



(12) 发明专利申请

(10) 申请公布号 CN 105446821 A

(43) 申请公布日 2016. 03. 30

(21) 申请号 201510766653. 5

(22) 申请日 2015. 11. 11

(71) 申请人 哈尔滨工程大学

地址 150001 黑龙江省哈尔滨市南岗区南通大街 145 号哈尔滨工程大学科技处知识产权办公室

(72) 发明人 孙玉山 张国成 李岳明 张英浩 吴海波 张磊 张强 冉祥瑞 曹东东 付甘立

(74) 专利代理机构 哈尔滨市船大专利事务所 23201

代理人 张贵丰

(51) Int. Cl.

G06F 11/07(2006. 01)

G06N 3/04(2006. 01)

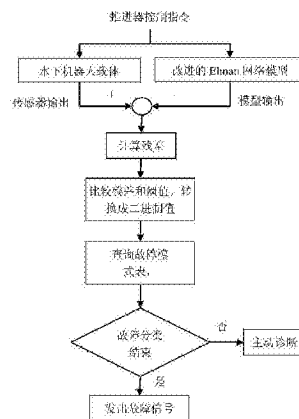
权利要求书2页 说明书9页 附图4页

(54) 发明名称

一种基于改进神经网络的智能水下机器人推进器故障诊断方法

(57) 摘要

本发明公开了一种基于改进神经网络的智能水下机器人推进器故障诊断方法。将水下机器人的推进器的控制指令分别传送给基于神经网络的水下机器人模型和实际推进器,将基于神经网络的水下机器人模型的输出值和水下机器人实际传感器的测量值进行作差,根据残差判断推进器是否故障;构建的神经网络水下机器人模型中,在联接单元中增加了固定增益的自反馈连接,并且采用动态反向传播学习算法对基于神经网络的水下机器人模型进行学习和训练,对神经网络层间权值进行调整。本发明能够提高推进器故障诊断的效率和准确性。



1. 一种基于改进神经网络的智能水下机器人推进器故障诊断方法,其特征在于:将水下机器人的推进器的控制指令分别传送给基于神经网络的水下机器人模型和 underwater 机器人实际推进器,将基于神经网络的水下机器人模型的输出值和 underwater 机器人实际传感器的测量值进行作差,根据残差判断推进器是否故障;

所述的基于神经网络的水下机器人模型为:

$$x(k) = f(W^1 x_c(k) + W^2 u(k-1) + W^4 y_c(k))$$

$$x_c(k) = x(k-1) + \alpha x_c(k-1)$$

$$y_c(k) = y(k-1) + \gamma y_c(k-1)$$

$$y(k) = g(W^3 x(k))$$

$u(k)$ 为 k 时刻输入层输入, $x(k)$ 为 k 时刻隐含层的输出, $x_c(k)$ 为 k 时刻联接层1的输出, $y_c(k)$ 为 k 时刻联接层2的输出, $y(k)$ 为 k 时刻输出, γ 为联接层2的自联结反馈增益因子, α 为联接层1的自联结反馈增益, W^1 为联接层1到隐层的联接权值, W^2 是输入层到隐层的联接权值, W^3 为联接层1到输出层的联接权值, W^4 为联接层2到隐层的联接权值, $g(\cdot)$ 为线性函数。

2. 根据权利要求1所述的一种基于改进神经网络的智能水下机器人推进器故障诊断方法,其特征在于:采用动态反向传播学习算法对基于神经网络的水下机器人模型进行学习和训练,对神经网络层间权值进行调整;神经网络层间修正权值为:

$$\Delta w_{ij}^3 = \eta_3 \delta_i^0 x_j(k), \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n$$

$$\Delta w_{jq}^2 = \eta_2 \delta_j^h u_q(k-1), \quad j = 1, 2, \dots, n; \quad q = 1, 2, \dots, r$$

$$\Delta w_{jl}^1 = \eta_1 \sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{jl}^1}, \quad j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n$$

$$\Delta w_{js}^4 = \eta_4 \sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{js}^4}$$

其中, η_1 为 W^1 的学习步长, η_2 为 W^2 的学习步长, η_3 为 W^3 的学习步长, η_4 为 W^4 的学习步长, $f(\cdot)$ 为 sigmoid 函数, $\delta_i^0 = (y_{d,i}(k) - y_i(k)) g_i'(\bullet)$, $\delta_j^h = \sum_{i=1}^m (\delta_i^0 w_{ij}^3) f_j'(\bullet)$,

$$\frac{\partial x_j(k)}{\partial w_{jl}^1} = f_j'(\bullet) x_j(k-1) + \alpha \frac{\partial x_j(k-1)}{\partial w_{jl}^1}, \quad \frac{\partial x_j(k)}{\partial w_{js}^4} = f_j'(\bullet) y_s(k-1) + \lambda \frac{\partial x_j(k-1)}{\partial w_{js}^4}.$$

3. 根据权利要求1所述的一种基于改进神经网络的智能水下机器人推进器故障诊断方法,其特征在于:所述的水下机器人配有8个推进器,包括:垂直面主推进器、水平面主推进器、垂向推进器及侧向推进器4组,每组推进器均由两个推进器组成;

基于神经网络的水下机器人模型的输入为:

$$u(k) = [\text{Thr1}(k) \text{ Thr2}(k) \text{ Thr3}(k) \text{ Thr4}(k) \text{ Thr5}(k) \text{ Thr6}(k) \text{ Thr7}(k) \text{ Thr8}(k)]^T$$

基于神经网络的水下机器人模型的输出为:

$$y(k) = [u(k) \ v(k) \ w(k) \ \text{roll}(k) \ \text{pitch}(k) \ \text{yaw}(k)]^T$$

$u, v, w, \text{roll}, \text{pitch}, \text{yaw}$ 分别为水下机器人的纵向速度、横向速度、垂向速度、横摇角、纵摇角、艏摇角, $\text{Thr1}(k), \text{Thr2}(k), \text{Thr3}(k), \text{Thr4}(k), \text{Thr5}(k), \text{Thr6}(k), \text{Thr7}(k), \text{Thr8}(k)$

为水下机器人的8个推力器的电压指令。

4. 根据权利要求1所述的一种基于改进神经网络的智能水下机器人推进器故障诊断方法,其特征在于:所述的根据残差判断推进器是否故障的具体方法为:

水平面主推出现故障时纵向速度、艏向角和横向速度各自的残差均会超过其阈值,左主推故障时艏向角的残差会偏向负,横向速度的残差偏向负;右主推故障时艏向角的残差和横向速度的残差偏向正;

垂直面主推故障时垂向速度、纵摇角和纵向速度各自的残差均会超过其阈值,并且上主推故障时纵摇角会的残差偏向正,下主推故障时纵摇角的残差偏向负;

侧向推进器故障时,横向速度和艏向角的残差同时超过其阈值,执行主动检测诊断,将艏侧推与尾侧推同时施加同样大小的正向推力,若艏向角残差为负则艏侧推故障;若艏向角残差为正则尾侧推故障;

垂向推进器故障时,深度、垂向速度和纵摇角的残差均超过其阈值,执行主动检测诊断,将艏垂推与尾垂推同时施加同样大小的正向推力,若纵摇角艏向角残差为负则艏侧推故障,若艏向角残差为正则尾侧推故障。

一种基于改进神经网络的智能水下机器人推进器故障诊断方法

技术领域

[0001] 本发明涉及水下机器人故障诊断技术领域,尤其涉及一种基于改进的神经网络智能水下机器人推进器的故障诊断方法。

发明内容

[0002] 水下机器人在世界范围内的应用领域不断扩大,如海洋研究、科学考察、海洋开发和水下工程等。水下机器人一般工作在未知复杂的海洋环境下,一旦发生事故就可能造成巨大的财产损失,这就要求水下机器人具有应付突发事件的能力,即自动故障诊断和实现容错控制的能力。实现水下机器人控制系统故障自主诊断也是其智能化的重要体现。

[0003] 神经网络技术的出现为解决故障诊断问题提供了一种新的思路与方法,特别是对于非线性系统和复杂系统,基于解析模型的故障诊断方法面临着难以建立数学模型的实际困难,而基于知识的故障诊断方法成了重要的、也是切实可行的方法。神经网络的I/O非线性映射特性、信息的分布存储、并行处理和全局集体作用,特别是其高度的自组织和自学习能力,使其成为故障诊断的一种有效方法和手段,在许多实际系统中得到了成功的应用。

[0004] 对于水下机器人来说,由于其运动是多自由度的,而且运动特性具有很强的非线性,同时在开发时设计变更和实用时有效载荷的频繁变化使形状和重量平衡发生变化,运动特性也相应变化,对于这种动态的非线性系统,传统的辨识方法有很大的局限性,由于神经网络具有以任意精度逼近任意非线性映射的能力;固有的学习能力降低了不确定性,增加了适应环境变化的泛化能力;分布式信息存储和处理结构,使之具有的容错能力等等性质,采用动态的神经网络,通过一定的学习算法对运动状态数据进行学习辨识出水下机器人的运动特性,即用神经网络来建立水下机器人的运动模型,可以作为自适应控制器的辨识器,可以作为虚拟传感器,可以为故障诊断提供信息等。

[0005] 2007年1月《系统仿真学报》第19卷第1期的“基于小波神经网络的水下机器人执行器故障诊断”和2008年6月第49卷第2期的“基于小波神经网络的水下机器人推进器故障诊断”分别提出了一种改进的小波神经网络,并用于水下机器人的推进器的故障诊断。

[0006] 上述文献均是采用小波网络进行水下机器人推进器的故障诊断,但是利用小波网络进行故障诊断小波基选择是关键,如何选择小波基目前还没有一个理论的标准,大多依赖于经验。由于小波基不具有唯一性,且小波基是不规则的,不同小波基形状差别很大,支撑范围和规则性都有很大的差别。因而,对同一信号选用不同的小波基进行信号处理,往往得到的结果差别较大,必然影响最终的处理结果。遗憾的是,作为重要的小波基如何选择及选择依据上述文献并未给出。另外,上述文献虽然给出了小波网络的原理与方法,但故障诊断流程及如何诊断并未给出详细的原则与依据。本发明中不仅详细描述了改进的Elman神经网络的构造与分析,并给出了智能水下机器人推进器故障诊断流程及故障诊断的原则与依据。

发明内容

[0007]

[0008] 本发明的目的是提供一种推进器故障诊断效率高、准确性高的,基于改进神经网络智能水下机器人推进器故障诊断方法。

[0009] 一种基于改进神经网络智能水下机器人推进器故障诊断方法,将水下机器人的推进器的控制指令分别传送给基于神经网络的水下机器人模型和水下机器人实际推进器,将基于神经网络的水下机器人模型的输出值和水下机器人实际传感器的测量值进行作差,根据残差判断推进器是否故障;

[0010] 所述的基于神经网络的水下机器人模型为:

$$[0011] \quad x(k) = f(W^1 x_c(k) + W^2 u(k-1) + W^4 y_c(k))$$

$$[0012] \quad x_c(k) = x(k-1) + \alpha x_c(k-1)$$

$$[0013] \quad y_c(k) = y(k-1) + \gamma y_c(k-1)$$

$$[0014] \quad y(k) = g(W^3 x(k))$$

[0015] $u(k)$ 为 k 时刻输入层输入, $x(k)$ 为 k 时刻隐含层的输出, $x_c(k)$ 为 k 时刻联接层1的输出, $y_c(k)$ 为 k 时刻联接层2的输出, $y(k)$ 为 k 时刻输出, γ 为联接层2的自联结反馈增益因子, α 为联接层1的自联结反馈增益, W^1 为联接层1到隐层的联接权值, W^2 是输入层到隐层的联接权值, W^3 为联接层1到输出层的联接权值, W^4 为联接层2到隐层的联接权值, $g(\cdot)$ 为线性函数。

[0016] 本发明一种基于改进神经网络智能水下机器人推进器故障诊断方法,还可以包括:

[0017] 1、采用动态反向传播学习算法对基于神经网络的水下机器人模型进行学习和训练,对神经网络层间权值进行调整;神经网络层间修正权值为:

$$[0018] \quad \Delta w_{ij}^3 = \eta_3 \delta_i^0 x_j(k), \quad i=1,2,\dots,m; \quad j=1,2,\dots,n$$

$$[0019] \quad \Delta w_{jq}^2 = \eta_2 \delta_j^h u_q(k-1), \quad j=1,2,\dots,n; \quad q=1,2,\dots,r$$

$$[0020] \quad \Delta w_{jl}^1 = \eta_1 \sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{jl}^1}, \quad j=1,2,\dots,n; \quad l=1,2,\dots,n$$

$$[0021] \quad \Delta w_{js}^4 = \eta_4 \sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{js}^4}$$

[0022] 其中, η_1 为 W^1 的学习步长, η_2 为 W^2 的学习步长, η_3 为 W^3 的学习步长, η_4 为 W^4 的学习步长, $f(\cdot)$ 为sigmoid函数, $\delta_i^0 = (y_{a,i}(k) - y_i(k))g'_i(\bullet)$, $\delta_j^h = \sum_{i=1}^m (\delta_i^0 w_{ij}^3) f'_j(\bullet)$,

$$\frac{\partial x_j(k)}{\partial w_{jl}^1} = f'_j(\bullet) x_l(k-1) + \alpha \frac{\partial x_j(k-1)}{\partial w_{jl}^1}, \quad \frac{\partial x_j(k)}{\partial w_{js}^4} = f'_j(\bullet) y_s(k-1) + \lambda \frac{\partial x_j(k-1)}{\partial w_{js}^4}。$$

[0023] 2、水下机器人配有8个推进器,包括:垂直面主推进器、水平面主推进器、垂向推进器及侧向推进器4组,每组推进器均由两个推进器组成;

[0024] 基于神经网络的水下机器人模型的输入为:

[0025] $u(k)=[\text{Thr1}(k) \text{ Thr2}(k) \text{ Thr3}(k) \text{ Thr4}(k) \text{ Thr5}(k) \text{ Thr6}(k) \text{ Thr7}(k) \text{ Thr8}(k)]^T$

[0026] 基于神经网络的水下机器人模型的输出为:

[0027] $y(k)=[u(k) \text{ v}(k) \text{ w}(k) \text{ roll}(k) \text{ pitch}(k) \text{ yaw}(k)]^T$

[0028] $u, v, w, \text{roll}, \text{pitch}, \text{yaw}$ 分别为水下机器人的纵向速度、横向速度、垂向速度、横摇角、纵摇角、艏摇角, $\text{Thr1}(k), \text{Thr2}(k), \text{Thr3}(k), \text{Thr4}(k), \text{Thr5}(k), \text{Thr6}(k), \text{Thr7}(k), \text{Thr8}(k)$ 为水下机器人的8个推力器的电压指令。

[0029] 3、根据残差判断推进器是否故障的具体方法为:

[0030] 水平面主推出现故障时纵向速度、艏向角和横向速度各自的残差均会超过其阈值,左主推故障时艏向角的残差会偏向负,横向速度的残差偏向负;右主推故障时艏向角的残差和横向速度的残差偏向正;

[0031] 垂直面主推故障时垂向速度、纵摇角和纵向速度各自的残差均会超过其阈值,并且上主推故障时纵摇角会的残差偏向正,下主推故障时纵摇角的残差偏向负;

[0032] 侧向推进器故障时,横向速度和艏向角的残差同时超过其阈值,执行主动检测诊断,将艏侧推与尾侧推同时施加同样大小的正向推力,若艏向角残差为负则艏侧推故障;若艏向角残差为正则尾侧推故障;

[0033] 垂向推进器故障时,深度、垂向速度和纵摇角的残差均超过其阈值,执行主动检测诊断,将艏垂推与尾垂推同时施加同样大小的正向推力,若纵摇角艏向角残差为负则艏侧推故障,若艏向角残差为正则尾侧推故障。

[0034] 有益效果:

[0035] Jordan网络具有输出反馈环节,可以反映系统的输出特性,但无法反映状态特性原型Jordan网络只有输出反馈,映射非线性动态系统的能力具有局限性,无法表示任意的动态系统,加入状态反馈可以使网络更具有动态性能。

[0036] 本发明通过设计固定增益的自反馈连接及增加了输出层节点的反馈,构造了一种神经网络,本发明构建的神经网络具有更快的学习收敛速度和泛化能力,能够反映系统的状态特性,对智能水下机器人这类高阶非线性系统有着更强的辨识能力,有助于对于水下机器人的建模与推进器的故障诊断效率和准确性的提高。

[0037] 针对水下机器人系统的不确定性建模困难的特点,提出的神经网络并应用于水下机器人的运动建模,通过比较水下机器人实际测量值与模型的输出(运动状态估计值)计算残差可以进行水下机器人推进器故障诊断。

[0038] 神经网络在联接单元中增加了固定增益的自反馈连接,改善了网络的状态反馈,提高了网络动态反映能力。

[0039] 神经网络在输出层引出反馈至输入层,可以反映系统输出特性。网络具有更快的学习收敛速度和泛化能力,对高阶非线性系统有着更强的辨识能力。

附图说明

[0040] 图1是本发明神经网络结构示意图;

[0041] 图2是蓬莱海试时某次在水下1米目标搜索时轨迹曲线;

[0042] 图3是实际测量值与本发明神经网络的纵向速度辨识结果;

- [0043] 图4是实际测量值与本发明神经网络的横向速度辨识结果；
 [0044] 图5是实际测量值与本发明神经网络的艏向角辨识结果；
 [0045] 图6是基于改进的Elman神经网络的推进器故障诊断原理；
 [0046] 图7是水下机器人故障诊断规则表。

具体实施方式

- [0047] 下面将结合附图对本发明做进一步详细说明。
 [0048] 本发明的目的在于提供推进器故障诊断方法,基于改进神经网络的智能水下机器人推进器故障诊断方法。

[0049] 本发明所采用的技术方案是:

[0050] 步骤1:构建神经网络模型;

[0051] 构建的神经网络描述的非线性状态空间表达式:

$$[0052] \quad x(k) = f(W^1 x_c(k) + W^2 u(k-1) + W^4 y_c(k))$$

$$[0053] \quad x_c(k) = x(k-1) + \alpha x_c(k-1)$$

$$[0054] \quad y_c(k) = y(k-1) + \gamma y_c(k-1)$$

$$[0055] \quad y(k) = g(W^3 x(k))$$

[0056] $u(k)$ 为 k 时刻输入层输入, $x(k)$ 为 k 时刻隐含层的输出, $x_c(k)$ 、 $y_c(k)$ 为 k 时刻联接层的输出, $y(k)$ 为 k 时刻输出, γ 为自联结反馈增益因子, W^3 为联接层1到输出层的联接权值, $g(\cdot)$ 为线性函数,即 $y(k) = g(W^3 x(k)) = W^3 x(k)$;

[0057] 步骤2:采用动态反向传播学习算法进行神经网络的学习、训练,层间权值的调整;

[0058] 神经网络的动态反向传播学习算法可归纳如下:

$$[0059] \quad \Delta w_{ij}^3 = \eta_3 \delta_i^0 x_j(k) \quad , \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n$$

$$[0060] \quad \Delta w_{jq}^2 = \eta_2 \delta_j^h u_q(k-1) \quad , \quad j = 1, 2, \dots, n; \quad q = 1, 2, \dots, r$$

$$[0061] \quad \Delta w_{jl}^1 = \eta_1 \sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{jl}^1} \quad , \quad j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n$$

$$[0062] \quad \Delta w_{js}^4 = \eta_4 \sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{js}^4}$$

[0063] Δw_{jl}^1 、 Δw_{jq}^2 、 Δw_{ij}^3 、 Δw_{js}^4 为修正权值;其中 η_1 、 η_2 、 η_3 、 η_4 分别为 W^1 、 W^2 、 W^3 、 W^4 的学习

步长; $\delta_i^0 = (y_{d,i}(k) - y_i(k))g_i'(\bullet)$, $\delta_j^h = \sum_{i=1}^m (\delta_i^0 w_{ij}^3) f_j'(\bullet)$

$$[0064] \quad \frac{\partial x_j(k)}{\partial w_{jl}^1} = f_j'(\bullet) x_l(k-1) + \alpha \frac{\partial x_j(k-1)}{\partial w_{jl}^1} \quad , \quad \frac{\partial x_j(k)}{\partial w_{js}^4} = f_j'(\bullet) y_s(k-1) + \lambda \frac{\partial x_j(k-1)}{\partial w_{js}^4}$$

[0065] $\frac{\partial x_j(k)}{\partial w_{jl}^1}$ 为偏导;

[0066] 步骤3:基于神经网络的水下机器人运动建模与辨识;

[0067] 将步骤1的神经网络应用于水下机器人的建模,将水下机器人的推力器的电压指

令作为网络输入,应用步骤2的神经网络的动态反向传播学习算法得到网络输出,将网络输出与实际标准模型进行比对,得出是否故障结论。

[0068] 如图1和图4所示,本发明包括以下几个步骤:

[0069] 步骤1:构建神经网络模型;

[0070] 本发明如图1所示为一种基于改进神经网络的水下机器人运动状态模型结构。 W^1, W^2, W^3, W^4 分别为联接层1到隐层、输入层到隐层、隐层到输出层、联接层2到隐层的联接权值, Z^{-1} 为通用符号,代表反馈;1为表示反馈的关系(比如1倍的前一时刻 $x(k-1)$ 的1倍,例如 $x_c(k) = x(k-1) + \alpha x_c(k-1)$)。构建的神经网络联接单元 x_c 中,增加一个固定增益 α 的自反馈连接,联接单元 x_c 在 k 时刻的输出,将等于隐层在 $(k-1)$ 时刻的输出加上联接单元在 $(k-1)$ 输出的 α 倍,即 $x_{c,1}(k) = \alpha x_{c,j}(k-1) + x_1(k-1) \quad l=1, 2, \dots, n$,其中 $x_{c,1}(k)$ 和 $x_1(k-1)$ 分别表示第1个联接单元和第1个隐层单元的输出, α 为联接层1的自联结反馈增益。本发明构建的神经网络具有更快的学习收敛速度和泛化能力,对高阶非线性系统有着更强的辨识能力。本发明构建的神经网络描述的非线性状态空间表达式为:

$$[0071] \quad x(k) = f(W^1 x_c(k) + W^2 u(k-1) + W^4 y_c(k)) \quad (1)$$

$$[0072] \quad x_c(k) = x(k-1) + \alpha x_c(k-1) \quad (2)$$

$$[0073] \quad y_c(k) = y(k-1) + \gamma y_c(k-1) \quad (3)$$

$$[0074] \quad y(k) = g(W^3 x(k)) \quad (4)$$

[0075] $u(k)$ 为 k 时刻输入层输入, $x(k)$ 为 k 时刻隐含层的输出(为向量,包含多个隐含层单元), $x_c(k)$ 为 k 时刻联接层1的输出(为向量,包含多个联接层单元), $y_c(k)$ 为 k 时刻联接层2的输出(为向量,包含多个联接层单元), $y(k)$ 为 k 时刻输出, γ 为自联结反馈增益因子, W^3 是联接层1到输出层的联接权(为向量), $g(\cdot)$ 为线性函数,即 $y(k) = g(W^3 x(k)) = W^3 x(k)$ 。

[0076] 步骤2:采用动态反向传播学习算法进行神经网络的学习、训练,层间权值的调整;

[0077] 考虑如下总体误差目标函数:

$$[0078] \quad E = \sum_{p=1}^N E_p \quad (5)$$

$$[0079] \quad E_p = \frac{1}{2} (y_d(k) - y(k))^T (y_d(k) - y(k))$$

[0080] 对隐含层到输出层的联接权 W^3

[0081]

$$\frac{\partial E_p}{\partial w_{ij}^3} = -(y_{d,i}(k) - y_i(k)) \frac{\partial y_i(k)}{\partial w_{ij}^3} = -(y_{d,i}(k) - y_i(k)) g'_i(\bullet) x_j(k) \quad (6)$$

[0082] w_{ij}^3 为隐含层第 i 单元到输出层第 j 单元的联接权值,

[0083] $y_{d,i}(k)$ 为 i 单元 k 时刻实际输出,

[0084] $f(\cdot)$ 为sigmoid函数,即 $f(x) = \frac{1}{1 + e^{-x}}$, $g(\cdot)$ 为线性函数,即 $y(k) = g(W^3 x(k)) =$

$W^3 x(k)$

[0085] 令 $\delta_i^o = (y_{d,i}(k) - y_i(k)) g'_i(\bullet)$,则

$$[0086] \quad \frac{\partial E_p}{\partial w_{ij}^3} = -\delta_i^0 x_j(k) \quad i=1,2,\dots,m; \quad j=1,2,\dots,n \quad (7)$$

[0087] $g'(\cdot)$ 为 $g(\cdot)$ 的偏导, $x_j(k)$ 为 k 时刻 j 联接单元, $y_i(k)$ 为 k 时刻 i 输出单元, δ_i^0 无具体含义, 有如下关系 $\delta_i^0 = (y_{a,i}(k) - y_i(k))g'_i(\cdot)$

[0088] 对输入层到隐含层的连接权 W^2

$$[0089] \quad \frac{\partial E_p}{\partial w_{jq}^2} = \frac{\partial E_p}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial x_j(k)} \frac{\partial x_j(k)}{\partial w_{jq}^2} \quad (8)$$

$$= \sum_{i=1}^m (-\delta_i^0 w_{ij}^3) f'_j(\cdot) u_q(k-1)$$

[0090] $f'(\cdot)$ 为 $f(\cdot)$ 的偏导, $u_q(k-1)$ 为 $k-1$ 时刻 q 输入单元,

[0091] w_{jq}^2 为输入层 j 单元到隐层 q 的联接权值,

[0092] 同样令 $\delta_j^h = \sum_{i=1}^m (\delta_i^0 w_{ij}^3) f'_j(\cdot)$, 则有

$$[0093] \quad \frac{\partial E_p}{\partial w_{jq}^2} = -\delta_j^h u_q(k-1) \quad j=1,2,\dots,n; \quad q=1,2,\dots,r \quad (9)$$

[0094] 类似对结构层单元到隐含层的连接权 W^1 , 则有

[0095]

$$\frac{\partial E_p}{\partial w_{jl}^1} = \frac{\partial E_p}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial x_j(k)} \frac{\partial x_j(k)}{\partial w_{jl}^1} = -\sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{jl}^1} \quad j=1,2,\dots,n; \quad l=1,2,\dots,n; \quad (10)$$

[0096] 忽略 $x_c(k)$ 与连接权 w_{jl}^1 之间的依赖关系, 故

[0097] $\partial x_j(k) / \partial w_{jl}^1 = f'_j(\cdot) x_{c,l}(k)$ 得 $f'_j(\cdot) x_{c,l}(k) = f'_j(\cdot) x_{l1}(k-1) + \alpha f'_j(\cdot) x_{c,l}(k-1)$

因而有

$$[0098] \quad \frac{\partial x_j(k)}{\partial w_{jl}^1} = f'_j(\cdot) x_{l1}(k-1) + \alpha \frac{\partial x_j(k-1)}{\partial w_{jl}^1}$$

[0099] 上式实际构成了梯度 $\frac{\partial x_j(k)}{\partial w_{jl}^1}$ 的动态递推关系。

[0100] 同理, 对于 W^4 有:

[0101]

$$\frac{\partial E_p}{\partial w_{js}^4} = \frac{\partial E_p}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial x_j(k)} \frac{\partial x_j(k)}{\partial w_{js}^4} = -\sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{js}^4}, \quad j=1,2,\dots,n; \quad s=1,2,\dots,m \quad (11)$$

[0102]

$$\frac{\partial x_j(k)}{\partial w_{js}^4} = f'_j(\cdot) y_s(k-1) + \lambda \frac{\partial x_j(k-1)}{\partial w_{js}^4}, \quad j=1,2,\dots,n; \quad s=1,2,\dots,m \quad (12)$$

[0103] w_{js}^4 联接层2第j单元到隐层第s单元的联接权值；

[0104] $y_s(k-1)$ 为输出层k-1时刻输出，下标s代表第s单元

[0105] 由于

$$[0106] \quad \Delta W_{ij} = -\eta \frac{\partial E_p}{\partial w_{ij}}$$

[0107] η 为学习步长；

[0108] 故神经网络的动态反向传播学习算法可归纳如下：

$$[0109] \quad \Delta w_{ij}^3 = \eta_3 \delta_i^0 x_j(k) \quad , \quad i=1,2,\dots,m; \quad j=1,2,\dots,n \quad (13)$$

$$[0110] \quad \Delta w_{jq}^2 = \eta_2 \delta_j^h u_q(k-1) \quad , \quad j=1,2,\dots,n; \quad q=1,2,\dots,r \quad (14)$$

$$[0111] \quad \Delta w_{jl}^1 = \eta_1 \sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{jl}^1} \quad , \quad j=1,2,\dots,n; \quad l=1,2,\dots,n \quad (15)$$

$$[0112] \quad \Delta w_{js}^4 = \eta_4 \sum_{i=1}^m (\delta_i^0 w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{js}^4} \quad (16)$$

[0113] Δw_{ij}^3 、 Δw_{jq}^2 、 Δw_{jl}^1 、 Δw_{js}^4 为修正权值。

[0114] 其中 $\eta_1, \eta_2, \eta_3, \eta_4$ 分别为 W^1, W^2, W^3, W^4 的学习步长。

$$[0115] \quad \delta_i^0 = (y_{d,i}(k) - y_i(k)) g_i'(\bullet), \quad \delta_j^h = \sum_{i=1}^m (\delta_i^0 w_{ij}^3) f_j'(\bullet)$$

$$[0116] \quad \frac{\partial x_j(k)}{\partial w_{jl}^1} = f_j'(\bullet) x_l(k-1) + \alpha \frac{\partial x_j(k-1)}{\partial w_{jl}^1}, \quad \frac{\partial x_j(k)}{\partial w_{js}^4} = f_j'(\bullet) y_s(k-1) + \lambda \frac{\partial x_j(k-1)}{\partial w_{js}^4}$$

[0117] $\frac{\partial x_j(k)}{\partial w_{jl}^1}$ 为偏导，隐层共m个单元(公式7)，h代表隐层(英文hide)。

[0118] 步骤3: 基于神经网络的水下机器人运动建模与辨识；

[0119] 将步骤1的神经网络应用于水下机器人的建模，将水下机器人的推力器的电压指令作为网络输入，应用步骤2的神经网络的动态反向传播学习算法得到网络输出，将网络输出与实际标准模型进行比对，得出是否故障结论。

[0120] 例如水下机器人配有8个推进器。根据推进器施力功能可以分成垂直面主推进器、水平面主推进器、垂向推进器及侧向推进器4组，每组推进器均由两个推进器组成。主推进器采用的是导管推力器，其中水平面主推进器的最大推力可达600N，其轴线与机器人的纵中轴线成13度角，垂直面主推进器的最大推力为210N，其轴线与机器人的纵中轴线成26度角。垂向推进器和侧向推进器则均采用槽道推力器。由于随着AUV纵向前进速度的增加，槽道推力器推力减额现象越来越严重，因此，为了节约能源，AUV在高速航行时关闭4个槽道推力器。AUV水下机器人装有8个推力器(两个水平面布置的主推进器，两个垂直面布置的主推进器，艏艉各一个侧向和垂向推力器)。水下机器人配置有测量3个线速度的多普勒测速仪、测量3个角度的罗经和8个推力器，根据水下机器人实际配置的传感器和执行器，设定网络输入输出如下：

[0121] $u(k)=[\text{Thr1}(k) \text{ Thr2}(k) \text{ Thr3}(k) \text{ Thr4}(k) \text{ Thr5}(k) \text{ Thr6}(k) \text{ Thr7}(k) \text{ Thr8}(k)]^T$

[0122] $y(k)=[u(k) \text{ v}(k) \text{ w}(k) \text{ roll}(k) \text{ pitch}(k) \text{ yaw}(k)]^T$

[0123] $u, v, w, \text{roll}, \text{pitch}, \text{yaw}$ ——水下机器人的纵向速度、横向速度、垂向速度、横摇角、纵摇角、摇艏角；

[0124] $\text{Thr1}, \text{Thr2}, \dots, \text{Thr8}$ ——水下机器人的8个推力器的电压指令。

[0125] 隐含层节点数取为28,学习率 $\eta_1=\eta_2=\eta_3=\eta_4=0.02$,因子 λ 取0.5。

[0126] 图2为在蓬莱海试时某次在水下1米目标搜索时轨迹曲线。图3为实际测量值与本发明神经网络的纵向速度辨识结果,图4为实际测量值与本发明神经网络的横向速度辨识结果,图5为实际测量值与本发明神经网络的艏向角辨识结果。

[0127] 神经网络建立水下机器人模型,此模型就代表水下机器人,若对此模型有一个输入(为推进器电压)就应该有水下机器人状态(艏向角、速度等信息)。实际故障诊断在水下机器人操作时,有一个输入同时给机器人载体和模型,比较二者输出差别来判别推进器故障情况。根据图6的辨识步骤,比如超过某阈值就代表有故障,发出故障信号。智能水下机器人故障诊断规则如图7所示,诊断水平面主推故障时选取航艏向角等残差信息融合判断,水平主推出现故障时纵向速度、艏向角和横向速度的残差均会超过其阈值,但左主推故障时艏向角会偏向负,横向速度偏向负;右主推故障时艏向角和横向速度偏向正,依此判断是左主推还是右主推发生故障;诊断垂直面主推故障时以垂向速度、纵摇角和纵向速度等残差信息融合判断,当垂直面上的主推出现故障时三二者的残差均会超过其阈值,但上主推故障时纵摇角会偏向正,下主推故障时纵摇角偏向负,依此判断是上主推故障还是下主推发生故障。侧向推进器故障时,横向速度和艏向角的残差同时超过其阈值,执行主动检测诊断,将艏侧推与尾侧推同时施加同样大小的正向推力,若艏向角残差为负则艏侧推故障,反之,若艏向角残差为正则尾侧推故障。垂向推进器故障时,深度、垂向速度和纵摇角的残差均超过其阈值,执行主动检测诊断,将艏垂推与尾垂推同时施加同样大小的正向推力(向下),若纵摇角艏向角残差为负则艏侧推故障,反之,若艏向角残差为正则尾侧推故障。

[0128] 仿真试验

[0129] 我们将神经网络用于智能水下机器人推进器的故障检测与诊断,采用 $8 \times 28 \times 6$ 的网络结构,网络的输入输出为3节所述,学习率取 $\eta_1=\eta_2=\eta_3=\eta_4=0.02$,动量因子 λ 取0.5。在系统投入使用之前对神经网络进行匀速直航、转艏、定点直航等数据的训练。从图3至图5可以看出该网络模型能很好地模拟水下机器人运动。我们通过比较改进Elman神经网络的输出与实际水下机器人状态的测量值得到6组残差。因残差中包含大量的水下机器人的故障信息,我们通过分析残差,便可以推断水下机器人推进器是否出现故障及具体哪一个推进器出现故障。为了尽量减小环境噪声对残差的影响,本文采用如下的方式来分析残差信息:在一个固定的时间间隔内记录一组残差,去掉一个最大值和一个最小值,然后求剩下的残差的平均值,如果残差超过了设定的阈值(阈值以及时间间隔的长短是通过大量的试验和经验事先设定的),就认为出现故障。

[0130] 相关的仿真试验,给出了左主推进器出现故障时的仿真试验结果。在没有故障时神经网络的输出(估计值)与实际测量值是一致的,而当出现主推故障时纵向速度的估计值就会出现偏差,而且随时间积累,偏差会越来越大直至估计值达到一个稳定值,在稳定的航

速下存在一个固定偏差,而且摇艏角也会出现偏差并会越来越来大,当残差超过设定的阈值时,就可以判定出现故障。根据纵向速度残差可以判定主推发生故障,再根据摇艏角度残差的正负可以判定是左推故障。

[0131] 海上试验

[0132] 为了测试与验证相关性能指标,AUV系统进行了一系列海上试验,完成了运动控制、水下导航、故障诊断、海底地形地貌探测、水下运动目标跟踪等海上试验。实验过程为从静止到定速1.0米/秒直线航行,在机器人稳定航速后,我们在第160个节拍将右主推输出指令置为零来模拟其右主推故障。在没有故障时(0~159个节拍),机器人纵向速度残差信息逐渐趋于零;而当机器人出现右主推故障时,纵向速度残差就会出现偏差,并随着时间累积偏差逐渐增大,最后稳定在一固定偏差左右,艏向角残差则会不断增大,且艏向角向右艏偏航。在水下机器人从静止到指定航速匀速直航的控制调整阶段(0~53节拍),纵向速度残差虽已超过其阈值范围,但艏向角度残差在零均值附近,但当机器人故障发生后,二者都会超过其阈值,因此可判断AUV是处于控制调整阶段还是发生了故障,从而有效防止误判。

[0133] 本发明构建神经网络对水下机器人推进器故障诊断方法,构建神经网络模型;构建的神经网络联接单元中,增加一个固定增益 α 的自反馈连接,联接单元在 k 时刻的输出,将等于隐层在 $(k-1)$ 时刻的输出加上联接单元在 $(k-1)$ 输出的 α 倍,采用动态反向传播学习算法进行神经网络的学习、训练,层间权值的调整;将神经网络应用于水下机器人的建模,将水下机器人的推力器的电压指令作为网络输入,应用动态反向传播学习算法得到网络输出,将网络输出与实际标准模型进行比对,得出是否故障结论。本发明的有益效果是对于水下机器人的推进器的故障诊断效率高,准确性较高。

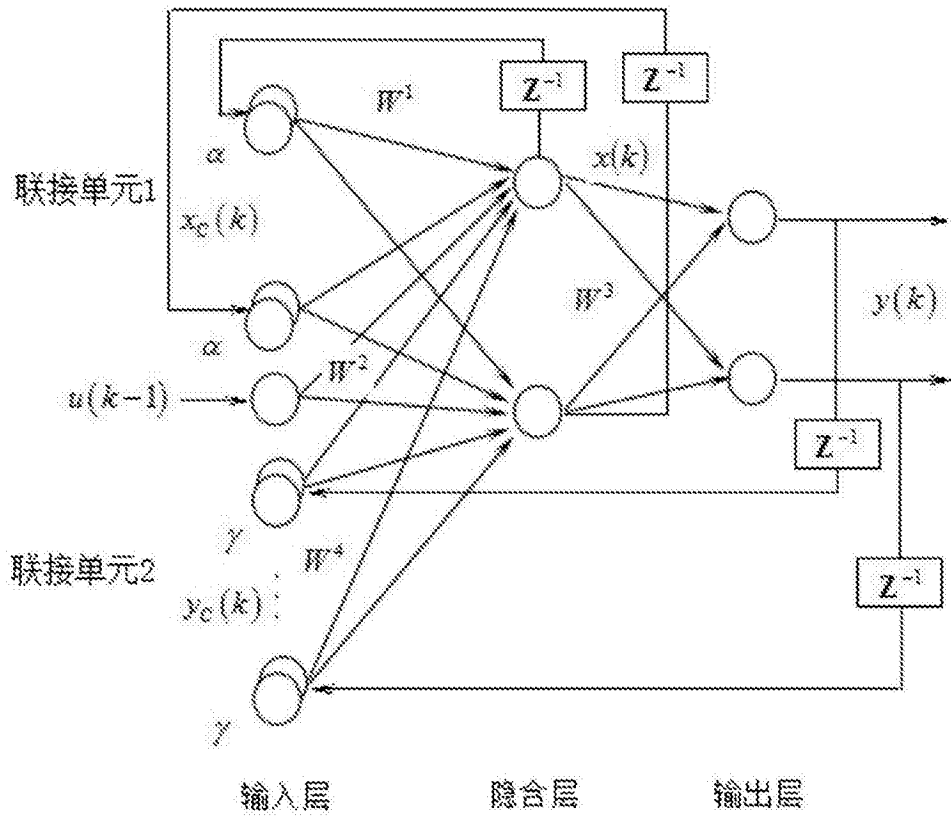


图1

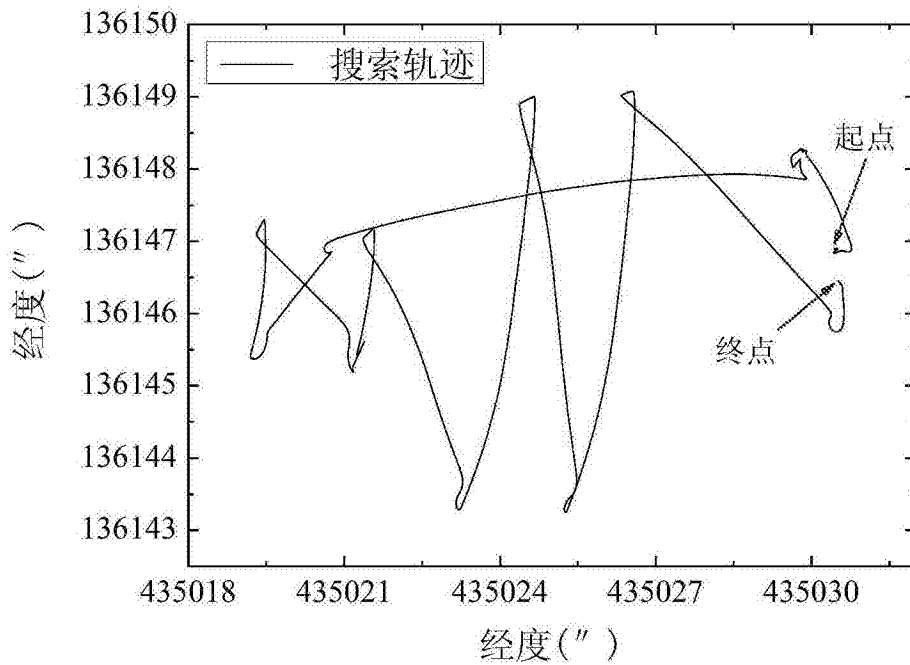


图2

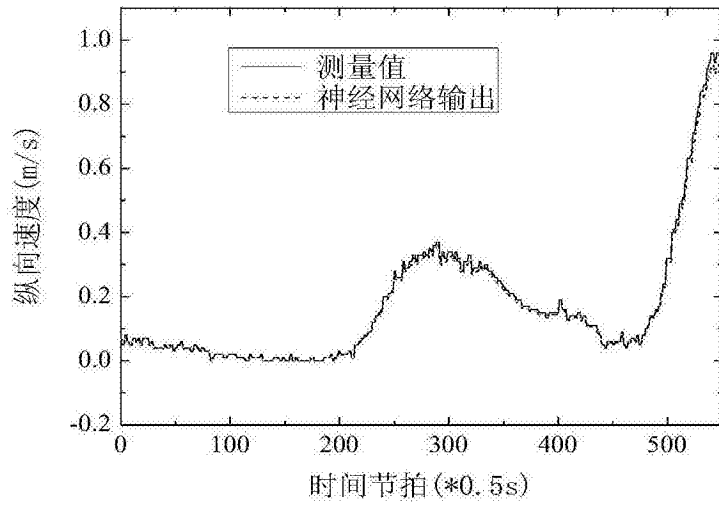


图3

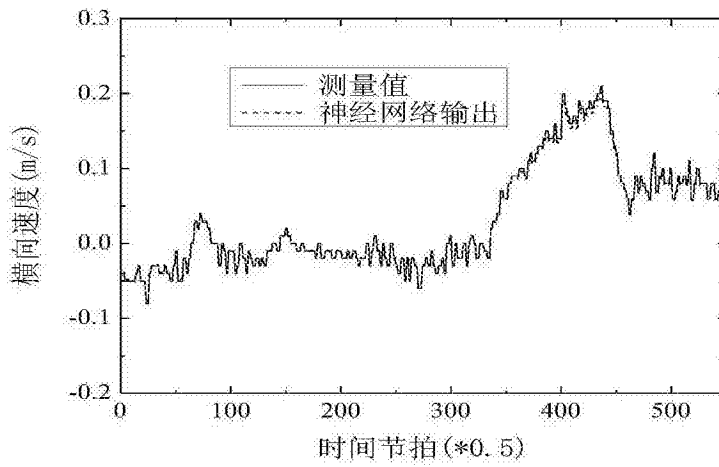


图4

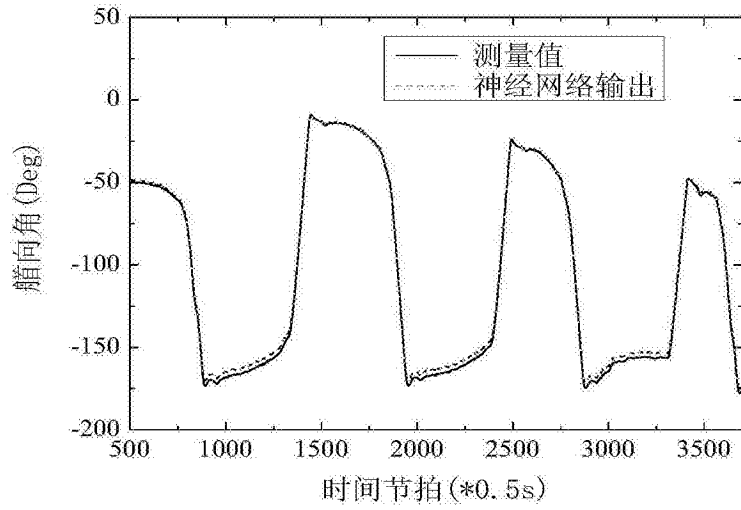


图5

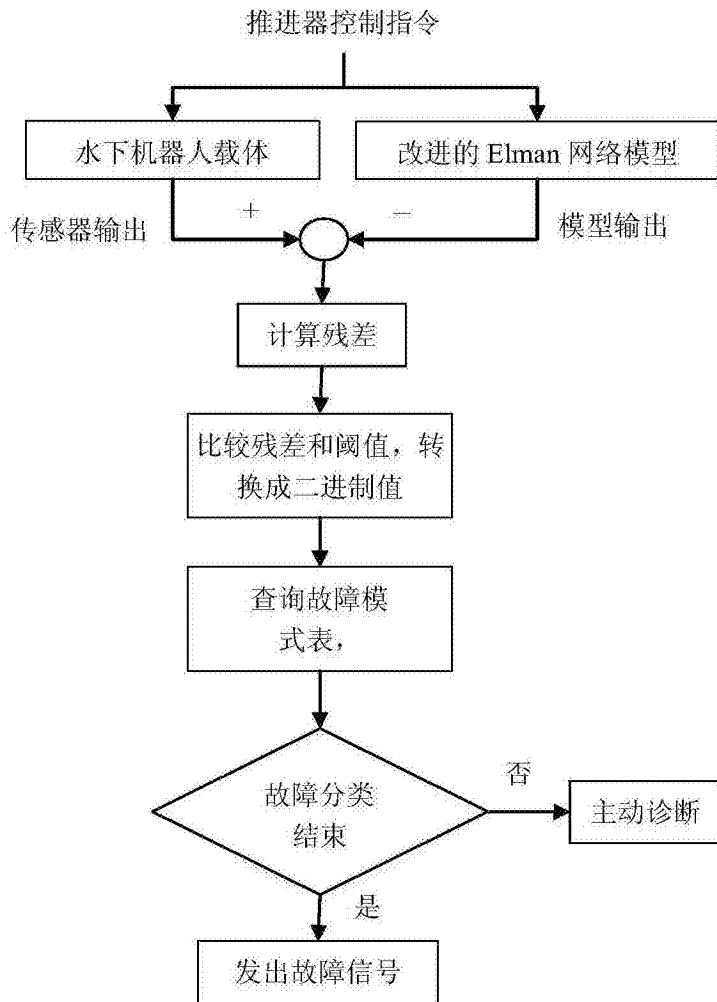


图6

故障模式	状态 0-正常, 1-故障						故障类型
	深度	纵向速度	横向速度	垂向速度	艏向角	纵摇角	
0	0	0	0	0	0	0	无
1		1					传感器(纵向速度)
2	0	1	1	0	1	0	左主推或右主推
3	1	1	0	1	0	1	上主推或下主推
4	0	0	1	0	1	0	侧向推进器
5	1	0	0	1	0	1	垂向推进器
6	1	0	0	1	0	0	传感器(垂向速度)
7	0	0	0	0	1	0	传感器(摇艏角)

图7