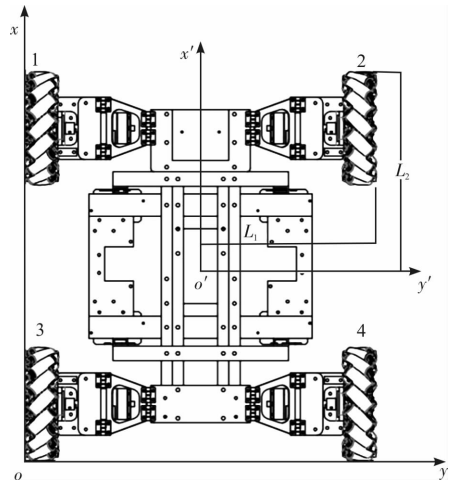


图1 底盘运动学模型示意图

Fig. 1 The kinematical model of the chassis

建立底盘方程,得:

$$y_k = g(u_k, n_k) \quad (1)$$

$$x_k = f(x_{k-1}, y_k, q_k) \quad (2)$$

式中: x_k 为底盘运动的状态量; y_k 为底盘运动的输出量; u_k 为底盘的控制量; f 为底盘的运动方程; g 为底盘的控制方程; n_k, q_k 为系统噪声;

以底盘运动起始点为原点,在二维平面内建立直角坐标系。首先考虑底盘不发生转动运动时的情况。

分析控制方程,根据实际控制经验,一般取 $|\omega_1| = |\omega_2|, |\omega_3| = |\omega_4|$ ^[12],故选取 $u_k = [\omega_1, \omega_2]^T, y_k = [v_k, \theta_k]^T$,其中 v_k 为底盘中心点 o 的运动速率, θ_k 为底盘中心点 o 的前进方向与 x 轴夹角。可得:

$$\begin{bmatrix} v_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \sqrt{\omega_1^2 + \omega_2^2} \\ \arctan\left(\frac{|\omega_1| + |\omega_2|}{|\omega_1| - |\omega_2|}\right) \end{bmatrix} + n_k \quad (3)$$

分析方程(3)可得,给定控制量 ω_1 和 ω_2 ,底盘即可在没有转动的情况下朝任意方向前进,实现全向运动。

分析全向底盘的状态方程,得:

$$x_{k+1} = x_k + \Delta x_k + q_k \quad (4)$$

选取 $x_k = [a_k, b_k]^T$,其中 a_k, b_k 为全向底盘在坐标系内的坐标值,可得:

$$\begin{bmatrix} a_k \\ b_k \end{bmatrix} = \begin{bmatrix} a_{k-1} + v_k \sin\theta_k \\ b_{k-1} + v_k \cos\theta_k \end{bmatrix} + q_k \quad (5)$$

再考虑底盘纯转动的情况。将底盘看成不发生形变的刚体,在其进行纯转动的情况时,必有 $\omega_1 = -\omega_2$ 成立。此时选取 $u_k = \gamma$,其中 γ 为底盘旋转的角速度,规定正方向为逆时针方向,得控制方程为:

$$|\gamma| = \frac{|\omega_1| R}{L_1 + L_2} \quad (6)$$

其状态方程同式(5)。

至此,全向底盘运动方程与控制方程全部建立完毕。所有全向运动过程都可视为不发生转动运动和纯转动运动的合成。

2 模糊免疫神经网络 PID 控制器

2.1 免疫算法

人体的简化免疫过程可以抽象得出 2 个反馈回路:①抗原激活相应免疫细胞,免疫细胞消灭抗原;②T 抑制细胞被激活,并抑制其它所有免疫细胞^[13]。基于上述过程,我们做出如下假设与简化^[14]:

1)由于 T 抑制细胞的作用过程较为复杂,假定其只对 B 细胞进行抑制;

2)假定 B 细胞直接产生抗体消灭抗原,忽略 T 杀伤细胞的过程。

定义第 k 代抗原数量为 $\epsilon(k)$,则此时 T 辅助细胞浓度 $T_{\text{help}}(k)$ 和 T 抑制细胞浓度 $T_{\text{sup}}(k)$ 分别为:

$$T_{\text{help}}(k) = k_1 \epsilon(k) \quad (7)$$

$$T_{\text{sup}}(k) = k_2 f(\Delta S(k)) \epsilon(k) \quad (8)$$

式中: k_1 为激励因子,符号为正; k_2 为抑制因子,符号为负; $f(\cdot)$ 为 T 抑制细胞的抑制量函数,表示细胞抑制能力的大小,是一个非线性函数。 $\Delta S(k)$ 为 B 细胞收到总刺激 $S(k)$ 的变化量。其中 $S(k)$ 和 $\Delta S(k)$ 分别定义为:

$$S(k) = T_{\text{help}}(k) - T_{\text{sup}}(k) \quad (9)$$

$$\Delta S(k) = S(k) - S(k-1) \quad (10)$$

将式(7)~(8),(10)都代入到式(9)中,得:

$$S(k) = (k_1 - k_2 f(\Delta S(k))) \epsilon(k) =$$

$$k_1 (1 - K f(\Delta S(k))) \epsilon(k) \quad (11)$$

式(11)即为免疫算法公式,其中 $K = \frac{k_2}{k_1}$ 。

2.2 模糊免疫神经网络 PID 算法

模糊神经网络算法结合了模糊算法表达规则的能力和神经网络的自主学习能力,在无法得到被控对象的模型时也可以很好地进行控制。同时,在应对非线性、多输入、强干扰的系统时能表现出较强的适应性。同时,引入免疫算法,实现神经网络学习率的整定,使系统具有更好的动态性。

2.2.1 模糊神经网络 PID 控制器

控制器的整体框图见图 2。将系统的输出误差

$e(t)$ 和误差变化量 $\Delta e(t)$ 输入模糊神经网络控制器中,实现对比例系数 K_p 、积分系数 K_i 、微分系数 K_d 的整定。

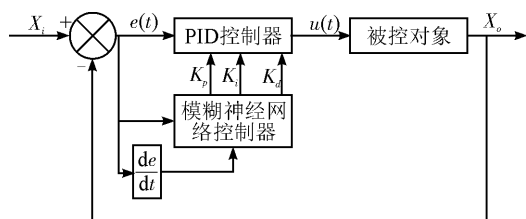


图 2 模糊神经网络 PID 算法流程图

Fig. 2 The flow chart of the fuzzy neural network

由模糊理论可知,模糊算法可分为模糊化、模糊推理、逆模糊化 3 个步骤^[15-16]。做如下定义:

1) $u_i^{(k)}$ 为第 k 层第 i 个节点输入信号, $\omega_{ij}^{(k)}$ 为第 k 层第 i 个节点第 j 个输出连接权值, $o_i^{(k)}$ 为第 k 层第 i 个节点输出信号, $f_i^{(k)}$ 为第 k 层第 i 个节点的激活函数输出。

2) 网络第 1 层为输入层,将系统的输出误差 $e(t)$ 和误差变化量 $\Delta e(t)$ 送入系统, $f_i^{(1)} = u_i^{(1)}$ 。该层所有节点连接权值为 1。

3) 第 2 层为模糊化层,由隶属度函数对输入量进行模糊化处理。隶属度函数选择为高斯分布模型函数,则 $f_{ij}^{(2)} = \exp(-\frac{(u_i^{(2)} - m_{ij})^2}{\sigma_{ij}^2})$ 。其中 m_{ij} 为第 i 个模糊变量第 k 项高斯函数中心, σ_{ij} 为该高斯函数宽度。该层所有节点连接权值为 1。

4) 第 3、4 层为模糊计算层,完成模糊计算, $f_i^{(3)} = \min(u_i^{(3)}, u_2^{(3)}, \dots, u_p^{(3)})$, $f_i^{(4)} = \frac{o_i^{(3)}}{\sum o_i^{(3)}}$,所有节点链接权值为 1。

5) 第 5 层为输出层,将结果逆模糊化并输出, $f_i^{(5)} = \sum \omega_i^{(5)} o_i^{(4)}$,该层的连接权值为 $\omega_{ij}^{(5)}$ 。

2.2.2 基于误差逆传播的学习方法

控制器的整体框图见图 3。将系统的输出误差 $e(t)$ 和误差变化量 $\Delta e(t)$ 作为输入送入模糊神经网络控制器中以实现 K_p 、 K_i 、 K_d 的整定。

神经网络模型建立完毕后,需要给定误差函数使得网络进行学习。误差函数定义如下:

$$E_k = \frac{1}{2} \sum_{i=1}^p (X_o - X_i)^2 = \frac{1}{2} \sum_{i=1}^p e(t)^2 \quad (12)$$

式中: X_o 和 X_i 分别为系统实际和给定的输出。从前文可以得到,系统中可调节的参数为 m_{ij} , σ_{ij} , $\omega_{ij}^{(5)}$, 本文采用误差反向传播(BP)算法对其进行学习。对输出 X_o 进行偏导,得:

$$\frac{\partial e_k}{\partial X_o} = \frac{\partial E_k}{\partial f_i^{(5)}} = (X_o - X_i) \quad (13)$$

进而求得:

$$\frac{\partial E_k}{\partial \omega_{ij}^{(5)}} = \frac{\partial E_k}{\partial f_i^{(5)}} \frac{\partial f_i^{(5)}}{\partial \omega_{ij}^{(5)}} = (X_o - X_i) o_i^{(4)} \quad (14)$$

同时可以求得:

$$\frac{\partial E_k}{\partial f_i^{(4)}} = \sum_{n=1} \frac{\partial E_k}{\partial f_n^{(4)}} \frac{\partial f_n^{(5)}}{\partial f_i^{(4)}} \quad (15)$$

$$\frac{\partial E_k}{\partial f_i^{(3)}} = \sum_{n=1} \frac{\partial E_k}{\partial f_n^{(4)}} \frac{\partial f_n^{(4)}}{\partial f_i^{(3)}} \quad (16)$$

$$\frac{\partial E_k}{\partial f_{ij}^{(2)}} = \sum_{n=1} \frac{\partial E_k}{\partial f_n^{(2)}} \frac{\partial f_n^{(3)}}{\partial f_{ij}^{(2)}} \quad (17)$$

根据式(15)~(17),可求得相应的梯度:

$$\frac{\partial E_k}{\partial m_{ij}} = \frac{\partial E_k}{\partial f_{ij}^{(2)}} \frac{\partial f_{ij}^{(2)}}{\partial m_{ij}} = \frac{\partial E_k}{\partial f_{ij}^{(2)}} \frac{2(o_i^{(2)} - m_{ij})}{\sigma_{ij}^2} \quad (18)$$

$$\frac{\partial E_k}{\partial \sigma_{ij}} = \frac{\partial E_k}{\partial f_{ij}^{(2)}} \frac{\partial f_{ij}^{(2)}}{\partial \sigma_{ij}} = \frac{\partial E_k}{\partial f_{ij}^{(2)}} \frac{2(o_i^{(2)} - m_{ij})^2}{\sigma_{ij}^3} \quad (19)$$

BP算法采用梯度下降的策略,以目标的负梯度方向对参数进行学习^[17]。记学习率为 η ,学习算法公式为:

$$\omega_{ij}(k+1) = \omega_{ij}(k) - \eta \frac{\partial E_k}{\partial \omega_{ij}} \quad (20)$$

$$m_{ij}(k+1) = m_{ij}(k) - \eta \frac{\partial E_k}{\partial m_{ij}} \quad (21)$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \eta \frac{\partial E_k}{\partial \sigma_{ij}} \quad (22)$$

BP算法通过式(20)~(22)进行参数学习。初始设置误差函数的阈值为 E_{k0} 。当式(12)值小于 E_{k0} 时,算法停止,参数学习完成。

在BP算法运行过程中,学习率 η 的值直接影响算法效果。 η 太小,算法收敛速度很慢; η 太大,算法可能无法取到最小值甚至出现震荡^[18]。为了较快较好的实现收敛,我们使用免疫算法进行 η 的自整定。

令 $\Delta S(k)$ 分别为 $\Delta \omega_{ij}$, Δm_{ij} , $\Delta \sigma_{ij}$,非线性函数 $f(x) = x^2$,代入式(11)分别计算出 η ,代入式(20~22)中,最终得:

$$\omega_{ij}(k+1) = \omega_{ij}(k) - k_1(1 - K(\Delta \omega_{ij})^2) \frac{\partial E_k}{\partial \omega_{ij}} \quad (23)$$

$$m_{ij}(k+1) = m_{ij}(k) - k_1(1 - K(\Delta m_{ij})^2) \frac{\partial E_k}{\partial m_{ij}} \quad (24)$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - k_1(1 - K(\Delta \sigma_{ij})^2) \frac{\partial E_k}{\partial \sigma_{ij}} \quad (25)$$

3 底盘路径跟踪控制器

3.1 运动控制系统控制器

全向底盘由4个有刷直流电机驱动,为了实现对其运动状态的精确控制,需要运动控制模块对电机进行闭环控制。电机是一种强耦合、宽输入的非

线性系统,单纯使用PID算法无法在全输入范围内达到良好的控制效果。故本文将之前提出的算法运用于电机控制。

设底盘以速度 v 直线前进,由式(4)可求得电机转速 ω ,将其作为给定值,与电机编码器测得的实际值 ω_0 代入PID控制器:

$$\Delta u(k) = u(k) - u(k-1) =$$

$$K_p \Delta e(k) + K_i e(k) + K_d [\Delta e(k) - \Delta e(k-1)] \quad (26)$$

式中: $u(k)$ 为输入驱动器的控制量,即电压值; $e(k) = \omega - \omega_0$; $\Delta e(k) = e(k) - e(k-1)$;

算法运行过程中,实时计算 $e(k)$ 和 $\Delta e(k)$,送入模糊免疫神经网络PID控制器中,实现对3个系数进行周期性的学习和更新,进而实现在全输入范围内运动控制系统的动态稳定。

3.2 惯性导航系统控制器

仅依靠运动控制系统并不能保证全向底盘在所有情况下都能对目标轨迹进行很好地跟踪。在强干扰、复杂的地形下,还需要惯性导航系统的配合^[19]。

如图3,全向底盘对弧线轨迹进行追踪,实线箭头为解算出的运动方向,即弧线的切线方向;虚线箭头为底盘实际的运动方向,两者之间存在角度误差 δ ,这个误差可以通过IMU测量得到。将其输入式(26),得到电机转速 ω 输出。同样使用模糊免疫神经网络PID控制器对参数进行动态更新,见图4。

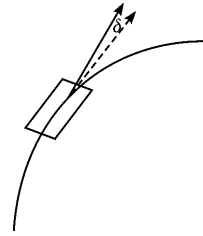


图3 轨迹跟踪图

Fig. 3 Track tracking diagram

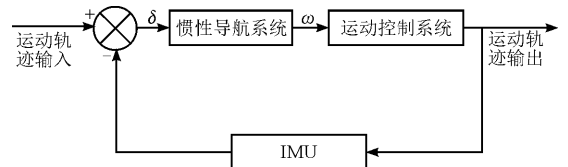


图4 底盘路径跟踪控制器流程图

Fig. 4 The flow chart of the chassis path tracking controller

4 仿真实验

4.1 Matlab 平台仿真

本文使用 Matlab 建模并对算法进行仿真。训

练前后 $e(t)$ 和 $\Delta e(t)$ 的隶属度函数选取为正态分布曲线,见图 5。经过 400 代训练, $e(t)$ 和 $\Delta e(t)$ 的隶属度函数见图 5(b)和图 5(c)所示。表中数据经过归一化处理。

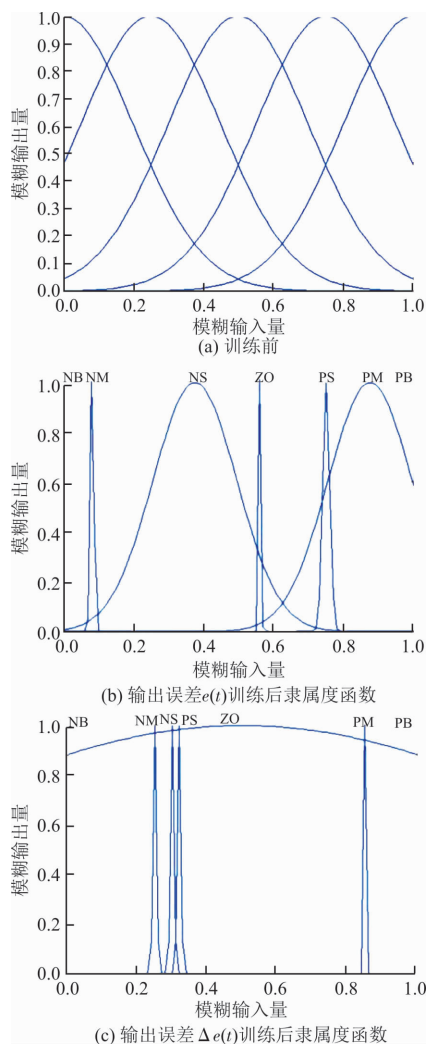


图 5 输出隶属度函数

Fig. 5 Membership functions of outputs

分析隶属度函数曲线变化情况,对于 $e(t)$,当其值处于模糊变量 NM、ZO、PS 顶点附近时,系统处于稳定阶段,可以提高系统的精度,故隶属度函数变化率较大;当处于其他模糊变量顶点附近时,系统稳定性成为需要考虑的首要因素,故隶属度函数斜率变小;对于 $\Delta e(t)$,其值在模糊变量 NM、NS、PS、PM 顶点附近时,系统比较稳定,故通过增大隶属度函数斜率来提高系统精度;当处于其他模糊变量顶点附近时,则需要减小隶属度函数斜率来提高稳定性。训练结果与实际情况十分吻合。

同时,基于 Simulink 平台分别搭建了运动控制系统和惯性导航系统的模型,分别使用模糊免疫神经网络 PID 算法、神经网络 PID 算法和常规 PID 算

法对其进行了阶跃响应实验。设定运动控制系统输入为 5 m/s 的阶跃信号,其输出结果见图 6。

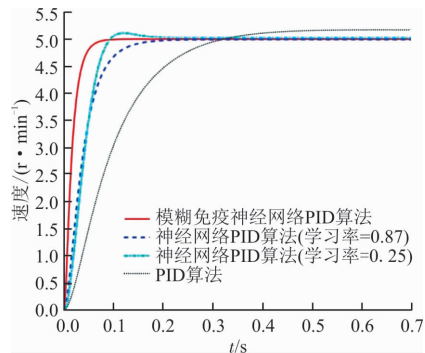


图 6 运动控制系统控制器响应曲线

Fig. 6 Response curves of the motion control system

从图 6 中可以得到,模糊免疫神经网络 PID 算法和神经网络 PID 算法都能得到较好的控制效果,但由于神经网络算法学习过程中学习率的选取对算法效果的影响很大,人工调参很难达到最好的控制效果;而常规的 PID 算法无论是在响应时间还是在稳态误差方面都存在较大差距。

设定惯性导航系统的输入为 10° 的阶跃信号,输出见图 7。同样可以看出模糊免疫神经网络 PID 算法的控制效果最佳。

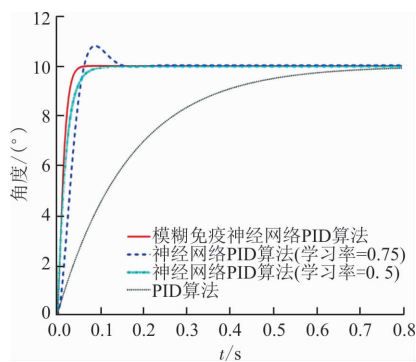


图 7 惯性导航系统控制器响应曲线

Fig. 7 Response curves of the inertial navigation system

4.2 轨迹跟踪实验

实验用全向底盘见图 8。其主体由碳纤维板雕刻加工而成,配备 4 个上海量能 P36H+SYK36 型行星减速电机和 4 个 152 mm 麦克纳姆轮。车体的底层控制由基于 ARM 构架芯片的下位机完成,而模糊免疫神经网络算法运行在基于 x86 构架的 Intel NUC5i7RYH 微型电脑上,两者通过串行接口进行通讯,波特率为 115 200,每隔 1 s 微型电脑对所有参数进行一次整定,每隔 5 ms 下位机向微型电脑上传一次所有运动状态数据。

通过基于 Linux Ubuntu 系统的 TensorFlow 深度学习框架,构建并训练了神经网络算法。经过约 400 代的学习,BP 算法的误差函数小于 1×10^{-6}

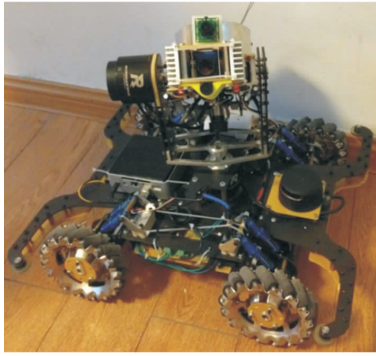


图8 实验底盘平台

Fig. 8 The experimental platform

的设定值,算法结束。整个学习过程耗时约 40 ms。

为了验证算法的效果,基于全向底盘进行了轨迹跟踪实验。在车身坐标系下,全向底盘以 5 m/s 的速度分别沿着 X 轴、Y 轴以及 $\theta=45^\circ$ 的方向沿直线前进 10 m。每个方向的轨迹跟踪实验进行 10 次,取偏差最大的一组,根据上传到笔记本电脑上的坐标信息,使用 matplotlib 库绘制出轨迹曲线。

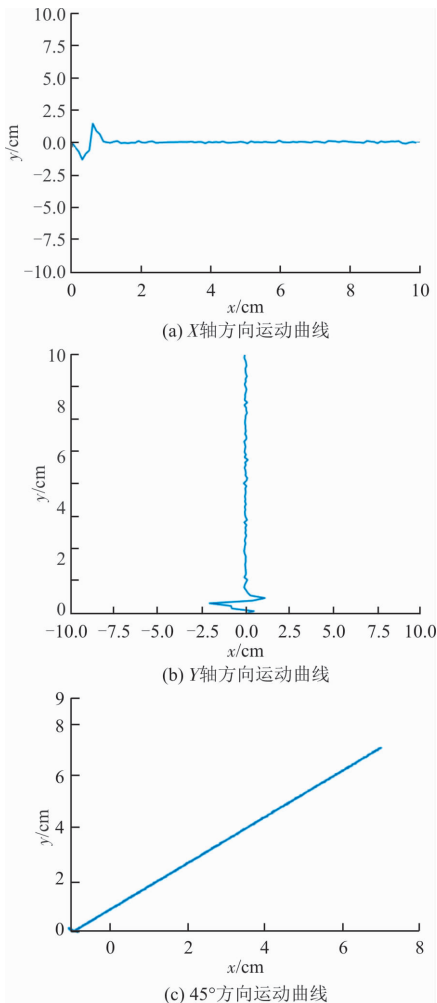


图9 路径跟踪实验曲线

Fig. 9 Experimental curves of the path planning

3 组实验的数据见表 1。

表 1 轨迹跟踪实验数据表

Tab. 1 The trajectory tracking experiment data sheet

实验	平均偏差/cm	最大偏差/cm	偏差均分差
X 轴方向	0.09	2.27	0.35
Y 轴方向	0.14	2.46	0.46
45°方向	0.53	4.88	0.97

从曲线和表中可以得出,在进行轨迹跟踪时,最大的偏差为 4.88 cm,发生在 $\beta=45^\circ$ 方向运动上。在轨迹跟踪过程中,平均误差为 0.25 cm,本文算法较好地实现了对轨迹的跟踪。

5 结语

本文提出了基于模糊免疫神经网络 PID 算法的全向底盘控制的方法。在分析全向底盘动力学模型的基础上,设计了包括运动控制系统和惯性导航系统的路径跟踪控制器并实现了对给定轨迹的跟踪。仿真和实验结果都表明,算法能较好地使底盘实现对轨迹的跟踪。在跟踪过程中,误差值在 0 左右波动,平均误差为 0.25 cm,最大误差为 4.88 cm,具有良好的控制效果,满足全向底盘控制的要求。后续还将会将该算法运用到无人机的控制中,实现算法更广泛的运用。

参考文献(References):

- [1] FENG Y, DING C, LI X, ZHAO X, et al. Integrating Mecanum Wheeled Omnidirectional Mobile Robots in ROS [C] //IEEE International Conference on Robotics and Biomimetics. Qingdao, China: IEEE, 2016.
- [2] YE C, JIANG X, YU S, JIANG C, et al. A Tracking Method of an Assembling Omnidirectional Mobile Robot [C] //IEEE International Conference on Robotics and Biomimetics. Qingdao: IEEE, 2016.
- [3] ZHAO C, GUO L. PID Controller Design for Second Order Nonlinear Uncertain Systems[J]. Science China Information Sciences, 2017, 60(2): 1-3.
- [4] 刘慧博, 王静, 吴彦合. 无刷直流电机模糊自适应 PID 控制研究与仿真[J]. 控制工程, 2014, 21(4): 583-587.
LIU H B, WANG J, WU Y H. Study and Simulation of Fuzzy Adaptive PID Control of Brushless DC Motor[J]. Control Engineer of China, 2014, 21(4): 583-587. (in Chinese)
- [5] 陈孝伟, 韩璞, 张悦. 一种变比例自适应 PID 控制方法的研究[J]. 计算机仿真, 2016, 33(6): 278-281.

- CHEN X W, WANG J, WU Y H. A Contril Method of Variable Proportion Adaptive PID[J]. Computer Simulation, 2016,33(6):278-281. (in Chinese)
- [6] 余婷. 基于改进遗传算法的船舶航向控制器设计[D]. 大连: 大连海事大学, 2015.
- YU T. A Design of Ship Autopilot on Improved Genetic Algorithm[D]. Dalian: Dalian Maritime University, 2015. (in Chinese)
- [7] ROUT M, SAIN D, SWAIN S, et al. PID Controller Design for Cruise Control System Using Genetic Algorithm[C] //IEEE International Conference on Electrical, Electronics, and Optimization Techniques. Chennai, India; IEEE,2016.
- [8] KANG J, MENG W, ABRAHAM A, et al. An Adaptive PID Neural Network for Complex Nonlinear System Control [J]. Neurocomputing, 2014, 135: 79-85.
- [9] FANG M C, ZHUO Y Z, LEE Z Y, et al. The Application of the Self-Tuning Neural Network PID Controller on the Ship Roll Reduction in Random Waves [J]. Ocean Engineering, 2010, 37 (7): 529-538.
- [10] 尹宝才, 王文通, 王立春. 深度学习研究综述[J]. 北京工业大学学报, 2015,41(1):48-59.
- YIN B C, WANG W T, WANG L C. Review of Deep Learning[J]. Journal of Beijing Polytechnic University,2015,41(1):48-59. (in Chinese)
- [11] DIEGEL O, BADVE A, BRIGHT G, et al. Improved Mechanism Wheel Design for Omni-Directional Robots[C] //Australasian Conference on Robotics and Automation. Auckland, 2002.
- [12] TóTH F, PAVOL K, MARTIN G, et al. Control Systems in Omni-Directional Robotic Vehicle with Mecanum Wheels[C]// IEEE International Conference on Process Control. Strbske Pleso, Slovakia; IEEE, 2013.
- [13] TANG K-S, MAN K F, CHEN G, et al. An Optimal Fuzzy PID Controller[J]. IEEE Transactions on Industrial Electronics, 2001,48(4):757-65.
- [14] 熊中刚, 叶振环, 贺娟, 等. 基于免疫模糊PID的小型农业机械路径智能跟踪控制[J]. 机器人, 2015,37(2):212-223.
- XIONG Z G, YE Z H, HE J, et al. Small Agricultural Machinery Path Intelligent Tracking Control Based on Fuzzy Immune PID[J]. Robot, 2015, 37(2): 212-223. (in Chinese)
- [15] 李镇, 樊炳辉, 王鑫, 等. 基于BP神经网络的仿生眼定位追踪算法[J]. 机器人, 2017,39(1):63-69.
- LI Z, FAN B H, WANG X, et al. Locating and Tracking Algorithm of Biomimetic Eye Based on Backpropagation Neural Network[J]. Robot, 2017, 39(1): 63-69. (in Chinese)
- [16] 张金龙, 徐慧, 刘京南, 等. 基于模糊神经网络的精密角度定位PID控制[J]. 仪器仪表学报, 2012,33(3):549-554.
- ZHANG J L, XU H, LIU J N, et al. PID Control Based on Fuzzy Neural Network for Precision Angular Alignment[J]. Chinese Journal of Scientific Instrument, 2012,33(3):549-554. (in Chinese)
- [17] NAWI N M, REHMAN M Z, KHAN A, et al. A New Bat Based Back-Propagation (BAT-BP) Algorithm[J]. Advances in Systems Science, 2014, 240: 395-404.
- [18] YANG Y, WANG G, YANG Y, et al. Parameters Optimization of Polygonal Fuzzy Neural Networks Based on GA-BP Hybrid Algorithm[J]. International Journal of Machine Learning and Cybernetics, 2014,5(5):815-822.
- [19] TIAN H, WANG J, XU W, et al. Automated System of Smart Ammeter Verification Based on Mobile Robot[C]// IEEE World Congress on Intelligent Control and Automation. Shenyang, China; IEEE,2014.

(编辑: 徐敏)