

# 免疫微粒群算法优化的水下机器人运动控制器

张磊<sup>1</sup>, 万磊<sup>1</sup>, 李晔<sup>1</sup>, 姚志广<sup>2</sup>

ZHANG Lei<sup>1</sup>, WAN Lei<sup>1</sup>, LI Ye<sup>1</sup>, YAO Zhi-guang<sup>2</sup>

1. 哈尔滨工程大学 船舶工程学院, 哈尔滨 150001

2. 中国石油集团工程技术研究院, 天津 300451

1. School of Naval Architecture, Harbin Engineering University, Harbin 150001, China

2. CNPC Research Institute of Engineering Technology, Tianjin 300451, China

E-mail: charlei08@163.com

**ZHANG Lei, WAN Lei, LI Ye, et al. Motion controller of autonomous underwater vehicle optimized by immune particle swarm. Computer Engineering and Applications, 2009, 45(28): 215-218.**

**Abstract:** Drawing ideas from concentration and fitness-based antibody-update strategy of immune algorithm, an Immune Particle Swarm Optimization (IPSO) algorithm is presented to enhance local searching ability, and it is analyzed about the convergency. IPSO tunes the acceleration factors dynamically according to particle's concentration and fitness. This guarantees the diversity and sustained searching ability of the swarm. In the comparison experiments with genetic algorithm, immune genetic algorithm, basic particle swarm optimization algorithm and so on, IPSO performs well with best solution and fastest convergence speed. And the control experiments conducted on AUV platform show that IPSO-based controller works well and has strong ability against current disturbance. The simulation results show feasibility of IPSO in application to AUV.

**Key words:** particle swarm optimization algorithm; immune algorithm; convergency; Autonomous Underwater Vehicle (AUV); fuzzy neural network

**摘要:** 借鉴免疫算法基于浓度和适应度的抗体更新策略, 提出一种改进微粒群局部搜索能力的免疫微粒群算法, 并对其进行收敛性分析。该算法在满足收敛性的条件下, 根据微粒浓度和适应度动态调整加速因子, 保证了群的多样性和持续搜索能力。在与遗传算法、免疫遗传算法、基本微粒群等算法的仿真比较试验中, 该算法不仅搜索到了最好的近优解, 而且收敛速度最快。在无人潜水器仿真平台上的控制试验表明, 基于免疫微粒群算法的控制器性能良好, 具有较强的抗海流干扰能力。仿真结果证明了该算法的可行性。

**关键词:** 微粒群算法; 免疫算法; 收敛性; 无人潜水器; 模糊神经网络

**DOI:** 10.3778/j.issn.1002-8331.2009.28.065 **文章编号:** 1002-8331(2009)28-0215-04 **文献标识码:** A **中图分类号:** TP24

## 1 引言

微粒群算法(Particle Swarm Optimization, PSO)是 Kennedy 和 Eberhart 于 1995 年提出的一种以鸟群运动行为为模型的通用优化算法。该算法结构简单, 容易实现, 基本不受问题峰数增加的影响, 受问题维数的影响也很小<sup>[1-2]</sup>, 因此成为参数寻优的常用工具。但该算法在搜索后期往往会因为微粒速度的降低而陷入局部极值。

模糊神经网络(Fuzzy Neural Network, FNN)因为兼具模糊系统和神经网络两者的优点, 在无人潜水器(Autonomous Underwater Vehicle, AUV)的基础运动控制中得到了广泛的应用<sup>[3]</sup>。由于 AUV 各自由度方向的运动能力不尽相同以及海洋环境的复杂性, 仅凭专家经验和先验知识手工确定控制器中隶属函数和模糊规则的参数将是一件非常困难和艰苦的工作, 而且难以保证所得参数是最优组合。

文献[4-6]分别利用遗传算法和免疫遗传算法对无人潜水器的控制器进行优化, 得到了较好的效果。但 FNN 控制器参数

相对较多, 使搜索空间的维数成倍增大, 因此上述算法很难保证其搜索速度及收敛性。文献[7]所提出的免疫粒子群混合优化算法(为方便表示, 将其记为 HPSO)在搜索过程中重新初始化部分微粒, 虽然可以提高微粒群的多样性, 但是无疑会使算法的搜索速度受到影响。该文提出的免疫微粒群算法借鉴免疫算法中基于浓度和适应度的抗体更新策略<sup>[8]</sup>, 根据群的环境动态调整算法的加速因子, 既保证了群的多样性, 避免陷入局部极值, 又不影响收敛速度。分别利用 GA、IGA、PSO、HPSO 和 IPSO 算法进行模糊神经网络基础运动控制器参数的自寻优设计, 并将所得的控制器在 AUV 仿真平台上进行试验验证。结果表明 IPSO 算法不仅具有更快的收敛速度, 而且寻优所得的参数组合更接近最优值。

## 2 模糊神经网络基础运动控制器

模糊神经网络基础运动控制器(Fuzzy Neural Network Motion Controller, FNNMC)采用 5 层的前向神经网络, 其结构

**作者简介:** 张磊(1983-), 男, 博士, 主要研究方向为无人潜水器智能控制与系统仿真。

**收稿日期:** 2008-05-30 **修回日期:** 2008-09-26

如图1所示。该FNNMC实质上为双输入单输出的模糊系统,其输入为位置偏差和速度,输出为作用于AUV各个自由度上所需要的力或力矩。

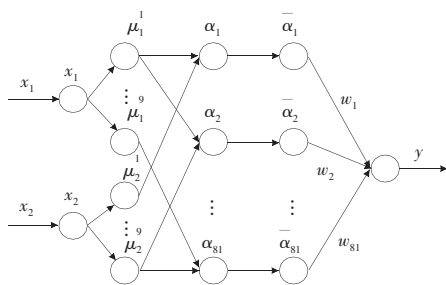


图1 模糊神经网络控制器结构图

图1中,第一层为输入层, $x_1=e$ , $x_2=\dot{e}$ 。第二层为模糊化层,对应 $e$ 和 $\dot{e}$ 分别有9个高斯函数形式的隶属函数

$$\mu_i = \exp \left[ -\frac{(x_i - m_{ij})^2}{\sigma_{ij}^2} \right], i=1, 2, j=1, 2, \dots, 9$$

$\alpha_k = \mu_1^m \cdot \mu_2^n$ ,  $m=1, 2, \dots, 9$ ,  $n=1, 2, \dots, 9$ , 因此第三层规则层有 $9 \times 9=81$ 条规则。第四层为归一化层, $\bar{\alpha}_k = \alpha_k / \sum_{i=1}^{81} \alpha_i$ 。第五层为输出层,仅有一个输出,为推力器控制信号, $w_i$ 为控制规则。控制器的输入和输出均为归一化后的值,其数值区间为 $[-1, 1]$ ,以便控制器可以方便地在不同的控制对象中移植。

FNNMC的可调参数有三类:一类是规则中结论的数值,它们是第四、五层间的权系数,代表规则参数。第二、三类可调参数分别是高斯型隶属函数的均值 $m_{ij}$ 和标准差 $\sigma_{ij}$ , $i=1, 2$ , $j=1, 2, \dots, 9$ 。它们位于第二层的18个结点中,代表输入变量隶属函数的参数。

对AUV进行实时控制时,FNNMC利用BP算法不断修正各层参数在线学习规划器<sup>[9]</sup>给出的信号。由于BP算法的局限性,FNNMC参数的初始取值相当重要。该文利用所提的IPSO算法对FNNMC参数进行寻优,以保证控制器的控制性能。

### 3 IPSO 算法

#### 3.1 IPSO 算法描述

同基本PSO算法一样,IPSO仍将每个个体看作 $D$ 维搜索空间中的一个没有体积的微粒(点),这样的微粒共有 $m$ 个,每个微粒在搜索空间中以一定的速度飞行。这个速度根据它本身的飞行经验以及同伴的飞行经验进行动态调整,经过若干次飞行微粒群最终找到搜索空间的最优位置,对应待求问题的最优解。

IPSO算法是通过如下两式更新粒子速度及位置的:

$$v_{id}(t+1) = wv_{id}(t) + c_1r_1(t)(p_{id}(t) - x_{id}(t)) + c_2r_2(t)(p_{gd}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + \eta v_{id}(t) \quad (2)$$

其中, $t$ 表示粒子群当前时刻(代数), $v_{id}$ 为第 $i$ 个粒子第 $d$ 维的速度,被 $V_{\max}$ 所限制,以免粒子速度过大飞离搜索区域; $x_{id}$ 为第 $i$ 个粒子第 $d$ 维的位置; $c_1$ 和 $c_2$ 为加速因子; $r_1$ 和 $r_2$ 为 $[0, 1]$ 区间上的随机数; $p_{id}$ 为粒子 $i$ 所经历过的最好位置; $p_{gd}$ 为粒子群所经过的最好位置; $w$ 为惯性权重; $\eta$ 为约束常数。

对于加速因子 $c_1$ 和 $c_2$ 而言,其一般为固定常值。而粒子的运动状态在算法迭代过程中往往是不可预测的,这就需要根据

群的环境变化对粒子的速度作出相应的调整,以保证粒子群的多样性,平衡其全局和局部搜索能力,避免陷入局部极值。IPSO引入免疫理论中基于浓度和适应度的抗体更新策略,对 $c_1$ 和 $c_2$ 进行动态调整,令:

$$D_i = \sum_d \sqrt{\sum_j (x_{id} - x_{jd})^2} \quad (3)$$

表示粒子 $i$ 和群中其他粒子的距离度量,则定义

$$A_i = \frac{1}{1 + D_i} \quad (4)$$

为粒子 $i$ 在群中的浓度,于是:

$$c_1 = c_2 = A_i \sqrt{1 - F_i} \quad (5)$$

其中 $F_i$ 为粒子 $i$ 的适应度,接下来的小节中将给出计算说明。

考察式(5),由免疫原理可知,若某粒子的浓度越大,亲和力和力越小,便越偏离最优解,因此,应对其当前位置作较大的调整,相应的加速因子应当较大;反之,若某粒子的浓度越小,亲和力和力越高,便与最优解越接近,相应的加速因子应当较小。

#### 3.2 IPSO 算法收敛性分析

假定 $r_1(t+1)=r_1(t)$ , $r_2(t+1)=r_2(t)$ , $p_{id}(t+1)=p_{id}(t)$ , $p_{gd}(t+1)=p_{gd}(t)$ 。由式(1)和(2)可得

$$x_{id}(t+1) = (1+w-c_1r_1-c_2r_2)x_{id}(t) - wx_{id}(t-1) + c_1r_1p_{id} + c_2r_2p_{gd} \quad (6)$$

$$\text{令 } A = \begin{bmatrix} 1+w-c_1r_1-c_2r_2 & -w & c_1r_1p_{id}+c_2r_2p_{gd} \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

则式(6)可改写为:

$$\begin{bmatrix} x_{id}(t+1) \\ x_{id}(t) \\ 1 \end{bmatrix} = A \cdot \begin{bmatrix} x_{id}(t) \\ x_{id}(t-1) \\ 1 \end{bmatrix} \quad (7)$$

式(7)中系数矩阵的特征多项式为:

$$(1-\lambda)[w-\lambda(1+w-c_1r_1-c_2r_2)+\lambda^2] \quad (8)$$

其根为:

$$\tilde{\lambda} = \begin{bmatrix} \lambda_{1d} \\ \lambda_{2d} \\ \lambda_{3d} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1+w-c_1r_1-c_2r_2+\sigma_{id}}{2} \\ \frac{1+w-c_1r_1-c_2r_2-\sigma_{id}}{2} \end{bmatrix}$$

其中 $\sigma_{id} = \sqrt{(1+w-c_1r_1-c_2r_2)^2 - 4w}$ 。

令 $\tilde{\theta} = [\theta_1 \ \theta_2 \ \theta_3]$ ,则式(2)可写为 $x_{id} = \tilde{\theta} \tilde{\lambda}$ 。由式(5)可知, $c_1=c_2 \leq 1$ 。于是当 $w>0$ 时, $\max(\|\lambda_{2d}\|, \|\lambda_{3d}\|) < 1$ ,则

$$\lim_{t \rightarrow \infty} x_{id}(t) = \lim_{t \rightarrow \infty} \tilde{\theta} \tilde{\lambda} < \lim_{t \rightarrow \infty} (\theta_1 + \theta_2 + \theta_3) = (\theta_1 + \theta_2 + \theta_3)$$

因此,当取 $w>0$ 时,算法是收敛的。

### 4 基于IPSO的模糊神经网络控制器参数优化设计

#### 4.1 控制器参数编码

每个FNNMC中,待优化的参数包括18个中心值、18个敏感域值、 $9 \times 9=81$ 个模糊规则,一共117个参数。这么多的参数完全由IPSO自寻优,对于目前的个人计算机来说将是一项相当艰巨的工作。

由于模糊系统中隶属函数一般是关于 $Y$ 轴左右对称的,对于FNNMC的一个输入变量的隶属函数而言,只需要知道 $X$ 轴一侧的参数即可。因此可以将隶属函数参数降至 $(4+5) \times 2=$

18个。如前所述,已经将控制器的两个输入变量规范到 $[-1,1]$ 区间内,因此对应于每个输入的隶属函数中心值必将包括0,1(或者-1,0)。并且,初始时刻各个隶属函数的形状是相同的,在线控制时,由BP算法微调控制参数来适应环境变化。这样所有的敏感域值都相同,隶属函数参数的数目为 $3 \times 2 + 1 = 7$ 。最后待优化的参数数目就变为 $81 + 7 = 88$ 。将这88个参数作为微粒群中粒子位置矢量在各维的分量,搜索空间的维数 $D$ 即为88,微粒群对位置的寻优即是对控制器参数的寻优。

#### 4.2 IPSO 算法参数的选取和粒子群初始化

考察式(1)、(2),待确定的参数为惯性权重 $w$ ,加速因子 $c_1$ 和 $c_2$ ,约束常数 $\eta$ ,以及粒子最大速度 $V_{\max}$ 。

根据经验,取 $w=0.6, \eta=0.8$ 。对于 $V_{\max}$ ,由于所有相关参数均经过归一化处理,可以统一为其设定 $[-1,1]$ ,群体规模 $m=80$ , $c_1$ 和 $c_2$ 由式(5)动态调整。用随机数初始化各粒子初始位置和速度,并将当前的位置记录为各粒子最优位置 $P_i$ ,计算出各粒子的适应度,以此确定群体最优位置 $P_g$ 。

#### 4.3 粒子目标函数的选择和适应度变换

根据问题需要,每个微粒实际上对应了一个模糊神经网络控制器,微粒目标函数则用来评价该控制器的性能,进而求出微粒的适应度,它是调整控制器参数的依据。采用控制系统最优设计中常用的 $ITAE = \alpha T + \beta E + \gamma S$ 指标作为评价控制器性能的一个标准,此指标越小则系统性能越好,目标函数 $J = \alpha T + \beta E + \gamma S$ 。式中: $T$ 表示上升时间, $E$ 表示最大超调量, $S$ 表示第一次上升到稳态后相应曲线和稳态值横坐标所围成的面积。 $\alpha, \beta$ 和 $\gamma$ 为各项加权系数,其取值要体现实际控制系统对稳定性和快速性的综合要求。为使 $J$ 极小,IPSO的适应度函数选 $F = 1/(\sigma + J)$ 。其中 $\sigma$ 为一小量,为了避免在特殊情况下 $J$ 为0,实际运算中取 $\sigma=0.1$ ,根据控制对象的运动特性,取 $\alpha=2, \beta=1, \gamma=0.1$ 。

#### 4.4 IPSO 算法流程

IPSO 算法流程如下:

- (1)初始化一群微粒(群体规模为 $m=80$ ),包括随机位置和速度;
- (2)评价每个微粒的适应度 $F$ ;
- (3)对每个微粒,将其适应度值与其经历过的最好位置 $P_i$ 作比较,如果较好,将其作为当前的最好位置,更新 $P_i$ ;
- (4)对每个微粒,将其适应度值与全局所经历的最好位置 $P_g$ 作比较,如果较好,重新设置 $P_g$ 的索引号;
- (5)根据式(1)、(2)、(5)更新微粒的速度和位置;
- (6)如未达到结束条件(达到一个预设最大代数 $G_{\max}=700$ ),则返回(2)。

#### 4.5 基于专家经验的加速策略

上述IPSO算法忽略了一部分设计者关于FNNMC的经验知识,是对现有资源的一种浪费。如果在其中予以补充,无疑会增加算法的搜索速度及效率。

根据经验,控制规则参数的变化范围可以精确到比 $[-1,1]$ 更小的区间内。比如敏感域值,可以将其固定在 $[0.1,0.2]$ 范围内,与此对应的粒子位置矢量的分量在初始化时可以在该区间内随机选值。另外,一部分控制规则参数是可以预知的,正如当位置偏差NB和偏差变化率NB时,控制规则参数很显然要取1;当位置偏差PB和偏差变化率PB时,控制规则参数很显然要取-1。因此可以将预知的参数引入粒子群初始时刻的最优位置矢量。同时,由于网络中隶属函数的排列是按照一定的次序

的,在每次更新粒子位置的时候,应将隶属函数参数所对应的位置矢量分量做一次排序。这样将会更好地帮助IPSO搜索到最优值。

### 5 仿真试验及结果分析

控制器优化仿真流程如图2所示。设定仿真时间步长为0.5s,每个方向的控制器用于学习的数据共取10组,每组数据用于每一代的每个粒子。样本数据选取时,综合考虑AUV的工作状态和海洋环境的影响(主要为流的干扰),比如:纵向定位控制时,初态有、无速度,距离目标点远、近以及有流、无流。仿真控制时间取400个仿真步长(这个时间可以保证控制器能够使系统达到所有样本数据提供的稳态),每次所得的偏差输出代入目标函数得到粒子的目标函数值,转化后即可得每个粒子的适应度。

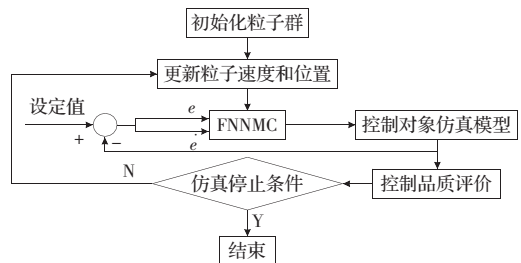


图2 控制器优化仿真流程图

表1所示为五种算法起始收敛代数(第一次达到收敛值的迭代次数)的平均值,从中可以看出,IPSO收敛速度最快,与HPSO相比,搜索效率有了较大幅度的提高。

表1 算法起始收敛代数

算法	控制器			
	纵向	横向	垂向	艏向
GA	569	633	647	582
IGA	491	575	523	548
PSO	519	587	563	531
HPSO	503	594	582	509
IPSO	456	503	474	462

为了验证所提出的控制器的可行性,在无人潜水器的仿真平台上进行了试验。该平台以某国家重点实验室研制的某型AUV为对象。该AUV是用于研究无人潜水器运动控制、路径规划和目标识别的试验平台。它由运动控制系统、视觉系统、路径规划系统、导航系统、应急处理系统和监视系统组成。其推进系统采用8个推进器。艏部4个导管桨,其中水平布置的2个作为主推,每个最大提供210N的力,另外2个提供纵倾力矩。垂推和侧推采用槽道推进器,分别为2个。考虑到AUV在水中的航速并不太高,对各自自由度方向运动进行解耦,每个自由度采用单独的控制器对该方向的运动进行控制,而各个自由度之间的耦合都作为外部干扰来处理。为了获得更精确的运动模型,数字仿真频率选择10Hz。而为了模拟实际情况,受到实际搭载在无人潜水器上传感器数据率的限制,控制器仿真频率选择2Hz。图3所示为无人潜水器的控制仿真平台。

图4为五种智能算法寻优得到的控制器在AUV四自由度运动方向上(考虑到AUV的运动能力和艇型,横摇和纵摇不予控制)的控制仿真曲线:给定参考输入分别为纵向5m,系统初态为0;横向5m,系统初态为0;垂向5m,系统初态为0;艏向



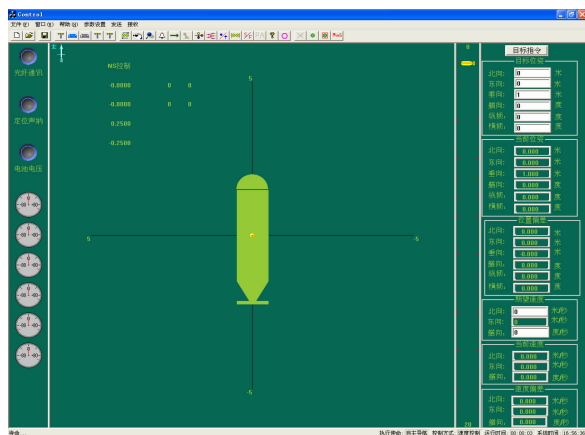
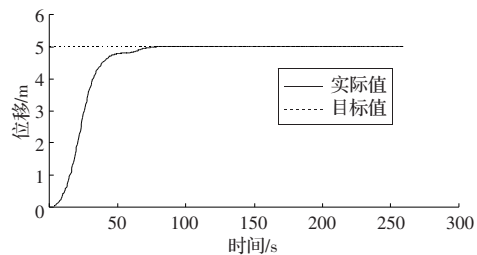


图3 无人潜水器控制仿真平台

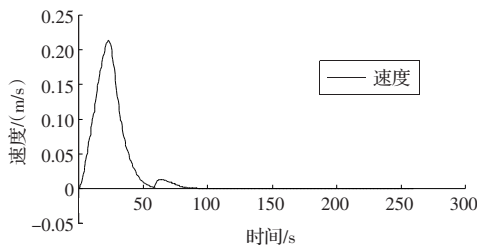
45°,系统初态为0。

按照响应速度、稳态误差和超调这些控制指标对五种算法寻优得到的控制器进行比较,基于 IPSO 的控制器效果最好。基于 GA、IGA 和 PSO 算法的控制器在横向定位控制和垂向定位控制时的控制效果远远低于纵向和艏向定位控制,这是由于推力分配时,横向和垂向这两个自由度方向的推力分别由两个槽道桨提供,推进效率比较低。而且 AUV 在这两个方向上运动时,受到艇体自身形状的影响比较大,运动能力相对较弱,不易控制,因此这两个自由度方向的控制器对参数初值比较敏感。总体看来,基于 HPSO 算法的控制器控制效果较之前三种算法有了明显的改进,但与 IPSO 算法寻优得到的控制器相比还有一些差距,特别在收敛速度上,IPSO 具有明显的优势。

在以上仿真试验的基础上,进行了海流环境下纵向定位控制的仿真试验,以验证基于 IPSO 的控制器在有外界干扰情况下的控制效果,试验结果如图5所示。图5(a)为海流环境下纵向定位控制时 AUV 的纵向运动轨迹曲线,图5(b)为相应的纵



(a)海流作用下纵向定位控制



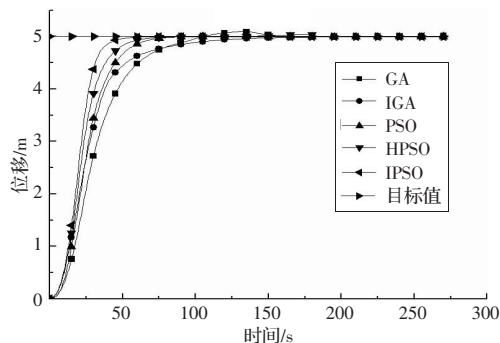
(b)纵向速度响应曲线

图5 海流作用下纵向定位控制

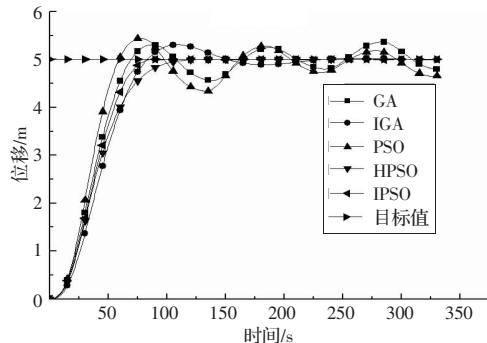
向速度响应曲线。海流速度为 0.3 m/s,流向角 180°(顶流),给定参考输入为纵向 5 m,系统初态为 0。考虑到 AUV 自身的运动能力,在有流的情况下,总是使其顶流前进,因此仅进行了纵向顶流定位控制的仿真。由图5可知,该控制器在海流环境中也具有响应快,超调和稳态误差小的优点,取得了良好的控制效果。

## 6 结语

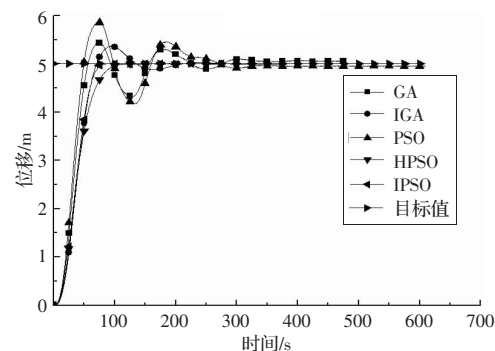
控制参数较多的控制器往往需要大量繁琐的手工调整,设计过程中不可避免地存在主观不确定性。该文的 IPSO 算法将免疫理论中基于浓度和适应度更新抗体的生物学原理引入



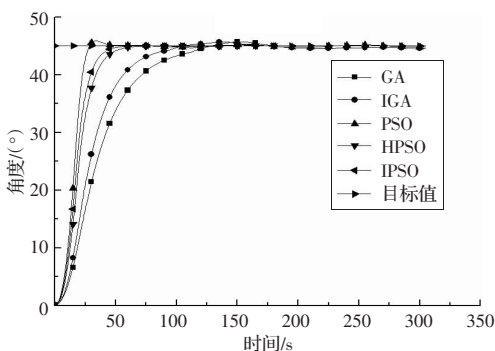
(a)纵向定位控制



(b)横向定位控制



(c)垂向定位控制



(d)艏向定位控制

图4 四自由度方向运动轨迹曲线

(下转 239 页)