

文章编号: 1001-0920(2012)02-0247-05

## 基于改进差分进化算法的在线轨迹优化

韩敏, 王明慧, 范剑超

(大连理工大学 电子信息与电气工程学院, 辽宁 大连 116024)

**摘要:** 首先提出一种基于混沌映射的差分进化算法, 通过引入混沌映射的概念, 在群体初始化和子代重构两个方面对经典差分进化算法进行改进, 提高其寻优精度及稳定性, 并通过对几个典型的 Benchmark 函数进行对比测试, 验证该算法的全局收敛能力与稳定性. 然后将该改进算法应用于在线轨迹优化, 利用其快速寻优、不依赖梯度信息等特点, 结合滚动窗口的思想, 提出局部极值逃逸方法, 实现了轨迹的在线优化. 最后在板球系统上通过仿真实验, 验证了所提出方法的有效性.

**关键词:** 差分进化; 混沌映射; 轨迹优化; 在线优化; 板球系统

**中图分类号:** TP273

**文献标识码:** A

## Trajectory optimization based on improved differential evolution algorithm

HAN Min, WANG Ming-hui, FAN Jian-chao

(Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China.  
Correspondent: HAN Min, E-mail: minhan@dlut.edu.cn)

**Abstract:** Firstly, the paper proposes a chaotic mapping based differential evolution algorithm. By introducing the concept of chaotic map, both the group initialization and reconstruction of offspring of classical differential evolution algorithm are modified to improve accuracy and stability. And several contrast tests on the typical Benchmark functions verify the global convergence and stability of the algorithm. Then, the improved algorithm is applied to online trajectory optimization. The best trajectory online is realized by using the characteristics of its fast searching and doing not rely on gradient information. By combining the ideas of rolling window, the method of local minimum escape is proposed. Finally, the simulation result on ball and plate system shows the effectiveness of the proposed method.

**Key words:** differential evolution; chaotic mapping; trajectory optimization; online optimization; ball and plate system

### 1 引言

轨迹优化的总目标是使运动物体运动到期望目标点, 且在整个过程中不碰到任何一个障碍物, 与此同时尽量使代价最小、路线最短或是运动时间最短. 随着运动控制领域的发展, 轨迹优化技术已向在线优化方面发展. 文献[1]提出了一种导航方法, 虽然实现了动态轨迹优化, 但是算法较复杂, 不易理解; [2]提出了一种有效的距离传播法, 该方法可以在较复杂的环境下优化轨迹, 但随着栅格数量的增多, 优化时间也会延长, 不利于在线优化; [3]将栅格法和局部势场法相结合实现了仿生机器鱼的实时避障, 但当环境较为复杂时, 轨迹的可达性便会降低. 因此, 轨迹的在

线优化还需要不断丰富优化方法, 引入新的算法. 而差分进化 (DE) 算法, 作为一种采用浮点矢量编码在连续空间中进行随机搜索的优化算法<sup>[4]</sup>, 它简单、鲁棒、收敛速度快, 且性能优于微粒群和其他进化类算法<sup>[5]</sup>, 在解决实际问题方面有很强的适应性和普遍性, 可以应用于轨迹在线优化之中. 但是与其他进化类算法一样, DE 存在早熟收敛现象, 现已有一些学者对 DE 进行了改进, 如文献 [6] 在种群进化过程中加入二次变异操作; [7] 将 DE 与免疫算法相结合; [8] 利用反向学习法对种群进行初始化以及跳变操作. 以上方法虽然可以加快收敛速度、降低早熟概率, 但算法的稳定性还有待加强.

收稿日期: 2010-09-16; 修回日期: 2010-12-19.

基金项目: 国家自然科学基金项目(61074096); 国家863计划项目(2007AA04Z158); 国家科技支撑计划项目(2006BAB14B05).

作者简介: 韩敏(1959—), 女, 教授, 博士生导师, 从事神经网络、混沌序列分析等研究; 王明慧(1987—), 女, 硕士生, 从事建模优化、系统辨识的研究.

本文首先针对 DE 随迭代次数增加而导致种群多样性快速下降、收敛速度变慢的问题,提出一种基于混沌映射的差分进化算法(CDE),提高其寻优精度以及稳定性,并通过几组典型 Benchmark 函数的测试,证明所提出算法的全局收敛能力强、稳定性高.然后,将 CDE 应用于在线轨迹优化,利用其快速寻优、不依赖梯度信息等特点,结合滚动窗口的思想,并提出局部极值逃逸方法,实现在线轨迹优化.最后,将上述方法在板球系统上进行仿真实验,验证了本文方法的有效性.

## 2 基于混沌映射的差分进化算法

DE 是根据父代个体间的差分矢量进行变异、交叉、选择操作,随着迭代次数的增加,种群多样性快速下降,算法向最优解收敛的速度明显减慢,导致产生早熟收敛.为了提高算法性能,避免早熟,必须在进化过程中保持收敛速度的同时,适当加强种群多样性.而混沌映射有着遍历性和内随机性的特点,恰好符合这样的需求,所以本文通过引入混沌映射的概念,在群体初始化和子代重构两方面对经典 DE 进行改进,以提高算法的全局收敛能力.

### 2.1 基于混沌映射的群体初始化

一维 Logistic 映射的迭代方程为

$$x^{k+1} = \mu x^k (1 - x^k), \quad x \in [0, 1], \quad (1)$$

其中  $x^k$  为混沌变量,当  $\mu = 4$  时,系统进入混沌状态,  $x^k$  在  $[0, 1]$  内遍历. 0, 0.25, 0.5, 0.75, 1 为 Logistic 映射的 5 个不动点. 对群体初始化的改进即是令分别由混沌映射和均匀随机分布法产生的两个群体进行竞争,选出较优个体作为初始群体,其具体步骤如下:

Step 1: 随机初始化群体  $RP(N_P)$ .

$$x_{i,j,1} = x_{j \min} + \text{rand} \times (x_{j \max} - x_{j \min}). \quad (2)$$

其中:  $i = 1, 2, \dots, N_P, j = 1, 2, \dots, D$ , rand 为  $[0, 1]$  之间的随机数,  $x_{j \min}$  和  $x_{j \max}$  分别为第  $j$  个变量的下界和上界.

Step 2: 随机生成一个  $(0, 1)$  之间的数  $x_1$  ( $x_1$  不能取到不动点),从  $x_1$  开始,根据式 (1) 生成混沌序列

$$\{x_1, x_2, \dots, x_{N_P \times D}\}. \quad (3)$$

Step 3: 将序列 (3) 重新按行排列得到混沌矩阵

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_P,1} & x_{N_P,2} & \cdots & x_{N_P,D} \end{bmatrix}. \quad (4)$$

Step 4: 将矩阵 (4) 中元素映射到种群个体每一维  $x_{ij}$  的取值范围  $[x_{j \min}, x_{j \max}]$ , 得到混沌初始群体  $CP(N_P)$ .

$$x_{ij} = x_{j \min} + x_{i,j}(x_{j \max} - x_{j \min}). \quad (5)$$

Step 5: 计算群体  $RP \cup CP$  中所有个体的目标函数值,并从小到大进行排序,取前  $N_P$  个个体作为最终的初始种群  $P(N_P)$ .

### 2.2 基于混沌映射的子代重构

在进化过程中,每隔一定代数  $n$  便跟踪观察最优个体的目标函数值 best\_value 的进化速度  $\alpha$ , 设当前最优个体目标函数值为 best\_value,  $n$  代前最优个体的目标函数值为 best\_value\_last, 则进化速度可用二者之比近似表示,即

$$\beta = \frac{\text{best\_value}}{\text{best\_value\_last}}. \quad (6)$$

$\beta$  越小,表明进化速度  $\alpha$  越大;反之,  $\beta$  越大,表明进化速度  $\alpha$  越小. 当  $\beta$  大于一定阈值  $\lambda$  时,通过混沌映射对子代群体进行一次重构,其步骤如下:

Step 1: 执行群体初始化中的 Step 2 和 Step 3.

Step 2: 将矩阵 (4) 中元素映射到种群个体每一维  $x_{ij}$  的取值范围  $[x_{j \min}, x_{j \max}]$ , 得到混沌迁移量

$$x_{i,j,G}^{ch} = x_{j \min} + x_{i,j}(x_{j \max} - x_{j \min}). \quad (7)$$

Step 3: 在子代群体  $P$  的基础上,根据下式:

$$x'_{i,j,G+1} = (1 - g) \times x_{i,j,G+1} + g \times x_{i,j,G}^{ch}, \quad (8)$$

进行混沌迁移,得到一个新的群体  $CP(N_P)$ . 其中  $g$  为  $[0, 1]$  之间的一个系数,即迁移度.

Step 4: 计算群体  $P \cup CP$  中所有个体的目标函数值,并从小到大进行排序,取前  $N_P$  个个体作为下一代进化的父代群体  $P_{G+1}$ .

除此之外,其他的基本进化过程与经典 DE 相同<sup>[4]</sup>.

## 3 静态环境下的在线轨迹优化

### 3.1 问题描述

本文讨论局部在线多目标轨迹优化,即在一定范围内运动物体无碰撞且轨迹最短,最终由起点  $p_{\text{ori}}$  安全地到达目标点  $p_{\text{end}}$ .

在运动区域内建立直角坐标系,每个轨迹点在该坐标系中都有确定的坐标  $(x, y)$ . 同时将坐标系分割成  $N_x \times N_y$  的栅格<sup>[3]</sup>, 每个栅格都对应一个栅格坐标  $(g_x, g_y)$ ,  $g_x$  为栅格所在行号,  $g_y$  为栅格所在列号. 记录有碰撞危险度值  $C_{ij}$  ( $i \in [0, N_x], j \in [0, N_y]$ ), 若该区域存在障碍物,则  $C_{ij} = 1$ ; 若该区域不存在障碍物,则  $C_{ij} = 0$ .

本文由欧几里得距离得到长度能量函数,通过碰撞罚函数来降低碰撞的可能性,因此,待优化目标函数为长度能量函数和碰撞罚函数的加权和,即

$$E = wE_l + (1 - w)E_c. \quad (9)$$

其中:  $w$  为权值,  $E_l$  为长度能量函数,  $E_c$  为碰撞罚函数, 且

$$E_c = \sum_{i=1}^D C_i, \quad (10)$$

$$E_l = \sum_{i=1}^D \|x_i - x_{i-1}\| + \|p_{\text{end}} - x_D\|. \quad (11)$$

### 3.2 在线轨迹优化

借鉴滚动窗口<sup>[9]</sup>的思想, 在以运动物体为中心、 $R$  为半径的窗口中, 利用 CDE 作为搜索算法, 进行局部轨迹优化, 从而得到目标函数值最优的一系列轨迹节点  $p_i (i = 1, 2, \dots, D)$  的位置坐标, 即

$$p_i = [(x_1, y_1), \dots, (x_i, y_i), \dots, (x_D, y_D)], \quad (12)$$

其中  $D$  为种群个体维数. 在线优化对算法的实时性要求很强, 故  $D$  不能过大. 每次寻优后, 将  $p_2$  作为前进的轨迹点, 并更新窗口, 由此不断向  $p_{\text{end}}$  趋近, 直至到达  $p_{\text{end}}$ .

在优化过程中, 当所遇障碍物为凹多边形或是运动物体与目标点连线垂直于障碍物边界时, 由于目标点的吸引, 采用滚动的规划方法轨迹便会陷入局部极值, 即运动物体停滞不前, 而且在没有外界的帮助或干扰下无法自动逃逸. 所以, 为了提高轨迹的可达性, 需要将运动物体从局部极值处引导出来. 以障碍物为凹多边形为例 (见图 1), 逃逸的具体步骤如下:

**Step 1:** 判断轨迹是否陷入局部极值, 即运动物体是否停滞不前, 若是, 则转 Step 2; 若否, 则转 Step 4.

**Step 2:** 在运动物体和目标点连线的垂线上远处任找一点作为临时目标点  $p'_{\text{end}}$ , 进行轨迹优化.

**Step 3:** 判断是否已经逃逸出局部极值, 若是, 则恢复原来的目标点  $p_{\text{end}}$ , 转 Step 4; 若否, 则转 Step 2.

**Step 4:** 按正常情况进行轨迹优化.

由此, 便实现了可达性较强的在线轨迹优化, 其整体流程如下:

**Step 1:** 初始化环境信息并建模.

**Step 2:** 获取起始点  $p_{\text{ori}}$  和目标点  $p_{\text{end}}$ .

**Step 3:** 判断是否陷入局部极值, 若是, 则执行上述逃逸算法, 再转 Step 6; 若否, 则转 Step 4.

**Step 4:** 建立并更新窗口.

**Step 5:** 在窗口中利用 CDE 算法搜索前进轨迹点  $p_i$ , 并滚动前进至  $p_i$ .

**Step 6:** 判断是否到达目标点, 若否, 则转 Step 3; 若是, 则结束.

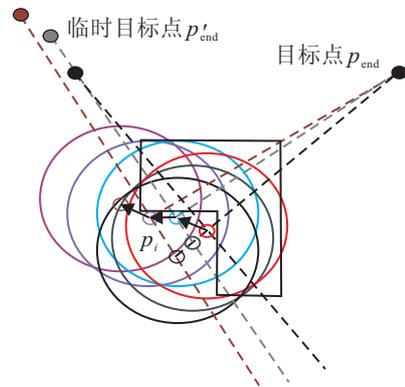


图 1 局部极值逃逸示意图

## 4 仿真实验

### 4.1 CDE 函数优化仿真

为了验证所提出算法 CDE 的有效性, 对 6 个典型 Benchmark 函数进行测试, 并与经典 DE 算法和 ODE (opposition-based differential evolution)<sup>[8]</sup> 算法进行比较. 选取测试函数如下:  $f_1(x)$  为 Sphere 函数,  $f_2(x)$  为 Rosenbrock 函数,  $f_3(x)$  为 Rastrigin 函数,  $f_4(x)$  为 Griewank 函数,  $f_5(x)$  为 Ackley's path 函数,  $f_6(x)$  为 Quartic 函数.

为了公平比较, 本文采用与文献 [8] 中相同的参数与设置, 群体大小  $N_P = 10D$ ,  $F = 0.5$ ,  $CR = 0.9$ , 算法停止条件为到达  $N_P \times \text{gen\_max}$ ,  $\text{gen\_max}$  为最大进化代数, 仿真时测试每种问题都运行 100 次, 表 1 为 3 种算法达到的平均最优解和标准差的结果比较.

表 1 3 种算法运行 100 次的平均最优解与标准差比较

测试函数	$D$	进化代数	DE		ODE <sup>[8]</sup>		CDE	
			平均最优值	标准差	平均最优值	标准差	平均最优值	标准差
$f_1(x)$	3	50	3.49E-07	5.12E-07	4.79E-08	5.81E-08	<b>3.92E-08</b>	<b>5.10E-08</b>
	50	5000	1.25E-13	4.88E-14	1.73E-64	1.59E-64	<b>1.16E-69</b>	<b>2.87E-70</b>
$f_2(x)$	2	50	8.00E-03	2.93E-02	4.80E-03	1.80E-02	<b>1.17E-03</b>	<b>8.79E-03</b>
	50	7000	5.3	9.56E-01	5.01E-01	8.90E-01	<b>2.78E-03</b>	<b>5.67E-04</b>
$f_3(x)$	2	3000	0	0	0	0	0	0
	50	10000	3.43E+02	9.80E+00	6.56E+01	6.80E+01	<b>1.72E+01</b>	<b>5.05E+00</b>
$f_4(x)$	2	180	5.19E-04	1.90E-03	1.49E-04	1.00E-03	<b>7.40E-05</b>	<b>7.40E-04</b>
	50	5000	6.13E-11	2.02E-11	1.53E-62	2.38E-62	<b>0</b>	<b>0</b>
$f_5(x)$	3	50	1.07E-02	7.40E-03	4.80E-03	3.70E-03	<b>2.88E-03</b>	<b>2.03E-03</b>
	50	5000	2.33E-06	2.33E-06	<b>7.02E-15</b>	1.66E-15	7.64E-15	<b>1.12E-15</b>
$f_6(x)$	30	5000	5.10E-03	9.69E-04	8.70E-04	3.30E-04	<b>8.57E-04</b>	<b>2.86E-04</b>

由表1中数据可以看出,本文算法CDE的优化结果优于DE和ODE.其中,在解决函数 $f_1(\mathbf{x})$ , $f_2(\mathbf{x})$ , $f_3(\mathbf{x})$ 和 $f_4(\mathbf{x})$ 的高维问题时,寻优精度明显优于DE和ODE;在解决其他问题时,除了 $f_5(\mathbf{x})$ 函数的高维问题以外,寻优精度也较高.同时由标准差可知,CDE算法更稳定.由此可以说明,CDE算法不仅寻优能力更强,而且稳定性也更高.

以 $f_3(\mathbf{x})$ 的高维问题和 $f_4(\mathbf{x})$ 的低维问题为例,如图2和图3所示,CDE的收敛速度与ODE算法相近,并明显优于经典DE;在解决 $f_3(\mathbf{x})$ 的高维问题时,DE和ODE都明显出现了早熟收敛现象,而CDE则可以收敛到更优解.由此可知,CDE算法不仅保证了收敛速度,还提高了寻优精度,加强了稳定性.

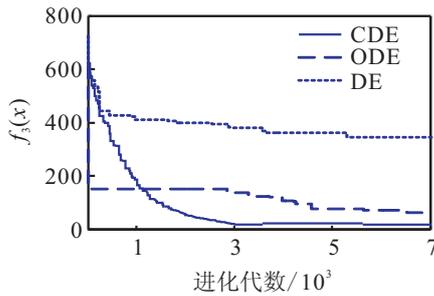


图2  $f_3(\mathbf{x})$ 的最优值进化曲线  
( $D=50$ ,  $\text{gen\_max}=10000$ )

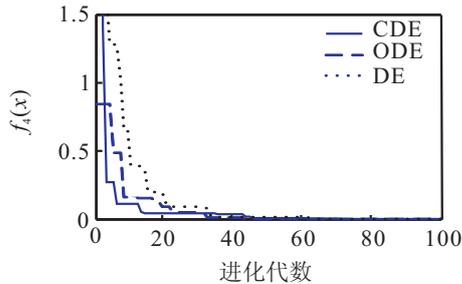


图3  $f_4(\mathbf{x})$ 最优值进化曲线  
( $D=2$ ,  $\text{gen\_max}=180$ )

## 4.2 基于板球系统的在线轨迹优化仿真实验

本文中的轨迹优化是根据运动物体的当前位置与环境信息实时进行的,所以板球系统的控制精度会直接影响到在线轨迹优化的结果,故首先解决板球系统的控制问题.

### 4.2.1 关于板球系统模型及其控制

板球系统是一个多变量、非线性、具有两个自由度的机械系统,同时也是一个存在耦合、不确定性和时滞性的动力学系统.其控制对象是具有两个相互垂直的旋转轴的板,目的是让一个自由滚动的小球能够平衡在平板上特定的位置,或者沿一定的轨迹滚动.平板绕 $X$ 轴和 $Y$ 轴的旋转由两个电机驱动,传感装置得到小球在板上的位置后反馈给控制系统,控制系统采用一定的控制策略控制板的转角,从而达到对小球平衡位置和运动轨迹的控制.

根据文献[10]中建立的模型,并忽略耦合及摩擦,即可得到如下板球系统的简化状态方程:

$$\begin{cases} \dot{x}_1 \\ \dot{x}_2 \end{cases} = \begin{bmatrix} x_2 \\ kx_1\alpha^2 \end{bmatrix} - \begin{bmatrix} 0 \\ k \end{bmatrix} u_\alpha, \\ \begin{cases} \dot{x}_3 \\ \dot{x}_4 \end{cases} = \begin{bmatrix} x_4 \\ kx_3\beta^2 \end{bmatrix} - \begin{bmatrix} 0 \\ k \end{bmatrix} u_\beta. \quad (13)$$

系统的状态变量为

$$X = (x_1, x_2, x_3, x_4)^T = (x, \dot{x}, y, \dot{y})^T, \quad (14)$$

以及系统输入

$$u_\alpha = g \sin \alpha, u_\beta = g \sin \beta, \quad (15)$$

系统输出

$$Y = (x_1, x_3)^T. \quad (16)$$

其中: $x$ 和 $y$ 为小球位置坐标; $\alpha$ 和 $\beta$ 为平板 $X$ 轴方向和 $Y$ 轴方向上的倾角; $k = r^2/(mr^2 + I_b)$ , $r$ 为小球半径, $m$ 为小球质量, $I_b$ 为小球的转动惯量.由此,板球系统可简化为 $X$ 轴、 $Y$ 轴方向上的两个一维子系统,实现了解耦控制.

本文采用如图4所示的双闭环控制,内环控制平板角度,反馈数据是电机当前旋转角度;外环控制小球的位置,反馈数据是摄像机得到的小球实际位置; $r(t)$ 为设定的期望目标; $y(t)$ 为小球的实际位置.

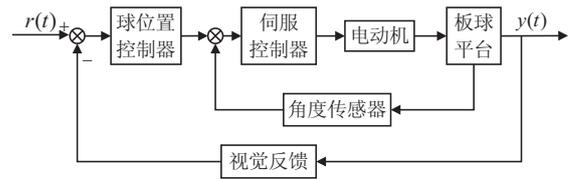


图4 板球系统控制示意图

在Matlab 7.1的Simulink环境下搭建模型进行仿真.其中,轨迹优化结果作为控制给定;位置控制器采用PD-模糊控制,输入为小球的位置误差和小球实际速度,输出为期望的电机角度;伺服控制器采用PID控制,板球系统的数学模型(13)~(16)通过S-Function模块来描述.

### 4.2.2 在线轨迹优化的仿真实验

轨迹优化部分的参数选取: $D=4$ , $N_P=40$ , $F=0.5$ , $\text{CR}=0.9$ , $n=5$ , $\lambda=0.3$ , $g=0.001$ , $\text{gen\_max}=100$ , $R=10$ , $w=0.00001$ ,起始点 $p_{\text{ori}}=(5,10)$ ,目标点 $p_{\text{end}}=(95,95)$ .

板球控制部分的参数选取:PD控制器中 $K_P=1.5$ , $K_D=1$ ;PID控制器中 $K_P=100$ , $K_I=20$ , $K_D=10$ .仿真结果如图5所示.

由图5可以看出,本文提出的在线轨迹优化算法可以根据小球当前的位置,实时计算出符合优化目标的轨迹,引导小球及时地避开障碍物,向目标点趋近,

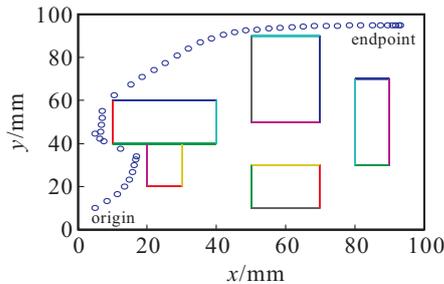


图5 在线轨迹优化在板球系统上的实现

最终停留在目标点. 局部极值逃逸方法的加入, 可以使小球在进入障碍物凹处时能够逃逸出来, 而不致停滞不前. 由此表明, 基于CDE的在线轨迹优化算法是有效的.

## 5 结 论

为了加强DE的收敛能力、提高其寻优精度, 本文提出了CDE算法, 改进了DE群体初始化和子代构建方式, 使算法在保持种群多样性的同时, 平衡了全局搜索能力和局部搜索能力. 在非线性函数优化的对比仿真分析中, 验证了该算法不仅收敛速度快, 而且寻优精度高、稳定性强. 故将CDE应用于轨迹优化领域之中, 提出了一种简单可行的在线轨迹优化方法, 为移动机器人、板球系统等轨迹优化提供一种有效的选择; 并针对滚动优化容易陷入局部极值的问题提出了局部极值逃逸方法. 由在板球系统上的仿真实验结果可知, 本文提出的基于CDE的在线轨迹优化方法是行之有效的.

## 参考文献(References)

- [1] Belkhouche F. Reactive path planning in a dynamic environment[J]. IEEE Trans on Robotics, 2009, 25(4): 902-911.
- [2] Willms A R, Yang S X. Real-time robot path planning via a distance-propagating dynamic system with obstacle clearance[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2008, 38(3): 884-893.
- [3] 陈尔奎, 喻俊志, 王硕, 等. 一种基于视觉的仿生机器人实时避障综合方法[J]. 控制与决策, 2004, 19(4): 452-458.  
(Chen E K, Yu J Z, Wang S, et al. Real-time obstacle-avoidance synthesized method for bionic robot fish based on visual information[J]. Control and Decision, 2004, 19(4): 452-458.)
- [4] Storn R, Price K. Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces[R]. Berkeley: International Computer Science Institute, 1995: 1-15.
- [5] Vesterstrom J, Thomsen R. A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems[C]. Proc of the 2004 Congress on Evolutionary Computation. Portland: IEEE, 2004: 1980-1987.
- [6] 吴亮红, 王耀南, 袁小芳, 等. 自适应二次变异差分进化算法[J]. 控制与决策, 2006, 21(8): 898-902.  
(Wu L H, Wang Y N, Yuan X F, et al. Differential evolution algorithm with adaptive second mutation[J]. Control and Decision, 2006, 21(8): 898-902.)
- [7] 吴燕玲, 卢建刚, 孙优贤. 基于免疫原理的差分进化[J]. 控制与决策, 2007, 22(11): 1309-1312.  
(Wu Y L, Lu J G, Sun Y X. Differential evolution based on immunity[J]. Control and Decision, 2007, 22(11): 1309-1312.)
- [8] Shahryar R, Hamid R Tizhoosh, Magdy M A Salama. Opposition-based differential evolution[J]. IEEE Trans on Evolutionary Computation, 2008, 12(1): 64-79.
- [9] 席裕庚, 张纯刚. 一类动态不确定环境下机器人的滚动路径规划[J]. 自动化学报, 2002, 28(2): 161-175.  
(Xi Y G, Zhang C G. Rolling path planning of mobile robot in a kind of dynamic uncertain environment[J]. Acta Automatica Sinica, 2002, 28(2): 161-175.)
- [10] Wang H R, Tian Y T, Fu S Y, et al. Nonlinear control for output regulation of ball and plate system[C]. Proc of the 27th Chinese Control Conf. Kunming: IEEE, 2008: 382-387.
- [11] Fitzgibbon A W. Simultaneous linear estimation of multiple view geometry and lens distortion[C]. Proc of CVPR. Hawaii: IEEE Press, 2001: 125-132.
- [12] Brown D C. Close-range camera calibration[J]. Photogrammetric Engineering, 1971, 37(8): 855-866.
- [13] Chernov N, Lesort C. Least squares fitting of circles[J]. J of Mathematical Imaging and Vision, 2005, 23(3): 239-252.

(上接第246页)