

一种水下机器人运动的过程神经元控制

唐旭东^{1,2}, 庞永杰^{1,2}, 李 晔^{1,2}

(1. 哈尔滨工程大学 水下机器人技术国防科技重点实验室, 黑龙江 哈尔滨 150001;

2. 哈尔滨工程大学 船舶工程学院, 黑龙江 哈尔滨 150001)

摘要: 过程神经网络是传统神经网络的拓展, 增加了一个对于时间的聚合算子, 从而更好地模拟了生物神经元的信息处理机制. 这是由于水下机器人运动控制系统的输入、输出均是随时间连续变化的过程量. 结合S函数和预先规划思想, 建立水下机器人过程神经元的运动控制模型. 仿真试验证明, 该新型控制模型, 对于水下机器人的运动非线性控制器具有设计简单、响应速度快、超调小、鲁棒性好等优点.

关键词: 水下机器人; 过程神经元控制算法; S函数; 预先规划

中图分类号: TP24 **文献标识码:** A

A process neural control algorithm for autonomous underwater vehicle

TANG Xu-dong^{1,2}, PANG Yong-jie^{1,2}, LI Ye^{1,2}

(1. Key Laboratory of Science and Technology for Nation Defence of Autonomous Underwater Vehicle, Harbin Engineering University, Harbin Heilongjiang 150001, China;

2. College of Shipbuilding Engineering, Harbin Engineering University, Harbin Heilongjiang 150001, China)

Abstract: Process neural network is extended from a traditional neural network by adding an integration operator of time to better imitate the information processing system of a biological neuron. This is because both the input and output of motion control system for automatic underwater vehicles (AUVs) is processing vectors which is related with time. Combining the S function and the pre-planning idea, we develop a motion-control model for the process neuron of an AUV. Simulation results show that the new control is useful for underwater vehicles, featuring with higher precision, simpler design, faster response, and better robustness.

Key words: autonomous underwater vehicles; process neural control algorithm; S function; pre-planning

1 引言(Introduction)

受到输入的同步瞬时限制, 传统人工神经网络难以表达时间序列中实际存在的时间累积效应, 也难以解决较大样本的学习和泛化问题, 因此传统人工神经网络在解决复杂非线性时间序列问题时还存在一定的不适应性. 为解决此类问题, 文献[1]提出了一种新的人工神经网络模型—过程神经网络, 它与传统人工神经元在结构上类似, 不同之处在于过程神经元的输入及相应的连接权可以是时变函数, 其聚合运算既体现了对空间的加权聚合, 又体现了对时间过程的累积聚合. 因此, 过程神经网络对于求解与时变过程相关的众多实际问题具有较强的适应性^[2].

作为水下机器人运动控制系统, 受到非线性流体动力和以潮流为代表的环境干扰, 以及机器人自身

运动的影响, 系统的输入、输出均是随时间连续变化的过程量^[3,4], 并且系统的输入/输出之间具有实时的对应关系. 本文旨在文献[2]的基础上, 尝试构建一种基于S模型的过程神经元控制器. 本控制器在基本过程神经元模型的基础上结合具有模糊逻辑的S函数, 同时引入系统输出的预先规划思想, 从而实现了简易的过程神经元控制器. 通过水下机器人动力学模型进行仿真试验验证了这种控制方法的可行性.

2 过程神经元(Process neural)

过程神经元由加权、聚合、激励3部分组成^[1,2], 其结构如图1所示. 根据算子 $K(\cdot)$ 的不同, 可将过程神经元认为2类.

第1类过程神经元的聚合运算只含有加权聚合运算, 其输入、输出关系可表示为

收稿日期: 2007-10-22; 收修改稿日期: 2008-08-29.

基金项目: 863基金资助项目(2008AA092301); 中国博士后科学基金资助项目(20080440838); 黑龙江省博士后资助项目; 哈尔滨工程大学基础研究基金资助项目(HEUFT08001); 水下智能机器人技术国防科技重点实验室开放课题研究基金资助项目(2008003, 2007001).

$$y(t) = f\left(\sum_{i=1}^n \sum_{l=1}^L w_i^l b_l(t) x_i(t) - \theta\right). \quad (1)$$

第2类过程神经元的聚合运算不仅仅含有空间加权聚合运算,而且含有时间过程累积聚和运算,其输入、输出关系可表示为

$$y(t) = f\left(\sum_{i=1}^n \int_0^T \sum_{l=1}^L w_i^l b_l(t) x_i(t) dt - \theta\right). \quad (2)$$

其中: $x_i(t)$ 为过程神经元的输入函数, w_i^l 为相应的连接权函数; $b_l(t)$ 为权值基函数; L 为权值基函数的个数; θ 为过程神经元阈值; $f(\cdot)$ 为激励函数; $[0, T]$ 为样本采样区间。

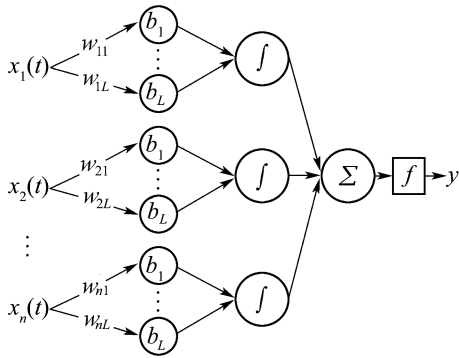


图1 过程神经元

Fig. 1 The process neural

3 基于S模型的过程神经元控制器(Process neural control based on S model)

过程神经元的输入是一个过程,数据量较大,网络训练需要花费较大的时间和空间开销。水下机器人的作业任务及工作环境具有突变性,其本身也是一个强耦合的非线性系统,受控于对象的非线性、不确定性,过程神经元样本的选取和在线的训练都具有较大难度。同时,控制系统的输入一般为传感器采集的当前位姿信息,这是一瞬态量,如何将瞬态量转换成过程量,这也是需要解决的问题。

3.1 S模型的引入(S model introduction)

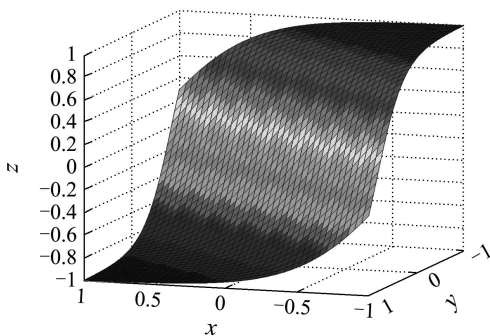


图2 S面函数

Fig. 2 Curve of S surface

为了改善控制的精度,简化离线学习过程,本文

引入S模型的概念:从常规的模糊控制规则出发,取对角线数值连成折线,用一条光滑曲线来拟合。设计模糊控制器的时候,通常选用两边疏松、中间密的形式(即偏差大时控制比较粗,偏差小的时候控制比较细),在这一点上,其变化趋势与Sigmoid函数的变化趋势是一致的。因此呈S型变换形式的Sigmoid函数在一定程度上体现了模糊控制的思想(见图1)。同理,整个模糊控制规则库对应的折线面,也就可以用光滑曲面代替^[5]。

一般的, Sigmoid曲面函数为

$$z = \frac{2.0}{1.0 + \exp(-k_1 x - k_2 y)} - 1.0. \quad (3)$$

这里,所引入的S模型可以表示为

$$f = \frac{2.0}{1.0 + \exp(-k_1 x_1 - k_2 x_2)} - 1.0. \quad (4)$$

对于水下机器人的运动控制器来说,控制器的输入通常采用经过归一化处理后的机器人位置偏差 e 和偏差变化率 e' 。取 $x_1 = e$, $x_2 = e'$,则所引入的S模型就相当于一个函数化后的简易模糊控制器,其自身体现的控制思想和模糊控制是吻合的。

将过程神经元的激励函数 $f(\cdot)$ 取为S模型,构建了一个简易的过程神经元控制模型。

3.2 学习算法(Learning algorithm)

1) 数据采集与处理。

水下机器人运动控制系统通常采样 $T = 0.5$ s一个周期,为了获得足够多传感器数据样本,传感器采样间隔为0.05 s。采样得到的一组数据,选取一组正交三角基函数,基函数的个数可以依据逼近函数的精度要求和网络训练的泛化能力调节。在选定的正交基函数系下将拟合的输入多项式展开成 $[0, T]$ 的傅立叶级数^[6]。

2) 在线学习过程。

过程神经网络的学习可以借鉴传统的梯度下降法。考虑到水下机器人各个自由度运动控制是独立分开的,本文设计的控制器也是各自由度的解耦控制,因此,这里给出通用的控制模块。将式(2)改写为

$$u(k) = f\left(\sum_{i=1}^2 \int_0^T w_i(t) x_i(t) dt - \theta(k)\right). \quad (5)$$

其中 $w_i(t) = \sum_{l=1}^L w_i^l(k) b_l(t)$ 。目标函数定义为

$$J = \frac{1}{2} e(k)^2 = \frac{1}{2} (u(k) - u_d(k))^2. \quad (6)$$

其中 u_d 为系统规划器的输出,即期望值输出; u 为实际输出。由梯度下降法,网络权值学习规则为

$$\Delta w_i^l(k) = \alpha \Delta w_i^l(k-1) - \eta \frac{\partial J}{\partial w_i^l(k-1)}, \quad (7)$$

$$\Delta \theta(k) = \beta \Delta \theta(k-1) - \mu \frac{\partial J}{\partial \theta(k-1)}, \quad (8)$$

$$w_i^l(k) = w_i^l(k-1) + \Delta w_i^l(k), \quad (9)$$

$$\theta(k) = \theta(k-1) + \Delta \theta(k), \quad (10)$$

其中 $i = 1, 2$, α, β 为动量学习因子, η, μ 为学习率. 将S模型代入, 令

$$m(k) = \sum_{i=1}^2 \int_0^T w_i(t) x_i(t) dt - \theta(k), \quad (11)$$

$$f(m(k)) = \frac{2}{1 + \exp(m(k))} - 1.0, \quad (12)$$

则

$$\frac{\partial J}{\partial w_i^l(k-1)} = e(k-1) f'(m(k-1)) \cdot \int_0^T b_l(t) x_i(t) dt, \quad (13)$$

$$\frac{\partial J}{\partial \theta(k-1)} = -e(k-1) f'(m(k-1)). \quad (14)$$

其中 $f'(m(k-1)) = -\frac{\exp(m(k-1))}{(1 + \exp(m(k-1)))^2}$.

4 系统预先规划(Desired state programming)

传统神经网络控制中, 雅可比系数 $\frac{\partial y}{\partial u}$ 求取是一大难题. 规划器的建立, 从机器人当前时刻最大驱动能力出发, 得到机器人控制器的近似期望输出. 而将过程神经元控制器输出 u 与规划器输出 u_d 相联系, 从数值上近似取代了雅可比系数. 在整个控制精度影响不大的情况下, 降低求解的难度.

记机器人位置为

$$\hat{R} = \{n, e, d, \Phi\}, \quad (15)$$

目标位置为

$$\hat{R}_n = \{n_n, e_n, d_n, \Phi_n\}, \quad (16)$$

以机器人位置为基准的目标位置可记为

$$\begin{aligned} \hat{S} &= \hat{R}_n - \hat{R} = \\ &\{n_n - n, e_n - e, d_n - d, \Phi_n - \Phi\}, \end{aligned} \quad (17)$$

那么, 控制就是要使 $\hat{S}$ 与 $\hat{V}$ 趋向零, 对控制而言, 机器人随体坐标系最为方便, 所以, 通过坐标变换可改记为

$$\hat{S} = \{S_x, S_y, S_z, S_\Phi\}. \quad (18)$$

对输出的预先规划可表示为:

$$u_d = \hat{V}' = \{a_x, a_y, a_z, a_\Phi\} = \hat{f}(\hat{S}, \hat{V}). \quad (19)$$

正如最优控制性能指标函数的选取具有很大的自由度, 式(19)中的 $f(\cdot)$ 的具体形式可以根据动力系统得驱动能力与设计者的综合考虑作多种选择:

$$u_d = P \hat{a}_{\max}. \quad (20)$$

其中 $\hat{a}_{\max}$ 为限制机器人的最大加速度, 主要取决于动力系统驱动能力, P 为4阶对角阵:

$$P = \begin{bmatrix} p_1 & & & 0 \\ & p_2 & & \\ & & p_3 & \\ 0 & & & p_4 \end{bmatrix}, \quad (21)$$

$$\begin{cases} p_1 = (p_x/p_{xy}) \text{th}(p_{xy}/2), \\ p_2 = (p_y/p_{xy}) \text{th}(p_{xy}/2), \\ p_3 = \text{th}(p_z/2), \\ p_4 = \text{th}(p_\Phi/2), \end{cases} \quad (22)$$

$$\begin{cases} p_x = S_x^* - c_x V_x, \\ p_y = S_y^* - c_y V_y, \\ p_z = S_z^* - c_z V_z, \\ p_\Phi = S_\Phi^* - c_\Phi V_\Phi, \\ p_{xy} = \sqrt{(p_x^* p_x + p_y^* p_y)}, \end{cases} \quad (23)$$

$S_x^*, S_y^*, S_z^*, S_\Phi^*$ 分别称为 x, y, z, Φ 方向的牵引距离, 要定义为

$$S_i^* = \begin{cases} S_{i \max}^*, & S_i \geq S_{i \max}^*, \\ S_i, & -S_{i \max}^* < S_i < S_{i \max}^*, \\ -S_{i \max}^*, & S_i \leq -S_{i \max}^*, \end{cases} \quad (24)$$

其中 $i, x, y, z, \Phi, S_{i \max}^*$ 与 C_i 为待定正参数, 如果希望 i 方向的远程转移速度为 $V_{i \max}$, 则

$$S_{i \max}^* - C_i V_{i \max} = 0. \quad (25)$$

另一个可用的约束条件是从下列微分方程中求得这样的 $S_{i \max}^*$ 与 C_i , 使得对于所有的 $t > t_0$, 有 $S_i > 0$, 并有尽可能最小的 $t_n > t_0$:

$$\begin{aligned} S_i^* &= a_{i \max} \frac{1 - \exp(C_i S_i - S_i)}{1 + \exp(C_i S_i)}, \\ S_i &= S_{i \max}^*, S_i' = V_{i \max}, t = t_0. \end{aligned} \quad (26)$$

并使得对于所有的 $t > t_n$ 有

$$S_i < \text{error}_i, \quad (27)$$

error_i 为预先规定的 i 方向的位置精度.

5 试验结果(The results of experiment)

5.1 机器人数学模型(AUV mathematic model)

考虑到水下机器人一般存在3个平面的对称, 水

下机器人的运动模型可以简化为^[7]

$$M_v \dot{v} + C_v(v)v + D(v)v + G = T_v. \quad (28)$$

其中 $v = [u \ v \ w \ p \ q \ r]$ 分别为6个自由度上的运动速度与角速度, M_v 为质量阵, 包括了质量以及惯性质量, 它是正定阵, $C_v(v)$ 是与速度有关的惯性力系数矩阵, $D(v)$ 为粘性水动力矩阵, G 表示为重力浮力等静态力(矩)在艇体坐标系下的分量, T_v 为执行机构所产生的6个自由度上的推力与力矩。

实际的水下机器人一般都是上下对称、左右对称, 为了问题的简化, 我们假设水下机器人关于3个平面对称, 我们可以得到以下简化的系数矩阵:

$$M_v = \text{diag}[m - X_{\dot{u}}, m - Y_{\dot{v}}, m - Z_{\dot{w}}, I_x - K_{\dot{p}}, I_y - M_{\dot{q}}, I_z - N_{\dot{r}}]. \quad (29)$$

D_v 为粘性水动力系数矩阵, $X_{\dot{u}}$ 等为惯性水动力系数。通过简化可以得到以下几个重要关系式:

$$\dot{M} = 0, \quad C_v + C_v^T = 0, \quad D_v = f(u) > 0. \quad (30)$$

5.2 水下机器人运动控制方法(Control method of AUV)

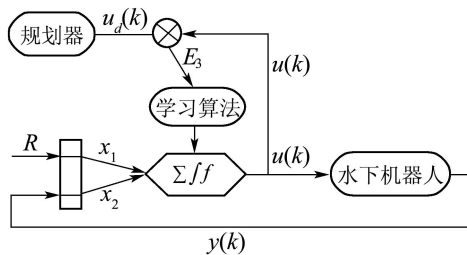


图3 过程神经元控制结构图

Fig. 3 Configuration of the process neuron control

水下机器人过程神经元控制结构图3所示。图中 $R, y(k)$ 为机器人目标位置和当前位置; $u(k)$ 为控制器实际输出值; x_1 和 x_2 为控制器的输入。至此, 水下机器人的各自由度过程神经网络的学习算法可总结为:

Step 1 初始化网络结构参数 w_i^l, θ , 动量学习因子 α, β , 学习率 μ, η , 给定机器人网络学习误差精度 ϵ ;

Step 2 通过传感器获得当前机器人位姿信息, 计算控制器的输入—位置偏差和偏差变化率。用基函数 $b_l(t)$ 将网络的输入函数及相应连接权函数展开, 其中 $l = 1, \dots, L, L$ 为第2层过程神经元个数;

Step 3 计算水下机器人该自由度运动规划器的预先规划 u_d 和对应控制器的当前输出 u ;

Step 4 根据式(6)计算网络误差函数 J , 如果 $J < \epsilon$ 则转控制器输出 $u = 0$;

Step 5 根据规则(7)和(8), 调整网络待训练参数转Step 2。

5.3 结果与分析(Results and analysis)

仿真试验以机器人的北向和垂向两个自由度复合运动为控制对象。仿真时间步长为0.05 s, 将机器人各自由度偏差和偏差变化率序列中的连续10个数据进行拟合, 即 $T = 0.5$ s, 构成一个时变函数作为网络的输入, 参考输入分别为北向30 m, 垂向10 m, 系统初态均为0, 海流速度为零。

将网络的输入函数及相应得连接权函数用如下正交基函数展开, 其中 $\xi_l = l/T$, 基函数的个数为10。学习误差精度设定为 $\epsilon = 0.002$, 学习率均为0.2, 动量学习因子均为0.3。

$$b_l(t) = \sin(\xi_l \pi t). \quad (31)$$

图4可以看出, 基于S模型的过程神经元控制器能够获得优异的控制结果。在收敛速度, 超调、稳态误差方面均比较令人满意。水下机器人常常工作在未知的复杂的海洋环境下, 受到各种干扰因素的影响。复杂的海洋环境有风、浪和流。为了说明在外界干扰的条件下, 该控制算法仍可较好对机器人实施控制。在不改变控制器任何参数的前提下, 分别模拟了波浪(式(32), (33))和顺流(北向 $V_c = 0.1$ m/s)两种海况下的机器人运动控制。鉴于水下机器人的定位控制是最有代表性的工作状态, 选取机器人深度和北向两个自由度进行抗干扰仿真试验

$$\text{North}U_u(t) = 0.1\sin(t\pi/5.5)\cos(t\pi/5), \quad (32)$$

$$\text{Depth}U_w(t) = 0.2\cos(t\pi/2.5)\cos(t\pi/3). \quad (33)$$

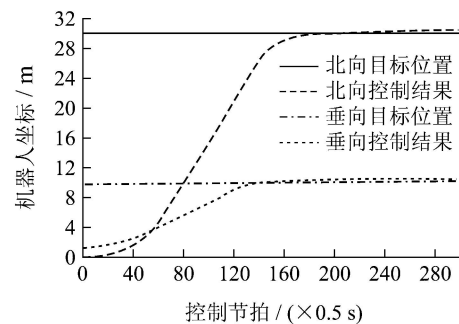


图4 复合运动控制仿真结果

Fig. 4 Simulation result of complex motion control

图5、图6是模拟了在波浪作用下控制的结果和北向控制器输出。图7、图8是模拟了在海流作用下控制的结果和北向控制器输出。可以看出基于S模型的过程神经元控制器具有很强的鲁棒性, 即使存在波浪等外界的干扰的影响, 也能很好的控制住水下机器人。虽然运动在目标位置存在一定的振荡, 但是振荡幅度和控制精度都可以满足实际工程上的应用。

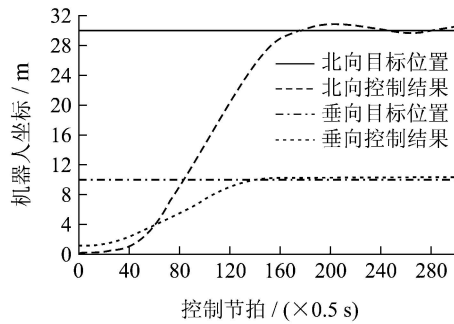


图5 增加干扰后复合运动控制仿真结果

Fig. 5 Simulation result of complex motion control by adding disturbance

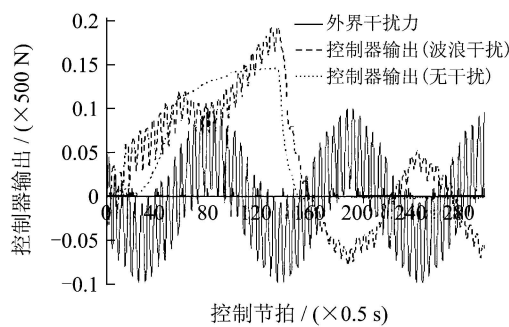


图6 增加波浪干扰下的北向控制器输出

Fig. 6 Output of north control system by adding wave

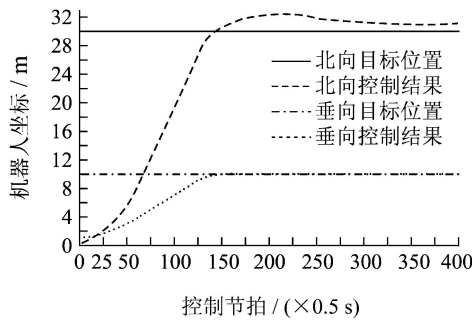


图7 增加北向流下复合运动控制仿真结果

Fig. 7 Simulation result of complex motion control by adding north current

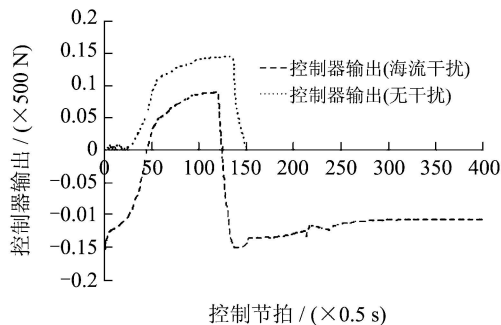


图8 增加北向流下的北向控制器输出

Fig. 8 Output of north control system by adding north current

6 结论(Conclusion)

在水下机器人运动控制过程中尝试引入过程神经算法, 过程神经网络从生物学角度出发更符合生物神经元的运行机制, 可以解决大量与时间有关的过程输入问题. 在具有模糊逻辑的S函数基础上, 同时引入系统输出的预先规划思想, 从而实现了简易的过程神经元控制器. 由于该模型提出不久, 又很多问题有待完善和优化.

参考文献(References):

- [1] 何新贵, 梁久祯. 过程神经网络的若干理论问题[J]. 中国工程科学, 2000, 2(12): 40-44.
(HE Xingui, LIANG Jiuzhen. Some theoretical issues on procedure-neural networks[J]. *Engineering Science*, 2000, 2(12): 40-44.)
- [2] 何新贵, 梁久祯, 许少华. 过程神经网络的训练及其应用[J]. 中国工程科学, 2001, 3(4): 31-35.
(HE Xingui, LIANG Jiuzhen, XU Shaohua. Learning and application of procedure neural networks[J]. *Engineering Science*, 2001, 3(14): 31-35.)
- [3] 俞建成, 李强, 张艾群. 水下机器人的神经网络自适应控制[J]. 控制理论与应用, 2008, 25(1): 9-13.
(YU Jiancheng, LI Qiang, ZHANG Aiqun. Neural network adaptive control for underwater vehicles[J]. *Control Theory & Applications*, 2008, 25(1): 9-13.)
- [4] LI Y, LIU J C, SHEN M X. Dynamics model of underwater robot motion in 6 degrees of freedom[J]. *Journal of Harbin Institute of Technology*, 2005, 12(4): 456-459.
- [5] 刘学敏, 徐玉如. 水下机器人运动的S面控制方法[J]. 海洋工程, 2001, 19(3): 81-84.
(LIU Xuemin, XU Yuru. S control automatic underwater vehicles[J]. *Ocean Engineering*, 2001, 19(3): 81-84.)
- [6] 许少华, 何新贵. 基于函数正交基展开的神经网络学习算法[J]. 计算机学报, 2004, 27(5): 645-650.
(XU Shaohua, HE Xingui. Learning algorithms of process neural networks based on orthogonal function basis expansion[J]. *Chinese Journal of Computers*, 2004, 27(5): 645-650.)
- [7] KAZUO I, TAMAKI U. An adaptive neural net controller system for an underwater vehicle[J]. *Control Engineering Practice*, 2000, 8(2): 177-184.

作者简介:

唐旭东 (1983—), 男, 博士, 目前研究方向为水下机器人智能控制与模式识别, E-mail: tangxudong@yahoo.cn;

庞永杰 (1955—), 男, 教授, 博士生导师, 目前研究方向为水下机器人设计与运动仿真;

李 晔 (1978—), 男, 讲师, 博士, 目前研究方向为水下机器人智能控制与运动仿真.