

2-19-2020

AUV Vertical Plane Control Based on Improved PID Neural Network Algorithm

Runan Huang

College of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China;

Ding Ning

College of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

AUV Vertical Plane Control Based on Improved PID Neural Network Algorithm

Abstract

Abstract: *An improved PID neural network controller is designed for the movement control of a small low-speed autonomous underwater vehicle (AUV) in vertical plane, and the global control of the depth and pitch angle of the underwater vehicle in vertical plane is obtained.* The AUV simulation control system is built by using REMUS underwater vehicle model in Simulink. The simulation results show that the improved control method with better dynamic performance has solved the original excessive saturation issue, and can adapt to different learning rates and network initial weight, and is of certain reference value to the practical application of the underwater vehicle.

Keywords

low-speed underwater vehicle, improved PID neural network, nonlinear system, multivariable global control, simulink simulation

Recommended Citation

Huang Runan, Ding Ning. AUV Vertical Plane Control Based on Improved PID Neural Network Algorithm[J]. Journal of System Simulation, 2020, 32(2): 229-235.

基于改进 PID 神经网络算法的 AUV 垂直面控制

黄茹楠, 丁宁

(燕山大学电气工程学院, 河北 秦皇岛 066004)

摘要: 针对一类小型低速自主水下航行器(AUV)的垂直面运动控制问题,设计了一种改进的 PID 神经网络控制器,实现对水下航行器在垂直面内深度和俯仰角的全局控制。利用 REMUS 水下航行器模型搭建了 Simulink 下 AUV 垂直面仿真控制系统,仿真结果表明,改进的控制方法克服了原方法中饱和区过大的问题,具有良好的动态性能同时能够适应不同的学习速率和网络初始权重,对水下航行器的工程实际应用具有一定参考价值。

关键词: 低速水下航行器; 改进 PID 神经网络; 非线性系统; 多变量全局控制; Simulink 仿真

中图分类号: TP273.2 文献标识码: A 文章编号: 1004-731X (2020) 02-0229-07

DOI: 10.16182/j.issn1004731x.joss.17-9168

AUV Vertical Plane Control Based on Improved PID Neural Network Algorithm

Huang Runan, Ding Ning

(College of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China)

Abstract: An improved PID neural network controller is designed for the movement control of a small low-speed autonomous underwater vehicle (AUV) in vertical plane, and the global control of the depth and pitch angle of the underwater vehicle in vertical plane is obtained. The AUV simulation control system is built by using REMUS underwater vehicle model in Simulink. The simulation results show that the improved control method with better dynamic performance has solved the original excessive saturation issue, and can adapt to different learning rates and network initial weight, and is of certain reference value to the practical application of the underwater vehicle.

Keywords: low-speed underwater vehicle; improved PID neural network; nonlinear system; multivariable global control; simulink simulation

引言

水下无人潜航器技术始于 20 世纪 50 年代,早期的水下航行器仅用于水下电缆铺接、深水勘探以及沉船打捞等民用领域,直至 20 世纪 90 年代,伴随着集成电路、计算机、网络通讯技术的飞速发展,

自主式水下航行器的军事能力也开始备受关注^[1-2]。但是由于水下环境复杂,动力学本质上是非线性的,并且水流的存在会产生多种不可测量的干扰,因此水下航行器的动力学问题对传统线性设计方法来说难以解决^[3-4]。研究人员贾鹤鸣等利用自适应反步法实现了对 AUV 的变深控制^[5], Riedel 针对 AUV 解耦模型提出了基于滑模控制的干扰补偿控制器并在 Phoenix AUV 上进行了实验验证^[6],曾德伟等利用反步法和小增益定理设计了一种 ISS 型神经网络控制器对 AUV 深度进行控制^[7]。但上述研究成果均需要对系统进行辨识或根据被控对象的



收稿日期: 2017-12-12 修回日期: 2018-11-08;
基金项目: 国家自然科学基金(61472341);
作者简介: 黄茹楠(1966-), 女, 河北, 博士, 副教授, 研究方向为计算机仿真、水下航行器运动控制; 丁宁(1993-), 男, 黑龙江, 硕士, 研究方向为水下航行器运动控制。

<http://www.china-simulation.com>

数学模型来设计控制器。本文以一种具有多变量解耦能力的 PID 神经网络的控制方法(MPIDNN)为基础设计了一种新型控制器,该控制器既具有原方法中无需对被控对象进行测量和辨识的优点,又克服了原方法中饱和区过大,网络收敛速率较慢等问题。仿真结果表明该控制器一方面具有传统 PID 控制器可靠性高、鲁棒性强的特点,另一方面通过神经网络在线学习的特性使控制器具有对非线性系统内的多变量实现全局控制的能力。

1 水下航行器的动力学模型

1.1 水下航行器的运动坐标系

为了建立水下航行器的动力学模型,应选用合适的坐标系作为参照,这里采用国际通用坐标系建立如图 1 所示的 2 个坐标系,分别为地坐标系 $EXYZ$ 和体坐标系 $Bxyz$ ^[8-9]。水下航行器沿 Bx 、 By 、 Bz 轴的速度分别为 u 、 v 、 w ,绕 Bx 、 By 、 Bz 轴的角速度 p 、 q 、 r ,偏航角 φ 为 Bx 轴分解到平面 EXY 内的向量和 EX 之间的夹角,头部偏向 EX 轴右侧为正;俯仰角 θ 为 Bx 轴与平面 EXY 之间的夹角,头部在平面 EXY 上时为正;横滚角 ψ 为平面 Bxz 与平面 EXZ 之间的夹角,尾部向头部看时 AUV 右滚为正^[10]。

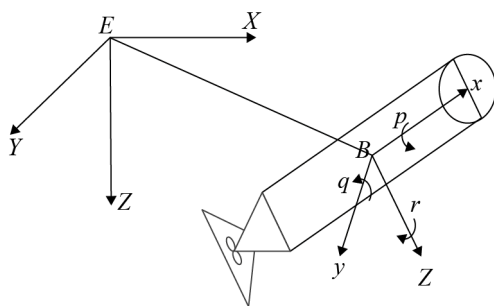


图 1 水下航行器坐标系示意图

Fig. 1 Schematic diagram of coordinate system for underwater vehicle

1.2 水下航行器垂直面运动建模

Fossen 曾建立 AUV 的非线性动力学方程^[11]:

$$M\dot{\mathbf{v}} = \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\theta) + \boldsymbol{\tau} \quad (1)$$

式中: M 为惯性矩阵(包括附加重力); $C(v)$ 为科氏

力和向心力矩阵; $D(v)$ 为阻力矩阵; $g(\theta)$ 为重力和浮力矩阵; τ 为控制输入矩阵。根据式(1)给出水下航行器在垂直面内受力情况如图 2 所示。

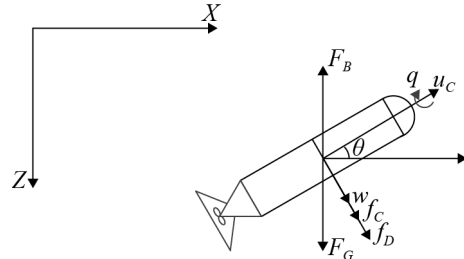


图 2 水下航行器垂直面内受力示意图

Fig. 2 Schematic diagram of force in vertical plane of underwater vehicle

因为水下航行器的运动具有较强的耦合性,水平速度对垂向上的建模也有着影响,所以这里假设体坐标系下水平初速度初始条件为 $u_0 = u_c$, 在此条件下来对水下航行器进行垂直面建模。图 2 中 w 为体坐标系下的垂向速度; q 为纵倾角速度; θ 为俯仰角; $f_C(w, q)$ 为水下航行器所受到的科氏力和向心力; $f_D(w, q)$ 为水下航行器所受到的阻力; F_B 、 F_G 分别为水下航行器所受的浮力与重力。利用式(1)可以得到 AUV 在垂直面内的运动模型^[12-13]:

$$\begin{cases} M \begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = f_C + f_D + f_G + \tau \\ \begin{bmatrix} \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} -u_c \sin \theta \\ 0 \end{bmatrix} \end{cases} \quad (2)$$

$$\text{式中: } f_C = \begin{bmatrix} m(u_c q + z_g q^2) \\ -m(z_g w q + x_g u_c q) \end{bmatrix};$$

$$f_D = \begin{bmatrix} Z_{ww} w |w| + Z_{qq} q |q| + Z_{uq} u_c |q| + Z_{uw} u_c |w| \\ M_{ww} w |w| + M_{qq} q |q| + M_{uq} u_c |q| + M_{uw} u_c |w| \end{bmatrix};$$

$$f_G = \begin{bmatrix} (F_w - F_B) \cos \theta \\ -(z_g F_w - z_B F_B) \sin \theta - (x_g F_w - x_B F_B) \cos \theta \end{bmatrix};$$

$$M = \begin{bmatrix} m - Z_{\dot{w}} & -m x_g - Z_{\dot{q}} \\ -m x_g - M_{\dot{w}} & I_{yy} - M_{\dot{q}} \end{bmatrix}; \quad \tau = \begin{bmatrix} F \\ Z \end{bmatrix}; \text{ 其中}$$

F 、 Z 分别为垂直方向所受的外力和外力矩。

针对由式(2)给出的复杂非线性动力学模型,本文设计了一种改进 PID 神经网络控制器,目标是实现深度 z 和俯仰角 θ 的综合控制。

2 MPIDNN 神经网络算法及改进

2.1 PID 神经网络简介

PIDNN 神经元网络是由舒怀林教授提出的一种神经网络算法, 这种神经网络定义了具有比例、积分、微分功能的神经元, 将 PID 控制规律融合进神经网络之中^[14]。MPIDNN 是单变量 PIDNN 的扩充和发展形式, 如果被控对象为个 m 输入, n 个输出的系统, MPIDNN 网络则对应有 n 个输入, m 个输出, 其网络结构为 $2n \times 3n \times m$, 以 2 输入 2 输出系统为例, MPIDNN 网络结构如图 3 所示, 其中被控对象是无需测量和辨识的^[15]。

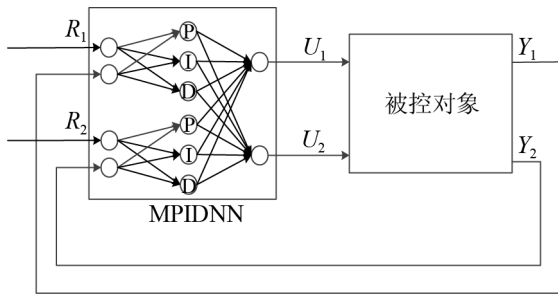


图 3 MPIDNN 网络结构图

Fig. 3 MPIDNN Network structure diagram

2.2 PID 神经网络算法^[16-18]

2.2.1 前向算法

(1) 输入层

输入层有 n 个神经元, 其输入为:

$$\begin{cases} net_{s1}(k) = r_s(k) \\ net_{s2}(k) = y_s(k) \end{cases} \quad (3)$$

神经元状态为:

$$u_{si}(k) = net_{si}(k) \quad (4)$$

神经元输出为:

$$x_{si}(k) = \begin{cases} -1, u_{si}(k) < -1 \\ u_{si}(k), -1 \leq u_{si}(k) \leq 1 \\ 1, u_{si}(k) > 1 \end{cases} \quad (5)$$

(2) 隐含层

隐含层有 $3n$ 个神经元, 其中有 n 个比例元, n 个积分元和 n 个微分元, 它们的输入都相同均为:

$$net'_{sj}(k) = \sum_{i=1}^2 w_{sij} x_{si}(k) \quad (6)$$

假设采样时间为 T_s , 比例、积分、微分神经元的状态分别为:

$$\begin{cases} u'_{s1}(k) = net'_{s1}(k) \\ u'_{s2}(k) = u'_{s2}(k-1) + net'_{s2}(k) * T_s \\ u'_{s3}(k) = (net'_{s3}(k) - net'_{s3}(k-1)) / T_s \end{cases} \quad (7)$$

3 者输出相同均为:

$$x'_{sj}(k) = \begin{cases} 1, u'_{sj}(k) > 1 \\ u'_{sj}(k), -1 \leq u'_{sj}(k) \leq 1 \\ -1, u'_{sj}(k) < -1 \end{cases} \quad (8)$$

(3) 输出层

输出层有 m 个神经元, 其输入为:

$$net''_h(k) = \sum_{s=1}^n \sum_{j=1}^3 w'_{sjh} x'_{sj}(k) \quad (9)$$

神经元状态为:

$$u''_h(k) = net''_h(k) \quad (10)$$

神经元输出为:

$$x''_h(k) = \begin{cases} 1, u''_h(k) > 1 \\ u''_h(k), -1 \leq u''_h(k) \leq 1 \\ -1, u''_h(k) < -1 \end{cases} \quad (11)$$

整个 MPIDNN 系统的输出值等于输出层神经元的输出值为:

$$v_h(k) = x''_h(k) \quad (12)$$

式中: s 为子网络序号($s=1,2,3 \dots n$); i 为输入层神经元序号($i=1,2$); j 为隐含层神经元序号($j=1,2,3$); h 为输出层神经元序号($h=1,2,3 \dots m$); 无上标表示输入层变量; 上标“'”表示隐含层变量, 上标“''”表示输出层变量。

2.2.2 反传算法

神经网络算法的最重要特点在于其能够自我学习, 不断修正各个网络层之间的连接权值。在 MPIDNN 中, 采用的权值修正是一种基于 δ 学习规则(梯度下降法)的方法。其核心是使网络中理想输出 $r_s(k)$ 和实际输出 $y_s(k)$ 的偏差的平方均值最小。定义系统偏差 J 为:

$$J = \frac{1}{2l} \sum_{s=1}^n \sum_{k=1}^l [r_s(k) - y_s(k)]^2 \quad (13)$$

式中: $y_s(k)$ 可以展开用神经网络权值矩阵 W 来表达, 根据网络的输出调整 W 是神经网络自我学习

的重要环节, W 的调整分为两部分, 一部分为式(6)中构成的输入矩阵, 另一部分为式(9)中构成的输出矩阵, 权值调整的目标是最小化系统偏差函数 $J(W)$ 。经过 n_0 步训练后, W 的迭代方程为:

$$W(n_0 + 1) = W(n_0) - \eta \frac{\partial J}{\partial W} \quad (14)$$

(1) 隐含层至输出层(输出矩阵)

由式(14)可得隐含层至输出层的权重迭代公式为:

$$W_o(n_0 + 1) = W_o(n_0) - \eta_1 \frac{\partial J}{\partial W_o} \quad (15)$$

根据前向算法展开这一项可得:

$$\frac{\partial J}{\partial W_o} = -\frac{1}{l} \sum_{s=1}^n \sum_{h=1}^m \sum_{k=1}^l \left\{ [r_s(k) - y_s(k)] \frac{\partial y_s(k)}{\partial v_h(k)} x'_{sj}(k) \right\} \quad (16)$$

式中: $\frac{\partial y_s(k)}{\partial v_h(k)} = \text{sgn}\left(\frac{y_s(k+1) - y_s(k)}{v_h(k) - v_h(k-1)}\right)$, 这样的替代

在算法上是许可的, 因为它的符号的正负决定了权值的变化方向, 大小只影响变化速度, 这可以用学习速率加以调节。

(2) 输入层至隐含层(输入矩阵)

同理权重迭代公式为:

$$W_i(n_0 + 1) = W_i(n_0) - \eta_2 \frac{\partial J}{\partial W_i} \quad (17)$$

展开有:

$$\frac{\partial J}{\partial W_i} = -\frac{1}{l} \sum_{s=1}^n \sum_{h=1}^m \sum_{k=1}^l \left\{ [r_s(k) - y_s(k)] \frac{\partial y_s(k)}{\partial v_h(k)} \cdot W_o \frac{\partial u'_{sj}(k)}{\partial \text{net}'_{sj}(k)} x_{si}(k) \right\} \quad (18)$$

同理其中 $\frac{\partial y_s(k)}{\partial v_h(k)} = \text{sgn}\left(\frac{y_s(k+1) - y_s(k)}{v_h(k) - v_h(k-1)}\right)$

$$\frac{\partial u'_{sj}(k)}{\partial \text{net}'_{sj}(k)} = \text{sgn}\left(\frac{u'_{sj}(k) - u'_{sj}(k-1)}{\text{net}'_{sj}(k) - \text{net}'_{sj}(k-1)}\right)$$

2.3 MPIDNN 算法的改进

(1) 活化函数改进

由式(11)可知, 当输出层网络状态 $u''_h(k)$ 持续出现 $u''_h(k) > 1$ 或 $u''_h(k) < -1$ 的情况时, 会导致 $v_h(k) = v_h(k-1)$ 造成网络卡顿, 大大影响收敛速度, 为了削弱这种影响, 本文采用 $\tanh(x)$ 作为系统

的活化函数, 这样即使 x 很大, 仍能分辨变化方向, 加快了网络收敛速度同时能够有效的将输出控制在 $[-1, 1]$ 内, 符合控制系统的需求, 同时将这种改进应用在隐含层上。

改进后的系统隐含层输出:

$$x'_{sj}(k) = \tanh(u'_{sj}(k)) \quad (19)$$

输出层输出:

$$x''_h(k) = \tanh(u''_h(k)) \quad (20)$$

改进后的权值修正算法:

$$\frac{\partial J}{\partial W_o} = -\frac{1}{l} \sum_{s=1}^n \sum_{h=1}^m \sum_{k=1}^l \left\{ [r_s(k) - y_s(k)] \frac{\partial y_s(k)}{\partial v_h(k)} \cdot [1 - \tanh^2(u''_h(k))] x'_{sj}(k) \right\} \quad (21)$$

$$\frac{\partial J}{\partial W_i} = -\frac{1}{l} \sum_{s=1}^n \sum_{h=1}^m \sum_{k=1}^l \left\{ [r_s(k) - y_s(k)] \frac{\partial y_s(k)}{\partial v_h(k)} [1 - \tanh^2(u''_h(k))] \cdot W_o \cdot [1 - \tanh^2(u'_{sj}(k))] \cdot \frac{\partial u'_{sj}(k)}{\partial \text{net}'_{sj}(k)} x_{si}(k) \right\} \quad (22)$$

(2) 增加动量项

PIDNN 调整权重时只按 n_0 步时误差的梯度下降方向考虑, 没有考虑以前梯度的方向, 从而使学习过程发生震荡, 收敛缓慢, 为了提高网络学习速度, 可以在权值调整公式中增加动量项即:

$$W(n_0 + 1) = W(n_0) - \eta \frac{\partial J}{\partial W} + \alpha(W(n_0) - W(n_0 - 1)) \quad (23)$$

式中: α 为动量系数。

(3) 变量归一化

与普通 MPIDNN 算法相比, 改进后的 MPIDNN 算法没有对神经网络输入层的输出进行阈值处理, 而是在整个网络的输入上直接对输入变量进行归一化处理, 这样做的目的有两点: 一是当净输入绝对值过大时, 容易进入激活函数的相对平坦区, 导致网络收敛速度变慢; 二是对所有输入变量进行归一化处理可以使网络学习一开始所有输入处于平等的地位, 有利于加快整个网络收敛速度。本文将所有输入数据线性压缩在区间内, 归一化公式如式(24):

$$\bar{x}_i = \frac{2x_i - (x_{\max} + x_{\min})}{x_{\max} - x_{\min}} \quad (24)$$

式中: x_i 为输入数据; $\bar{x}_i$ 为处理后的输入数据。

3 仿真研究

本文采用 Simulink 搭建了整个水下航行器运动控制仿真系统, 其中 MPIDNNController 采用 Subsystem 子模块的形式对 S 函数编写的 MPIDNN 控制算法进行封装。uuvmodel 是用 S 函

数编写的 REMUS 水下航行器模型, 控制器输出为垂向上的力 F 、力矩 Z 作为 REMUS 水下航行器的输入, 系统总体仿真框图如图 4 所示。

当 AUV 水平速度为 3 节, 即 $u \approx 1.54 \text{ m/s}$ 时, REMUS 水下航行器垂直方向上建模的相关参数如表 1 所示^[19]。

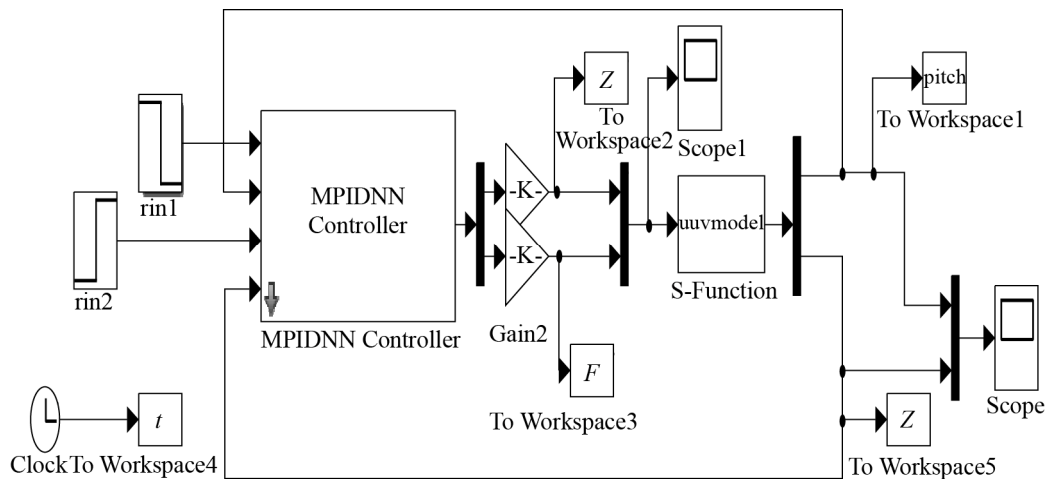


图 4 Simulink 下系统总体仿真框图

Fig. 4 Overall system simulation block diagram in Simulink environment

表 1 REMUS 模型仿真参数表

Tab. 1 Simulation parameter table of REMUS model

变量	数值	变量	数值
m	30.5 kg	Z_w	-35.5 kg
x_g	0 m	Z_q	-1.93 kg·m/rad
M_w	-1.93 kg·m	I_{yy}	3.45 kg·m ²
M_q	-4.88 kg·m ² /rad	X_B	-0.611 m
z_g	0.019 6 m	Z_{ww}	-131 kg·m
Z_{qq}	-0.632 kg·m/rad ²	Z_{uq}	-5.22 kg/rad
Z_{uw}	-28.6 kg/m	M_{ww}	3.18 kg
M_{qq}	-188 kg·m ² /rad ²	M_{uq}	-2 kg·m/rad
M_{uw}	24 kg	F_w	298.9 N
F_B	308 N	z_B	0 m

3.1 PIDNN 算法与改进 PIDNN 算法对比

输入层至隐含层连接初值选取 $w_{s1j}=1$, $w_{s2j}=-1$, 输出层连接初值选取当 $s=h$ 时, $w'_{s1h}=10$, $w'_{s2h}=1$, $w'_{s3h}=1$, 当 $s \neq h$ 时, $w'_{sjh}=0$, 采样时间 $T_s=0.001 \text{ s}$, 假设初始状态 $w_0=z_0=q_0=0$, $\theta_0=45^\circ$ 实现

期望为 $z_h=50 \text{ m}$, $\theta_h=0^\circ$ 的全局控制, 在仿真时间为 1 s 时给定期望信号, 仿真时长设定为 20 s, 动量系数 $\alpha=0.001$, 只考虑活化函数对 PIDNN 网络的影响, 仿真结果如图 5 所示。

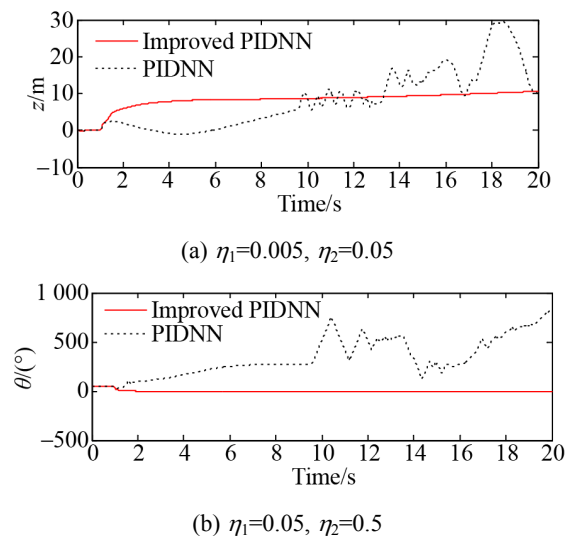


图 5 改进 PIDNN 与 PINN 仿真结果对比图

Fig. 5 Comparison of simulation results between improved PIDNN and PINN

由仿真结果可知, 当学习速率较小时, PIDNN 算法开始震荡经过一段时间后能够跟踪给定信号, 而进一步增大学习速率, PIDNN 算法很快进入饱和区, 造成神经网络卡顿, 系统开始发散, 而改进后的 PIDNN 算法克服了原方法中饱和区过小的问题, 并且能够很好的跟踪给定信号。

3.2 不同参数对改进 PIDNN 算法的影响

3.2.1 不同初始网络权重对改进 PIDNN 算法的影响

在 $\eta_1=0.005$, $\eta_2=0.05$ 情况下, 输入层至隐含层连接初值选取 $w_{s1j}=1$, $w_{s2j}=-1$, 分别选取不同隐含层至输出层连接初值当 $s=h$ 时, $w'_{s1h}=p$, $w'_{s2h}=i$, $w'_{s3h}=d$, 当 $s \neq h$ 时, $w'_{sjh}=0$, 仿真时长为 10 s, 不同初始网络权重对控制器效果影响如图 6 所示。

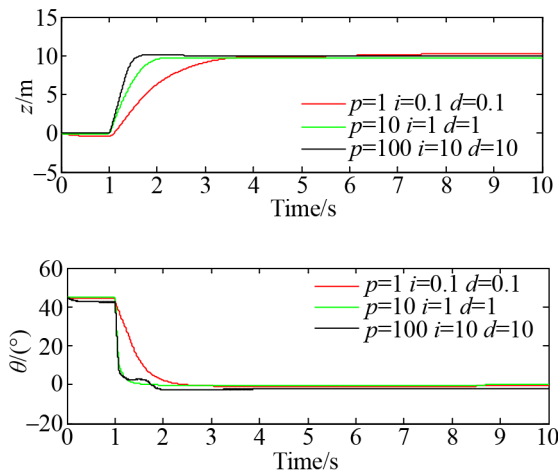


图 6 不同初始网络权重下控制器的全局控制效果图
Fig. 6 Global control effect graph of controller under different initial network weights

3.2.2 不同学习速率对改进 PIDNN 算法的影响

输入层至隐含层连接初值选取 $w_{s1j}=1$, $w_{s2j}=-1$, 隐含层至输出层选取当 $s=h$ 时, $w'_{s1h}=5$, $w'_{s2h}=0.5$, $w'_{s3h}=0.5$, 当 $s \neq h$ 时, $w'_{sjh}=0$, 不同的学习速率对控制器的影响如图 7 所示。

由仿真结果可知, PIDNN 算法能够对不同的学习速率和初始权重参数有良好的自适应性, 这大大减少了实际应用中参数调试环节的工作量, 对于

水下航行器工程实践环节具有一定的参考价值。

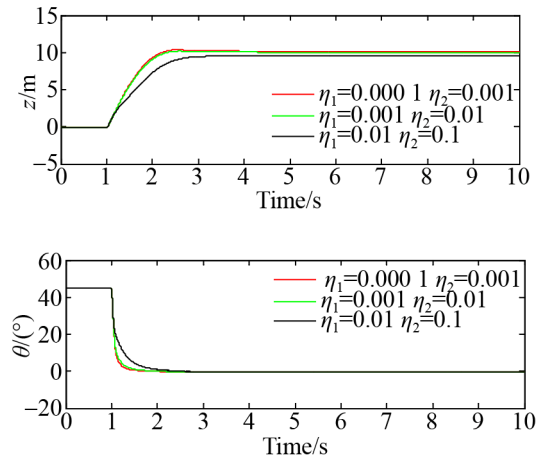


图 7 不同学习速率下控制器的全局控制效果图
Fig. 7 Global control effect graph of the controller under different learning rates

4 结论

(1) 提出了一种改进的 PID 神经网络控制方法, 解决了原方法中收敛速度慢、饱和区过大的问题。

(2) 通过 Simulink 下搭建的 AUV 运动控制仿真系统完成了整个实验, 证明了所设计控制器的可行性和有效性。

(3) 仿真结果表明: 与原方法相比, 改进的 PIDNN 算法克服了并且能够更快更稳定的实现 AUV 的深度和俯仰角的全局控制。同时能够较大范围的适应不同的学习速率和初始网络权重, 对于减小水下航行器工程实践中参数调节环节的工作量有所帮助。

参考文献:

- [1] 张乃千, 马建光. 无人潜航器: 未来的“海底侦察兵”[N]. 科技日报, 2015-02-10(12).
Zhang Naiqian, Ma Jianguang. Unmanned underwater vehicles: "sea scouts" of the future[N]. Science and Technology Daily, 2015-02-10(12).
- [2] Cook D A, Fernandez J E, Stroud J S, et al. AUV-based synthetic aperture sonar: Initial experiences and insights[J]. Journal of the Acoustical Society of America (S0001-4966), 2004, 115(115): 2615-2615.
- [3] Wadoo S, Kachroo P. Autonomous underwater vehicles:

- modeling, control design, and simulation[M]. Crc Press, 2011: 29-41.
- [4] 高富东, 潘存云, 杨政, 等. 多矢量推进水下航行器 6 自由度非线性建模与分析[J]. 机械工程学报, 2011, 47(5): 93-100.
Gao Fudong, Pan Cunyun, Yang Zheng, et al. Nonlinear mathematics modeling and analysis of the vectored thruster autonomous underwater vehicle in 6-DOF motions[J]. Journal of Mechanical Engineering, 2011, 47(5): 93-100.
- [5] 贾鹤鸣, 宋文龙, 陈子印. 基于自适应反步法的自主水下机器人变深控制[J]. 华南理工大学学报(自然科学版), 2013, 41(1): 15-20.
Jia Heming, Song Wenlong, Chen Ziyin. Diving Control of Autonomous Underwater Vehicle Based on Adaptive Backstepping Method[J]. Journal of South China University of Technology (Natural Science Edition), 2013, 41(1): 15-20.
- [6] Riedel J S. Shallow water stationkeeping of an autonomous underwater vehicle: the experimental results of a disturbance compensation controller[C]// Oceans. Providence, RI, USA: IEEE, 2000, 2: 1017-1028.
- [7] 曾德伟, 吴玉香, 王聪. AUV 深度的神经网络辨识和学习控制仿真研究[J]. 计算机工程与应用, 2017, 53(6): 258-263.
Zeng Dewei, Wu Yuxiang, Wang Cong. Research on simulation of neural network identification and learning for AUV depth control[J]. Computer Engineering and Applications, 2017, 53(6): 258-263.
- [8] Roberts G N, Sutton R. Advances in Unmanned Marine Vehicles[M]. Further Advances in Unmanned Marine Vehicles. UK: IET Digital Library, 2012: 429.
- [9] Elmokadem T, Zribi M, Youcef-Toumi K. Terminal sliding mode control for the trajectory tracking of underactuated Autonomous Underwater Vehicles[J]. Ocean Engineering, 2017, 129: 613-625.
- [10] Yuh J. Design and Control of Autonomous Underwater Robots: A Survey[J]. Autonomous Robots (S0929-5593), 2000, 8(1): 7-24.
- [11] Fossen T I. Guidance and control of ocean vehicles[M]. New York: John Wiley & Sons, 1994: 18-20.
- [12] Fossen T I. Nonlinear Modelling and Control of Underwater Vehicles[D]. Norwegian: Norwegian Institute of Technology, 1991: 52-60.
- [13] 高剑, 徐德民, 赵宁宁. 一类低速水下航行器的自适应反演运动控制[J]. 系统仿真学报, 2008, 20(7): 1800-1802.
Gao Jian, Xu Demin, Zhao Ningning. Adaptive Backstepping Motion Control of Underwater Vehicle with Vertical Thrusters[J]. Journal of System Simulation, 2008, 20(7): 1800-1802.
- [14] Zhang Z, Ma C, Zhu R. Self-Tuning Fully-Connected PID Neural Network System for Distributed Temperature Sensing and Control of Instrument with Multi-Modules[J]. Sensors (S1424-8220), 2016, 16(10): 1709.
- [15] 杨志刚, 钱俊磊. 基于 PID 神经网络的电加热炉温度控制[J]. 工业控制计算机, 2012, 25(12): 8-9.
Yang Zhigang, Qian Junlei. Electric Furnace Temperature Control Based on PID Neural Network[J]. Industrial Control Computer, 2012, 25(12): 8-9.
- [16] 舒怀林. PID 神经元网络多变量控制系统分析[J]. 自动化学报, 1999, 25(1): 105-111.
Shu Huailin. Analysis of PID Neural Network Multivariable Control Systems[J]. Acta Automatic Sinica, 1999, 25(1): 105-111.
- [17] Cheng M L, Chen R. Optimal Self-Tuning PID Controller Based on Low Power Consumption for a Server Fan Cooling System[J]. Sensors (S1424-8220), 2015, 15(5): 11685.
- [18] Rossomando F G, Soria C M. Identification and control of nonlinear dynamics of a mobile robot in discrete time using an adaptive technique based on neural PID[J]. Neural Computing and Applications (S0941-0643), 2015, 26(5): 1179-1191.
- [19] Prestero T. Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle[D]. USA: MIT/WHOI Joint Program in Applied Ocean, 2001: 101-112.