



# 第10章 智能算法及其应用

# 10.1 遗传算法的基本原理

遗传算法简称GA (Genetic Algorithms) 是1962年由美国Michigan大学的Holland教授提出的模拟自然界遗传机制和生物进化论而成的一种并行随机搜索最优化方法。

遗传算法是以达尔文的自然选择学说为基础发展起来的。自然选择学说包括以下三个方面：

(1) 遗传：这是生物的普遍特征，亲代把生物信息交给子代，子代总是和亲代具有相同或相似的性状。生物有了这个特征，物种才能稳定存在。

(2) 变异：亲代和子代之间以及子代的不同个体之间的差异，称为变异。变异是随机发生的，变异的选择和积累是生命多样性的根源。

(3) 生存斗争和适者生存：具有适应性变异的个体被保留下来，不具有适应性变异的个体被淘汰，通过一代代的生存环境的选择作用，性状逐渐逐渐与祖先有所不同，演变为新的物种。

遗传算法将“优胜劣汰，适者生存”的生物进化原理引入优化参数形成的编码串联群体中，按所选择的适应度函数并通过遗传中的复制、交叉及变异对个体进行筛选，使适应度高的个体被保留下来，组成新的群体，新的群体既继承了上一代的信息，又优于上一代。这样周而复始，群体中个体适应度不断提高，直到满足一定的条件。遗传算法的算法简单，可并行处理，并能到全局最优解。

遗传算法的基本操作为：

### (1) 复制 (Reproduction Operator)

复制是从一个旧种群中选择生命力强的个体位串产生新种群的过程。具有高适应度的位串更有可能在下一代中产生一个或多个子孙。

复制操作可以通过随机方法来实现。首先产生0~1之间均匀分布的随机数，若某串的复制概率为40%，则当产生的随机数在0.40~1.0之间时，该串被复制，否则被淘汰。

### (2) 交叉 (Crossover Operator)

复制操作能从旧种群中选出优秀者，但不能创造新的染色体。而交叉模拟了生物进化过程中的繁殖现象，通过两个染色体的交换组合，来产生新的优良品种。

交叉的过程为：在匹配池中任选两个染色体，随机选择一点或多点交换点位置；交换双亲染色体交换点右边的部分，即可得到两个新的染色体数字串。

交叉体现了自然界中信息交换的思想。交叉有一点交叉、多点交叉、还有一致交叉、顺序交叉和周期交叉。一点交叉是最基本的方法，应用较广。它是指染色体切断点有一处，例：

$A: 101100 \ 1110 \rightarrow 101100 \ 0101$

$B: 001010 \ 0101 \rightarrow 001010 \ 1110$

### (3) 变异 (Mutation Operator)

变异运算用来模拟生物在自然的遗传环境中由于各种偶然因素引起的基因突变，它以很小的概率随机地改变遗传基因（表示染色体的符号串的某一位）的值。在染色体以二进制编码的系统中，它随机地将染色体的某一个基因由1变为0，或由0变为1。若只有选择和交叉，而没有变异，则无法在初始基因组合以外的空间进行搜索，使进化过程在早期就陷入局部解而进入终止过程，从而影响解的质量。为了在尽可能大的空间中获得质量较高的优化解，必须采用变异操作。

## 10.2 遗传算法的特点

(1) 遗传算法是对参数的编码进行操作，而非对参数本身，这就是使得我们在优化计算过程中可以借鉴生物学中染色体和基因等概念，模仿自然界中生物的遗传和进化等机理；

(2) 遗传算法同时使用多个搜索点的搜索信息。传统的优化方法往往是从解空间的单个初始点开始最优解的迭代搜索过程，单个搜索点所提供的信息不多，搜索效率不高，有时甚至使搜索过程局限于局部最优解而停滞不前。遗传算法从由很多个体组成的一个初始群体开始最优解的搜索过程，而不是从一个单一的个体开始搜索，这是遗传算法所特有的一种隐含并行性，因此遗传算法的搜索效率较高。

(3) 遗传算法直接以目标函数作为搜索信息。传统的优化算法不仅需要利用目标函数值，而且需要目标函数的导数值等辅助信息才能确定搜索方向。而遗传算法仅使用由目标函数值变换来的适应度函数值，就可以确定进一步的搜索方向和搜索范围，无需目标函数的导数值等其他一些辅助信息。遗传算法可应用于目标函数无法求导数或导数不存在的函数的优化问题，以及组合优化问题等。

(4) 遗传算法使用概率搜索技术。遗传算法的选择、交叉、变异等运算都是以一种概率的方式来进行的，因而遗传算法的搜索过程具有很好的灵活性。随着进化过程的进行，遗传算法新的群体会更多地产生出许多新的优良的个体。

(5) 遗传算法在解空间进行高效启发式搜索，而非盲目地穷举或完全随机搜索；

(6) 遗传算法对于待寻优的函数基本无限制，它既不要求函数连续，也不要求函数可微，既可以是数学解析式所表示的显函数，又可以是映射矩阵甚至是神经网络的隐函数，因而应用范围较广；

(7) 遗传算法具有并行计算的特点，因而可通过大规模并行计算来提高计算速度，适合大规模复杂问题的优化。

## 10.3 遗传算法的发展及应用

### 10.3.1 遗传算法的发展

遗传算法起源于对生物系统所进行的计算机模拟研究。早在20世纪40年代，就有学者开始研究如何利用计算机进行生物模拟的技术，他们从生物学的角度进行了生物的进化过程模拟、遗传过程模拟等研究工作。进入20世纪60年代，美国密执安大学的Holland教授及其学生们受到这种生物模拟技术的启发，创造出一种基于生物遗传和进化机制的适合于复杂系统优化计算的自适应概率优化技术——遗传算法。

以下是在遗传算法发展进程中一些关键人物所做出的主要贡献：

## (1) J. H. Holland

20世纪70年代初，Holland教授提出了遗传算法的基本定理—模式定理，从而奠定了遗传算法的理论基础。模式定理揭示了群体中优良个体（较好的模式）的样本数将以指数级规律增长，从理论上保证了遗传算法用于寻求最优可行解的优化过程。1975年，Holland出版了第一本系统论述遗传算法和人工自适应系统的专著《自然系统和人工系统的自适应性》。20世纪80年代，Holland教授实现了第一个基于遗传算法的机器学习系统—分类器系统，开创了基于遗传算法的机器学习的新概念。

## (2) J. D. Bagley

1967年，Holland的学生Bagley在其博士论文中首次提出了“遗传算法”一词，并发表了遗传算法应用方面的第一篇论文。他发展了复制、交叉、变异、显性、倒位等遗传算子，在个体编码上使用了双倍体的编码方法。在遗传算法的不同阶段采用了不同的概率，从而创立了自适应遗传算法的概念。

### (3) K. A. De Jong

1975年，De Jong博士在其博士论文中结合模式定理进行了大量纯数值函数优化计算实验，树立了遗传算法的工作框架。他推荐了在大多数优化问题中都较适用的遗传算法的参数，建立了著名的De Jong五函数测试平台，定义了评价遗传算法性能的在线指标和离线指标。

### (4) D. J. Goldberg

1989年，Goldberg出版了专著《搜索、优化和机器学习中的遗传算法》，该书全面地论述了遗传算法的基本原理及其应用，奠定了现代遗传算法的科学基础。

## (5) L. Davis

1991年，Davis编辑出版了《遗传算法手册》一书，为推广和普及遗传算法的应用起到了重要的指导作用。

## (6) J. R. Koza

1992年，Koza将遗传算法应用于计算机程序的优化设计及自动生成，提出了遗传编程的概念，并成功地将遗传编程的方法应用于人工智能、机器学习和符号处理等方面。

## 10.3.1 遗传算法的应用

### (1) 函数优化。

函数优化是遗传算法的经典应用领域，也是遗传算法进行性能评价的常用算例。尤其是对非线性、多模型、多目标的函数优化问题，采用其他优化方法较难求解，而遗传算法却可以得到较好的结果。

### (2) 组合优化。

随着问题的增大，组合优化问题的搜索空间也急剧扩大，采用传统的优化方法很难得到最优解。遗传算法是寻求这种满意解的最佳工具。例如，遗传算法已经在求解旅行商问题、背包问题、装箱问题、图形划分问题等方面得到成功的应用。

### (3) 生产调度问题

在很多情况下，采用建立数学模型的方法难以对生产调度问题进行精确求解。在现实生产中多采用一些经验进行调度。遗传算法是解决复杂调度问题的有效工具，在单件生产车间调度、流水线生产车间调度、生产规划、任务分配等方面遗传算法都得到了有效的应用。

#### （4）自动控制。

在自动控制领域中有很多与优化相关的问题需要求解，遗传算法已经在其中得到了初步的应用。例如，利用遗传算法进行控制器参数的优化、基于遗传算法的模糊控制规则的学习、基于遗传算法的参数辨识、基于遗传算法的神经网络结构的优化和权值学习等。

#### （5）机器人

例如，遗传算法已经在移动机器人路径规划、关节机器人运动轨迹规划、机器人结构优化和行为协调等方面得到研究和应用。

## （6）图像处理

遗传算法可用于图像处理过程中的扫描、特征提取、图像分割等的优化计算。目前遗传算法已经在模式识别、图像恢复、图像边缘特征提取等方面得到了应用。

## （7）人工生命

人工生命是用计算机、机械等人工媒体模拟或构造出的具有生物系统特有行为的人造系统。人工生命与遗传算法有着密切的联系，基于遗传算法的进化模型是研究人工生命现象的重要基础理论。遗传算法为人工生命的研究提供了一个有效的工具。

## （8）遗传编程

遗传算法已成功地应用于人工智能、机器学习等领域的编程。

## （9）机器学习

基于遗传算法的机器学习在很多领域都得到了应用。例如，采用遗传算法实现模糊控制规则的优化，可以改进模糊系统的性能；遗传算法可用于神经网络连接权的调整和结构的优化；采用遗传算法设计的分类器系统可用于学习式多机器人路径规划。

## 10.4 遗传算法的优化设计

### 10.4.1 遗传算法的构成要素

#### (1) 染色体编码方法

基本遗传算法使用固定长度的二进制符号来表示群体中的个体，其等位基因是由二值符号集  $\{0, 1\}$  所组成。初始个体基因值可用均匀分布的随机值生成，如

$$x = 100111001000101101$$

就可表示一个个体，该个体的染色体长度是18。

(2) 个体适应度评价：基本遗传算法与个体适应度成正比的概率来决定当前群体中每个个体遗传到下一代群体中的概率多少。为正确计算这个概率，要求所有个体的适应度必须为正数或零。因此，必须先确定由目标函数值 $J$ 到个体适应度 $f$ 之间的转换规则。

(3) 遗传算子：基本遗传算法使用下述三种遗传算子：

① 选择运算：使用比例选择算子；

② 交叉运算：使用单点交叉算子；

③ 变异运算：使用基本位变异算子或均匀变异算子。

#### (4) 基本遗传算法的运行参数

有下述4个运行参数需要提前设定：

M：群体大小，即群体中所含个体的数量，一般取为20~100；

G：遗传算法的终止进化代数，一般取为100~500；

$P_c$ ：交叉概率，一般取为0.4~0.99；

$P_m$ ：变异概率，一般取为0.0001~0.1。

## 10.4.2 遗传算法的应用步骤

对于一个需要进行优化的实际问题，一般可按下述步骤构造遗传算法：

第一步：确定决策变量及各种约束条件，即确定出个体的表现型 $X$ 和问题的解空间；

第二步：建立优化模型，即确定出目标函数的类型及数学描述形式或量化方法；

第三步：确定表示可行解的染色体编码方法，即确定出个体的基因型 $x$ 及遗传算法的搜索空间；

第四步：确定解码方法，即确定出由个体基因型 $x$ 到个体表现型 $X$ 的对应关系或转换方法；

第五步：确定个体适应度的量化评价方法，即确定出由目标函数值到个体适应度的转换规则；

第六步：设计遗传算子，即确定选择运算、交叉运算、变异运算等遗传算子的具体操作方法。

第七步：确定遗传算法的有关运行参数，即 $M, G, P_c, P_m$ 等参数。

以上操作过程可以用图10-1来表示。

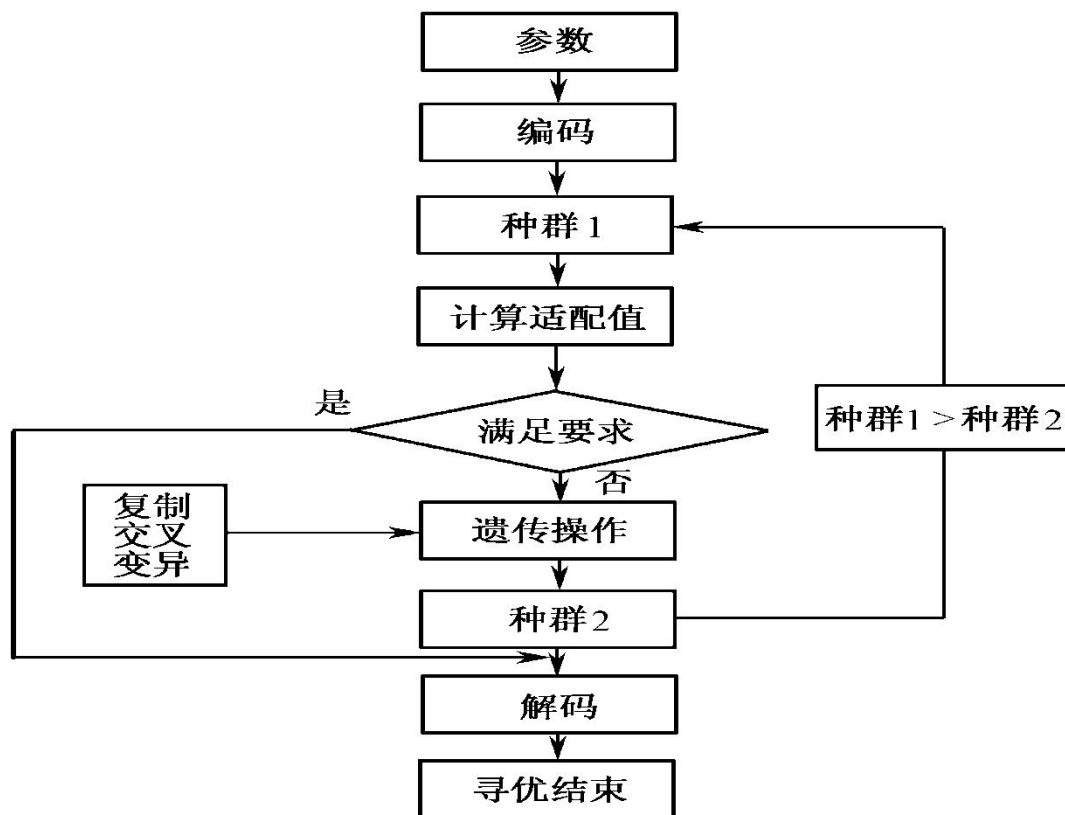


图10-1 遗传算法流程图

## 10.5 遗传算法求函数极值

利用遗传算法求Rosenbrock函数的极大值

$$\begin{cases} f_2(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \\ -2.048 \leq x_i \leq 2.048 \quad (i = 1, 2) \end{cases}$$

## 10.5.1 二进制编码遗传算法求函数极大值

求解该问题遗传算法的构造过程：

- (1) 确定决策变量和约束条件；
- (2) 建立优化模型；
- (3) 确定编码方法

用长度为10位的二进制编码串来分别表示两个决策变量 $x_1, x_2$ 。10位二进制编码串可以表示从0到1023之间的1024个不同的数，故将 $x_1, x_2$ 的定义域离散化为1023个均等的区域，包括两个端点在内共有1024个不同的离散点。从离散点-2.048到离散点2.048，分别对应于从0000000000(0)到1111111111(1023)之间的二进制编码。将 $x_1, x_2$ 分别表示的两个10位长的二进制编码串连接在一起，组成一个20位长的二进制编码串，它就构成了这个函数优化问题的染色体编码方法。使用这种编码方法，解空间和遗传算法的搜索空间就具有一一对应的关系。例如： $x: 0000110111 \quad 1101110001$  表示一个个体的基因型，其中前10位表示 $x_1$ ，后10位表示 $x_2$ 。

(4) 确定解码方法：解码时需要将20位长的二进制编码串切断为两个10位长的二进制编码串，然后分别将它们转换为对应的十进制整数代码，分别记为 $y_1$ 和 $y_2$ 。

依据个体编码方法和对定义域的离散化方法可知，将代码 $y$ 转换为变量 $x$ 的解码公式为

$$x_i = 4.096 \times \frac{y_i}{1023} - 2.048 \quad (i = 1, 2)$$

例如，对个体  $x$  : 0000110111    1101110001

它由两个代码所组成

$$y_1 = 55, y_2 = 881$$

上述两个代码经过解码后，可得到两个实际的值

$$x_1 = -1.828, x_2 = 1.476$$

(5) 确定个体评价方法：由于Rosenbrock函数的值域总是非负的，并且优化目标是求函数的最大值，故可将个体的适应度直接取为对应的目标函数值，即

$$F(x) = f(x_1, x_2)$$

选个体适应度的倒数作为目标函数

$$J(x) = \frac{1}{F(x)}$$

(6) 设计遗传算子：选择运算使用比例选择算子，交叉运算使用单点交叉算子，变异运算使用基本位变异算子。

(7) 确定遗传算法的运行参数：群体大小M=80，终止进化代数G=100，交叉概率Pc=0.60，变异概率Pm=0.10。

上述七个步骤构成了用于求函数极大值的优化计算基本遗传算法。

采用上述方法进行仿真，经过100步迭代，最佳样本为

$$\text{Best } S = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

即当  $x_1 = -2.0480$  ,  $x_2 = -2.0480$  时，**Rosenbrock**函数具有极大值，极大值为3905.9。

仿真程序：chap5\_1.m

遗传算法的优化过程是目标函数 $J$ 和适应度函数 $F$ 的变化过程。

由仿真结果可知，随着进化过程的进行，群体中适应度较低的一些个体被逐渐淘汰掉，而适应度较高的一些个体会越来越多，并且它们都集中在所求问题的最优点附近，从而搜索到问题的最优解。

## 10.5.2 实数编码遗传算法求函数极大值

求解该问题遗传算法的构造过程：

- (1) 确定决策变量和约束条件；
- (2) 建立优化模型；
- (3) 确定编码方法：用2个实数分别表示两个决策变量，分别将的定义域离散化为从离散点-2.048到离散点2.048的Size个实数。

(4) 确定个体评价方法:

个体的适应度直接取为对应的目标函数值, 即

$$F(x) = f(x_1, x_2)$$

选个体适应度的倒数作为目标函数

$$J(x) = \frac{1}{F(x)}$$

(5) 设计遗传算子：选择运算使用比例选择算子，交叉运算使用单点交叉算子，变异运算使用基本位变异算子。

(6) 确定遗传算法的运行参数：群体大小 $M=500$ ，终止进化代数 $G=200$ ，交叉概率 $P_c=0.90$ ，采用自适应变异概率

$$P_m = 0.10 - [1:1:\text{Size}] \times 0.01/\text{Size}$$

即变异概率与适应度有关，适应度越小，变异概率越大。

上述六个步骤构成了用于求函数Rosenbrock极大值的优化计算的实数编码遗传算法。

十进制编码求函数Rosenbrock极大值。仿真程序见chap10\_2.m。

仿真程序经过200步迭代，最佳样本为

$$\text{BestS} = [-2.0438 \quad -2.044]$$

即当  $x_2 = -2.044$ ， $x_1 = -2.0438$  时，函数具有极大值，极大值为3880.3。

## 10.6 基于遗传算法优化的RBF网络逼近

### 10.6.1 遗传算法优化原理

在7.3节的RBF网络逼近算法中，网络权值、高斯函数的中心矢量和基宽向量的初值难以确定，如果这些参数选择不当，会造成逼近精度的下降甚至RBF网络的发散。采用遗传算法可实现RBF网络参数的优化。

为获取满意的逼近精度，采用误差绝对值指标作为参数选择的最小目标函数。

$$J = 100 \sum_{i=1}^N e(i)$$

式中， $N$  为逼近的总步骤， $e(i)$  为第  $i$  步RBF网络的逼近误差。

在应用遗传算法时，为了避免参数选取范围过大，可以先按经验选取一组参数，然后再在这组参数的周围利用遗传算法进行设计，从而大大减少初始寻优的盲目性，节约计算量。

## 10.6.2 仿真实例

使用RBF网络逼近下列对象：

$$y(k) = u(k)^3 + \frac{y(k-1)}{1 + y(k-1)^2}$$

在RBF网络中，网络输入信号为2个，即  $u(k)$  和  $y(k)$ ，网络初始权值及高斯函数参数初始值通过遗传算法优化而得。

遗传算法优化程序为chap10\_3a.m，取逼近总步骤为，每一步的逼近误差由chap10\_3b.m求得。采用二进制编码方式，用长度为10位的二进制编码串来分别表示向量b、c和w中的每个值。

遗传算法优化中，取样本个数为**Size=30**，交叉概率为**Pc=0.60**，采用自适应变异概率，即适应度越小，变异概率越大，取变异概率为

$$P_m = 0.001 - [1:1:Size] \times 0.001/Size$$

取用于优化的RBF网络结构为2-3-1，网络权值 $w_j$ 的取值范围为 $[-1,+1]$ ，高斯函数基宽向量的取值范围为 $[0.1,3.0]$ ，高斯函数中心矢量的取值范围为 $[-3,+3]$ 。取  $p=[b_1 \ b_2 \ b_3 \ c_{11} \ c_{12} \ c_{13} \ c_{21} \ c_{22} \ c_{23} \ w_1 \ w_2 \ w_3]$  则共有12个参数需要优化。

经过150代遗传算法进化，优化后的结果为：

$p=[2.7732, 2.6343, 2.2630, 1.8680, -0.0616, -0.7126, -0.3959, 2.2669, -1.4047, -0.3099, 0.7478, -0.3353]$ 。

则RBF网络高斯基函数参数的初始值及权值的初始值为：

$$\mathbf{w} = [-0.3099, 0.7478, -0.3353]^T$$

$$\mathbf{c} = \begin{bmatrix} 1.8680 & -0.0616 & -0.7126 \\ -0.3959 & 2.2669 & -1.4047 \end{bmatrix}$$

$$\mathbf{b} = [2.7732, 2.6343, 2.2630]^T$$

**RBF网络遗传算法优化程序包括三部分，即遗传算法优化程序chap10\_3a.m、RBF网络逼近函数程序chap10\_3b.m和RBF网络逼近测试程序chap10\_3c.m。**

## 10.7 基于遗传算法的伺服系统静态摩擦参数辨识

摩擦分为静摩擦和动摩擦，它们之间的区别就是看两个物体之间的接触面有没有相对位移，分为三种情况：如果没有相对位移，也没有位移的趋势，则是没有摩擦力；如果没有相对位移，但是有位移的趋势，则为静摩擦力；如果有相对位移，则为动摩擦。

摩擦现象是一种复杂的、非线性的、具有不确定性的自然现象。在高精度、超低速伺服系统中，由于非线性摩擦环节的存在，使系统的动态及静态性能受到很大程度的影响，主要表现为低速时出现爬行现象，稳态时有较大的静差或出现极限环振荡。克服这一问题的有效方法是进行摩擦辨识，从而进行有效的补偿。

## 10.7.1 伺服系统的静态摩擦模型

基于SISO的伺服系统可描述为

$$J\ddot{\theta} = u - F \quad (10.7)$$

其中  $J$  为转动惯量,  $\theta$  为转角,  $u$  为控制输入力矩,  $F$  为摩擦力矩。

许多伺服系统在低速情况下存在静摩擦现象。静摩擦力矩与转速之间的稳态对应关系为

$$F = \left[ F_c + (F_s - F_c) e^{-\left(\frac{\dot{\theta}}{V_s}\right)^2} \right] \text{sgn}(\dot{\theta}) + \alpha \dot{\theta} \quad (10.8)$$

其中  $F_c$ 、 $F_s$ 、 $\alpha$ 、 $V_s$  为静态摩擦参数,  $F_c$  为库仑摩擦,  $F_s$  为静摩擦,  $\alpha$  为粘性摩擦系数,  $V_s$  为切换速度。

伺服系统在正反转速度方向运行时，其静态摩擦力矩的静态参数通常为不同的值。当  $\dot{\theta} > 0$  时，静态参数的值为  $F_c^+$ 、 $F_s^+$ 、 $\alpha^+$ 、 $V_s^+$ ；当  $\dot{\theta} < 0$ ，静态参数的值为  $F_c^-$ 、 $F_s^-$ 、 $\alpha^-$ 、 $V_s^-$ ，表示如下：

$$F^i = \begin{cases} \left( F_c^+ + (F_s^+ - F_c^+) e^{-\left(\frac{\dot{\theta}_i}{V_s^+}\right)^2} \right) \operatorname{sgn}(\dot{\theta}_i) + \alpha^+ \dot{\theta}_i, & \dot{\theta}_i > 0 \\ \left( F_c^- + (F_s^- - F_c^-) e^{-\left(\frac{\dot{\theta}_i}{V_s^-}\right)^2} \right) \operatorname{sgn}(\dot{\theta}_i) + \alpha^- \dot{\theta}_i, & \dot{\theta}_i < 0 \end{cases} \quad (10.9)$$

由上式所确定的转速-摩擦力矩曲线称为**Stribeck**曲线。对于摩擦模型中的静态参数，通过恒速跟踪实验可以得到**Stribeck**曲线，再由**Stribeck**曲线，采用遗传算法可辨识出静态参数[22]。

恒速跟踪实验为在线条件下运行，工程上很容易实现，而遗传算法是在离线情况下运行，因而本方法具有较强的实际工程价值。

## 10.7.2 静摩擦模型Stribeck曲线的获取

在静态摩擦条件下， $\ddot{\theta} = 0$ ，由式（10.7）可知， $u = F$ 。由于 $u$ 可测得，采用一组恒速跟踪，可获得一组相应的控制输入信号和静态摩擦力矩，从而获得Stribeck曲线。

具体方法为：取闭环系统的一组恒定转速序列  $\{\dot{\theta}_i\}_{i=1}^N$  作为速度指令信号，通过采用PD控制律，实现被控对象精确的速度跟踪，得到相应的控制力矩序列  $\{u\}_{i=1}^N$ ，从而获得一组相应的静态摩擦力矩序列。

PD控制律为

$$u_i = k_p e_i + k_d \dot{e}_i \quad (10.10)$$

### 10.7.3 基于遗传算法的静态摩擦参数辨识

取待辨识静态摩擦参数向量为个体，遗传算法的每步迭代得到静态摩擦参数的辨识值为：

$$\hat{x}_m = [\hat{F}_c^+ \quad \hat{F}_s^+ \quad \hat{\alpha}^+ \quad \hat{V}_s^+ \quad \hat{F}_c^- \quad \hat{F}_s^- \quad \hat{\alpha}^- \quad \hat{V}_s^-]^T, m=1,2,\dots,M \quad (10.11)$$

其中 $M$ 为种群规模。

由下式可得到相应的摩擦力矩辨识值：

$$\hat{F}^i = \begin{cases} \left[ \hat{F}_c^+ + (\hat{F}_s^+ - \hat{F}_c^+) e^{-\left(\frac{\dot{\theta}_i}{\hat{V}_s^+}\right)^2} \right] \text{sgn}(\dot{\theta}_i) + \hat{\alpha}^+ \dot{\theta}_i, & \dot{\theta}_i > 0 \\ \left[ \hat{F}_c^- + (\hat{F}_s^- - \hat{F}_c^-) e^{-\left(\frac{\dot{\theta}_i}{\hat{V}_s^-}\right)^2} \right] \text{sgn}(\dot{\theta}_i) + \hat{\alpha}^- \dot{\theta}_i, & \dot{\theta}_i < 0 \end{cases} \quad i=1,2,\dots,N \quad (10.12)$$

辨识误差为:

$$e_i = F^i - \hat{F}^i, \quad i=1,2,\dots,N$$

其中值根据所建立的Stribeck曲线得到。

取目标函数为

$$J_m = \frac{1}{2} \sum_{i=1}^N e_i^2 \quad m=1,2,\dots,M \quad (10.13)$$

选择个体适应度函数如下:

$$\begin{cases} C_{\max} = \max_m \{J_m\} \\ f_m = C_{\max} - J_m \end{cases} \quad \text{或} \quad f_m = \frac{1}{J_m}, \quad m=1,2,\dots,M \quad (10.14)$$

采用十进制浮点编码格式, 选择操作采取保存最优个体的随机采样选择方法, 交叉操作采用均匀交叉算子, 交叉概率  $p_c = 0.9$ , 变异操作采用高斯变异算子, 变异概率随进化代数自适应调整, 即,  $p_m(g) = 0.1 - (0.1 - 0.001) \frac{g}{G}$  其中g当前遗传代数。

遗传算法的步骤如下：

Step 1. 置进化代数计数器为  $t=0$ ，随机产生初始化种群  $P_0$ ；

Step 2. 计算个体适应度  $f(X_i)$ ， $i=1,2,\dots,M$ ；

Step 3. 判断是否达到最大进化代数，若是，则算法终止，否则，转step 4；

Step 4. 经过选择操作，产生新一代种群  $P(t)$ ；

Step 5. 以概率  $p_c$  进行交叉操作；

Step 6. 以概率  $p_m$  进行个体变异操作；

Step 7.  $t+1 \rightarrow t$ ，转step 2；

一旦辨识得到的参数估计值，便可以设计摩擦力矩的补偿环节，实现对系统的摩擦进行补偿，基于摩擦力矩补偿的控制系统描述为：

$$J\ddot{\theta}=u-F+\hat{F} \quad (10.15)$$

## 10.7.4 仿真实例

被控对象为（10.7）式，取  $J=0.20$ ，控制律取PD控制。  
参数辨识的仿真分以下两步实现。

仿真之一：Stribeck曲线的测试

恒速跟踪时，为静态摩擦，实际系统的摩擦模型取（10.9）式，被控对象为带有静态摩擦模型式（10.7）的实际对象，取  $F_c^+ = 0.15$ ,  $F_s^+ = 0.60$ ,  $\alpha^+ = 0.02$ ,  $V_s^+ = 0.05$ ,  $F_c^- = 0.20$ ,  $F_s^- = 0.70$ ,  $\alpha^- = 0.03$ ,  $V_s^- = 0.05$ 。取速度信号  $\{\dot{\theta}_i\}_{i=1}^N = [-1.0:0.05:+1.0]$  作为指令信号，共41个速度指令信号。针对每个指令信号，采用PD控制律，取  $k_p = 200$ ， $k_d = 100$ 。仿真结果如图10-8至10-10所示。仿真结束后，将所得到的静摩擦力矩估计值保存在文件Fi\_file.mat中。

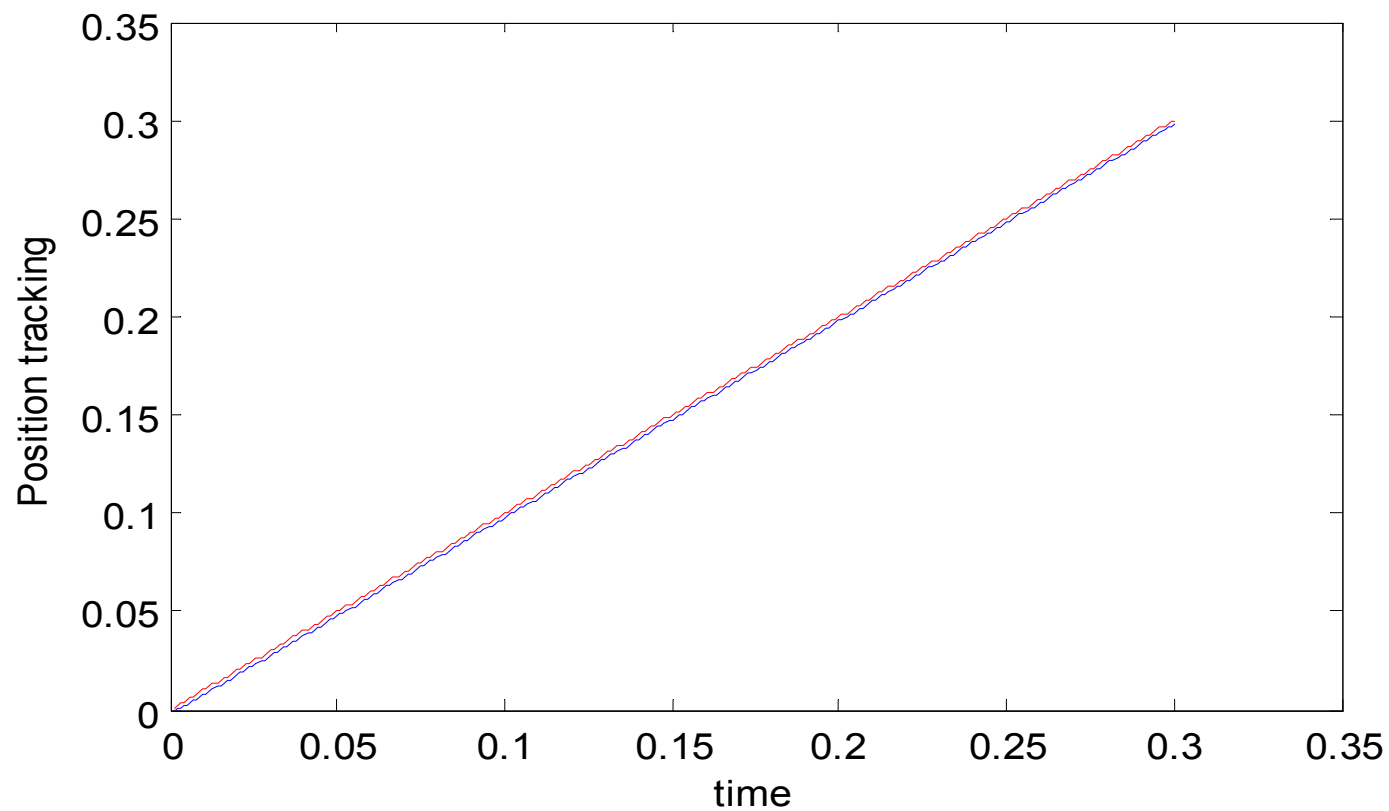


图10-8 恒速斜波跟踪（速度指令为1.0时）

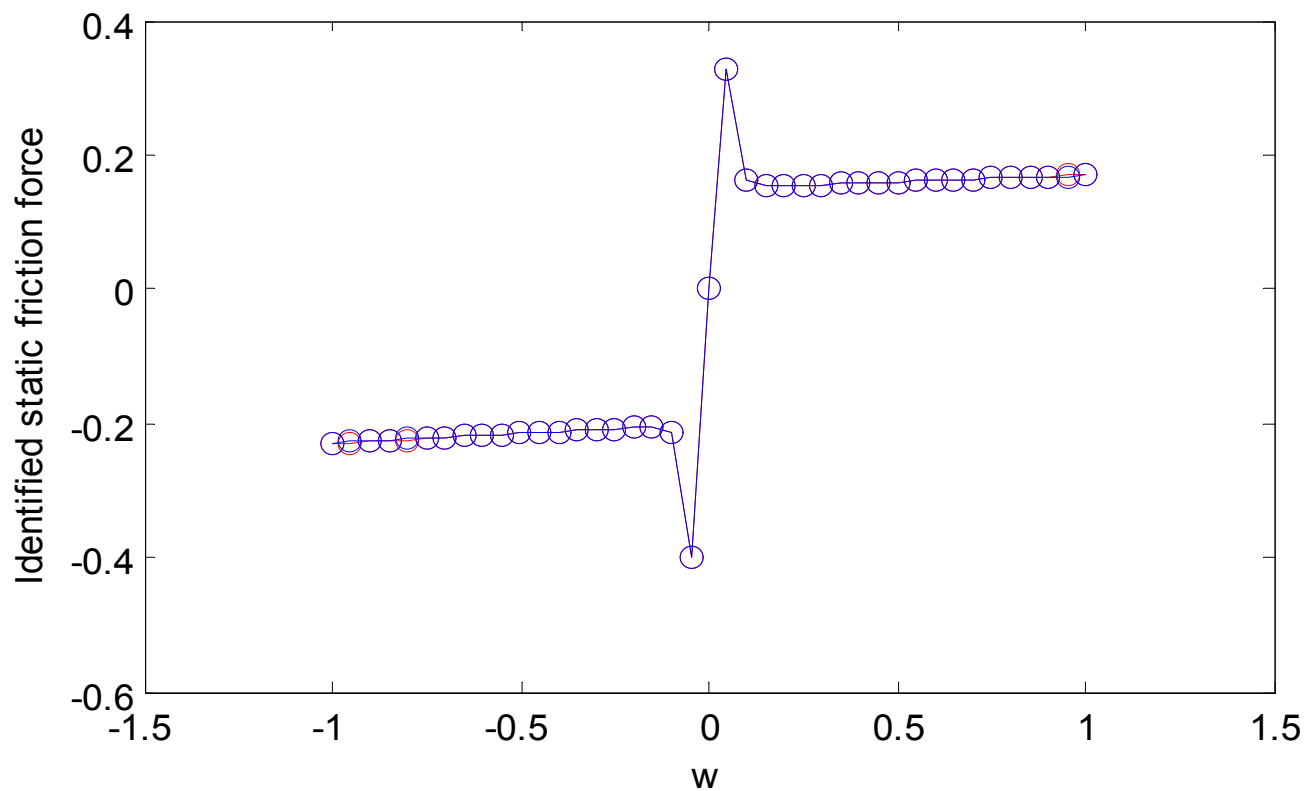


图10-9 Stribeck曲线的实际值与测试值的比较

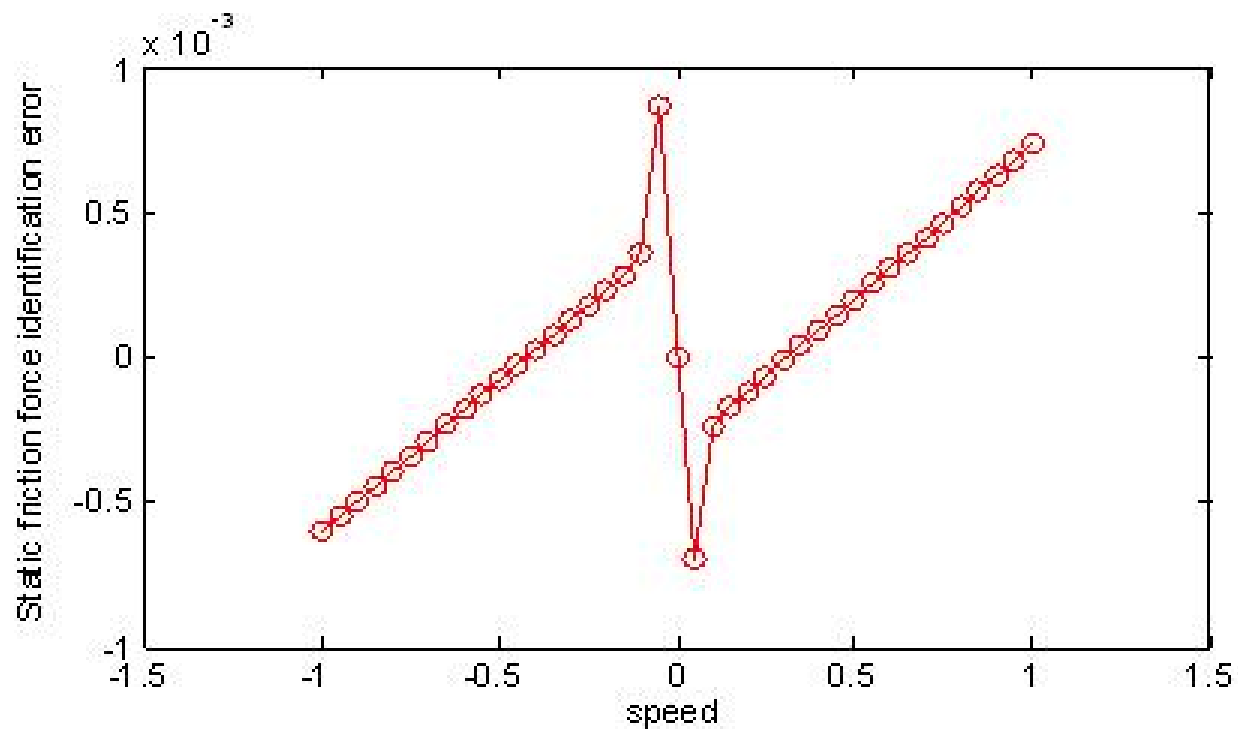


图10-10 Stribeck曲线的辨识误差

## 仿真之二：遗传算法的静态摩擦参数辨识

首先将仿真之一“Stribeck曲线设计”所得到的摩擦力矩估计值从文件Fi\_file.mat中调入，作为实际系统的静摩擦力矩的估计值。

恒速跟踪时为静态摩擦。取速度信号  $\{\dot{\theta}_i\}_{i=1}^N = [-1.0:0.05:+1.0]$  作为指令信号，共41个速度指令信号。针对每个指令信号，采用PD控制律，取  $k_p = 200$  ,  $k_d = 200$  。

在遗传算法仿真中，取种群规模，最大遗传代数  $G = 25000$ 。参数搜索范围为  $F_c \in [0,1]$  ,  $F_s \in [0,1]$  ,  $\alpha \in [0,0.1]$  ,  $V_s \in [0,0.1]$ ，静态摩擦模型取（10.9）式，将遗传算法设计所得到了静摩擦力矩与实际摩擦力矩比较得到目标函数值。当运行到最大遗传代数  $G = 25000$ 时，目标函数值为  $3.5786 \times 10^{-7}$

遗传算法的辨识过程及辨识曲线如图10-11和10-12所示，实际值与辨识值比较的仿真结果如表10-1所示。

表10-1 静态摩擦参数实际值与辨识值的比较

	$F_c^+$	$F_s^+$	$\alpha^+$	$V_s^+$	$F_c^-$	$F_s^-$	$\alpha^-$	$V_s^-$
真实值	0.15	0.60	0.02	0.05	0.20	0.70	0.03	0.05
辨识值	0.1503	0.6343	0.0191	0.0499	0.2001	0.73	0.0295	0.0506

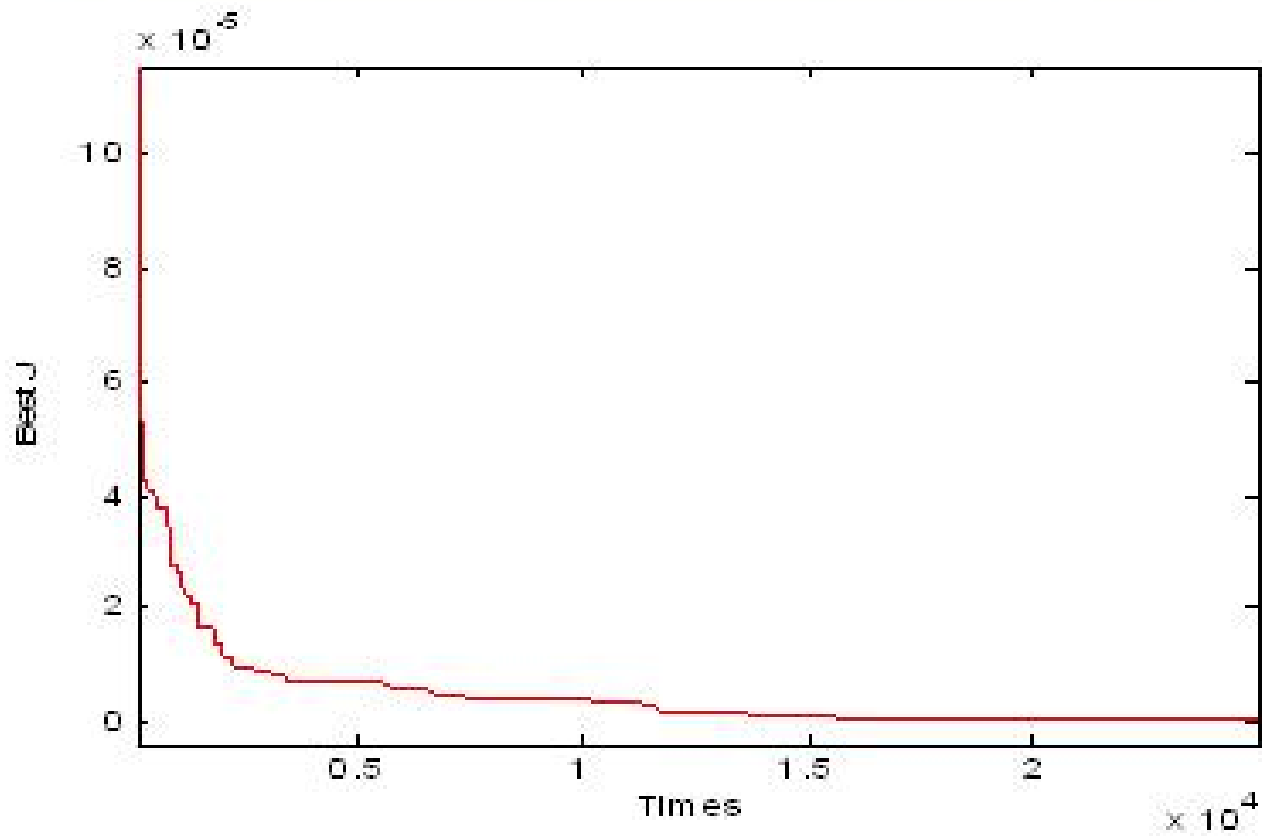


图10-11 目标函数值变化曲线局部放大图

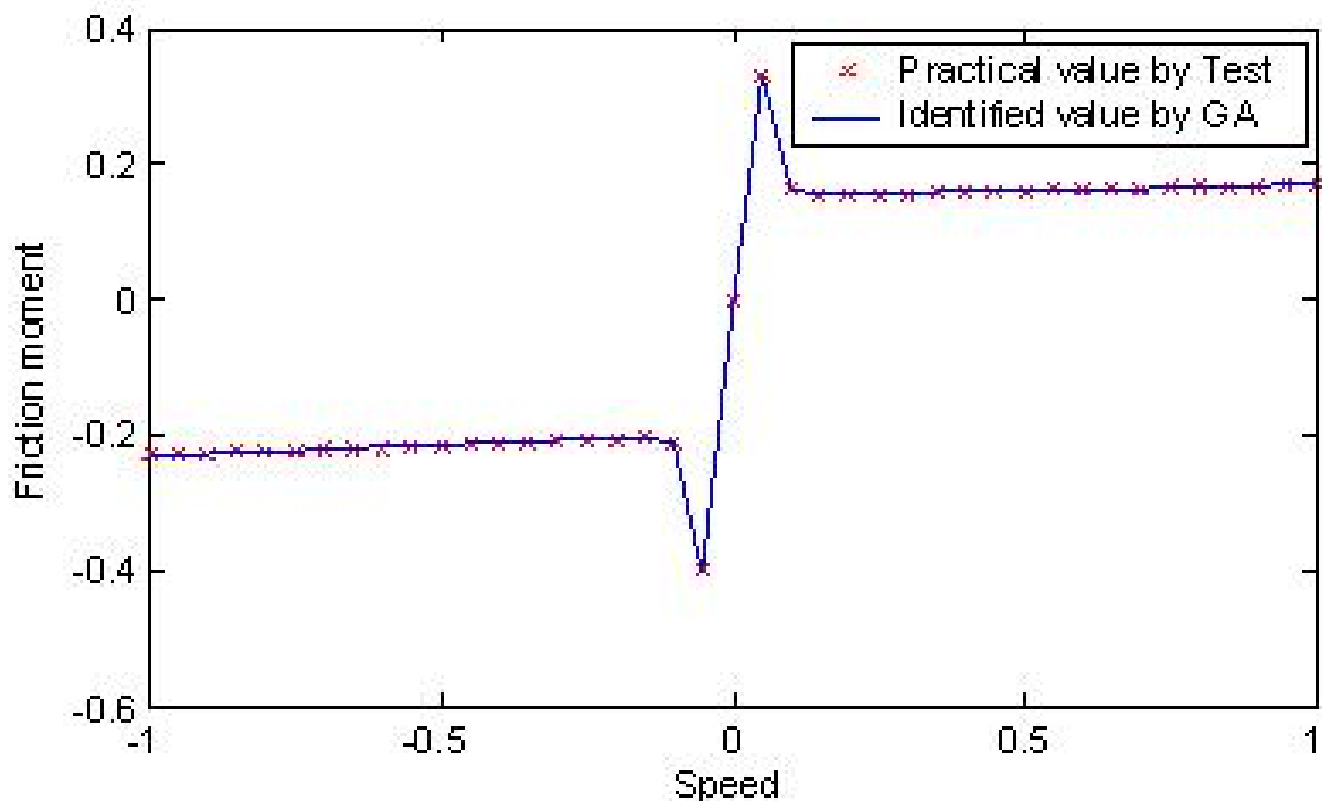


图10-12 辨识Stribeck曲线与实际  
Stribeck曲线

## 10.7.5 基于摩擦模型补偿的伺服系统控制

为了验证基于遗传算法的伺服系统静态摩擦参数辨识的效果，将PD控制与基于静态摩擦模型补偿的PD控制进行比较，采用低速正弦信号  $0.1\sin(t)$  作为位置指令。在PD控制中，取  $k_p = 50$  ,  $k_d = 5$  。采用手动转换开关实现两种控制方法的切换，一种为只采用PD控制，另一种为基于摩擦补偿的PD控制，仿真结果如图10-13和10-14所示。在图10-13中，由于存在静摩擦，造成了位置跟踪出现了“平顶”现象。

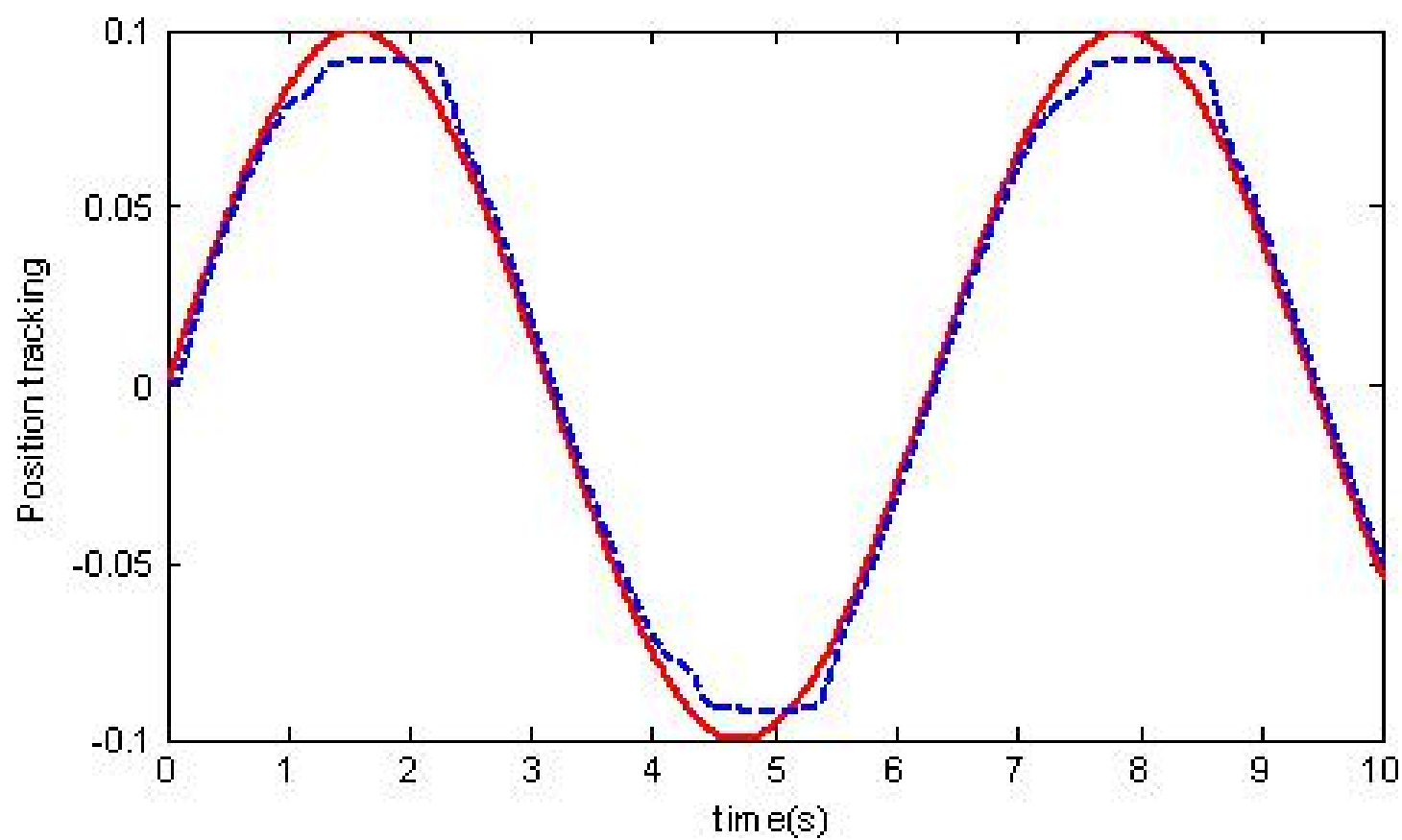


图10-13 未加补偿的PD控制位置跟踪

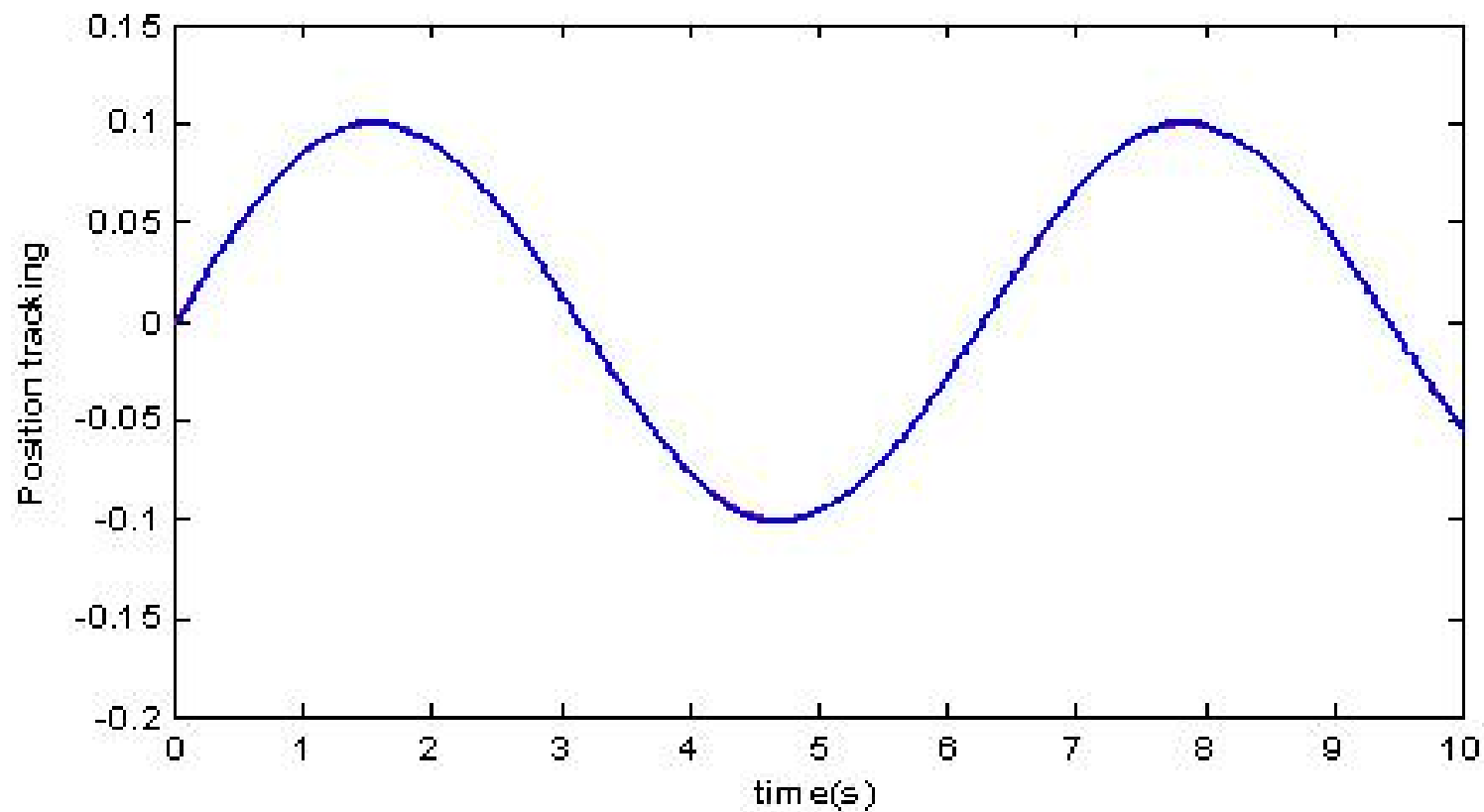


图10-14 基于摩擦补偿的PD控制位置跟踪

## 10.8 基于遗传算法的TSP问题优化

在第8.5节已经对旅行商问题进行了描述。遗传算法由于其全局搜索的特点，在解决TSP问题中有明显的优势。

### 10.8.1 TSP问题的编码

设 $D = \{d_{ij}\}$ 是由城市i和城市j之间的距离组成的距离矩阵，旅行商问题就是求出一条通过所有城市且每个城市只通过一次的具有最短距离的回路。

在旅行商问题的各种求解方法中，描述旅行路线的方法主要有如下两种：（1）巡回旅行路线经过的连接两个城市的路线的顺序排列；（2）巡回旅行路线所经过的各个城市的顺序排列。大多数求解旅行商问题的遗传算法是以后者为描述方法的，它们大多采用所遍历城市的顺序来表示各个个体的编码串，其等位基因为 $N$ 个整数值或 $N$ 个记号。

以城市的遍历次序作为遗传算法的编码，目标函数取路径长度。在群体初始化、交叉操作和变异操作中考虑TSP问题的合法性约束条件（即对所有的城市做到不重不漏）。

## 10.8.2 TSP问题的遗传算法设计

采用遗传算法进行路径优化，分为以下几步：

### 第一步：参数编码和初始群体设定

一般来说遗传算法对解空间的编码大多采用二进制编码形式，但对于TSP一类排序问题，采用对访问城市序列进行排列组合的方法编码，即某个巡回路径的染色体个体是该巡回路径的城市序列。

针对TSP问题，编码规则通常是N取进制编码，即每个基因仅从1到N的整数里面取一个值，每个个体的长度为N，N为城市总数。定义一个s行t列的pop矩阵来表示群体，t为城市个数N+1，即N+1，s为样本中个体数目。针对30个城市的TSP问题，t取值31，即矩阵每一行的前30个元素表示经过的城市编号，最后一个元素表示经过这些城市要走的距离。

参数编码和初始群体设定程序为：

```
pop=zeros(s,t);  
for i=1:s  
    pop(i,1:t-1)=randperm(t-1);  
end
```

## 第二步：计算路径长度的函数设计

在TSP的求解中，用距离的总和作为适应度函数，来衡量求解结果是否最优。将POP矩阵中每一行表示经过的距离的最后一个元素作为路径长度。

两个城市m和n间的距离为：

$$d_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \quad (10.16)$$

用于计算路径长度的程序为chap10\_1dis.m。

通过样本的路径长度可以得到目标函数和自适应度函数。

根据t的定义，两两城市组合数共有t-2组，则目标函数为：

$$J(t) = \sum_{j=1}^{t-2} d(j) \quad (10.17)$$

自适应度函数取目标函数的倒数，即：

$$f(t) = \frac{1}{J(t)} \quad (10.18)$$

### 第三步：计算选择算子

选择就是从群体中选择优胜个体、淘汰劣质个体的操作，它是建立在群体中个体适应度评估基础上。仿真中采用最优保存方法，即将群体中适应度最大的 $c$ 个个体直接替换适应度最小的 $c$ 个个体。选择算子函数为chap10\_7select.m。仿真中，取 $c=25$ 。

## 第四步：计算交叉算子

交叉算子在遗传算法中起着核心的作用，它是指将个体进行两两配对，并以交叉概率将配对的父代个体加以替换重组而生成新个体的操作。仿真中，取  $p_c = 0.90$

有序交叉法实现的步骤是：

步骤 1 随机选取两个交叉点 `crosspoint(1)` 和 `crosspoint(2)`；

步骤 2 两后代先分别按对应位置复制双亲 `x1` 和 `x2` 匹配段中的两个子串 `A1` 和 `B1`；

步骤 3 在对应位置交换双亲匹配段以外的城市，如果交换后，后代X1中的某一城市a与子串中A1的城市重复，则将该城市取代子串B1与A1中的城市a具有相同位置的新城市，直到与子串A1中的城市均不重复为止，对后代X2也采用同样方法，如图10-15所示：

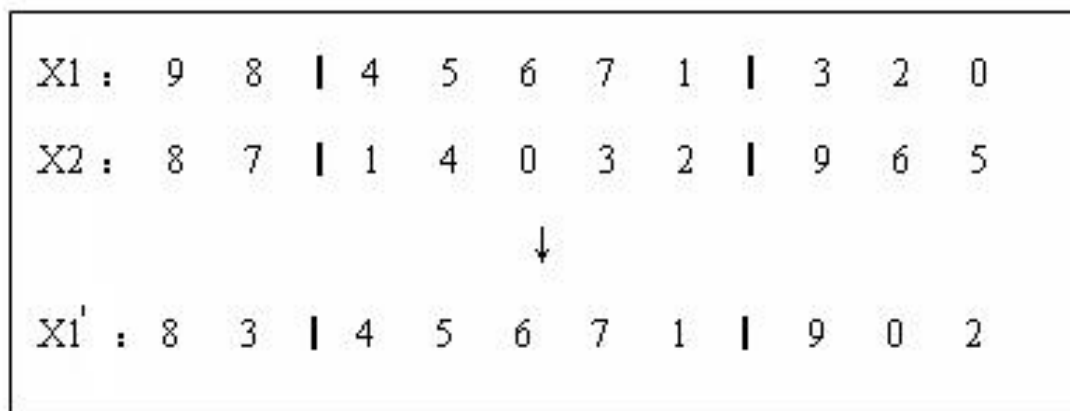


图10-15 有序交叉算子

从图10-15可知，有序交叉算子能够有效地继承双亲的部分基因成分，达到了进化过程中的遗传功能，使该算法并不是盲目搜索，而是趋向于使群体具有更多的优良基因，最后实现寻优的目的。交叉算子函数为chap10\_7corss.m

## 第五步：计算变异算子

变异操作是以变异概率 对群体中个体串某些基因位上的基因值作变动，若变异后子代的适应度值更加优异，则保留子代染色体，否则，仍保留父代染色体。仿真中，取 $p_m = 0.20$ 。

这里采用倒置变异法：假设当前个体X为(1 3 7 4 8 0 5 9 6 2 )，如果当前随机概率值 $< p_m$ ，则随机选择来自同一个体的两个点mutatepoint(1)和mutatepoint(2)，然后倒置该两个点的中间部分，产生新的个体。

例如，假设随机选择个体X的两个点“7”和“9”，则倒置该两个点的中间部分，即将“4805”变为“5084”，产生新的个体X为(1 3 7 5 0 8 4 9 6 2)。

变异算子函数为chap10\_7mutate.m。

### 10.8.3 仿真实例

以8个城市的路径优化为例，其城市路径坐标保存在路径e:\ljkmb\的程序cities8.txt中。遗传算法参数设定为：群体中个体数目  $s=100$  交叉概率  $p_c=0.90$  变异概率  $p_m=0.20$  通过改变进化代数K，观察不同进化代数下路径的优化情况，仿真结果见图10-16和图10-17所示，经过1000次进化，城市组合路径达到最小。最短路程为2.8937

仿真过程表明，在20次仿真实验中，有15次可收敛到最优解。

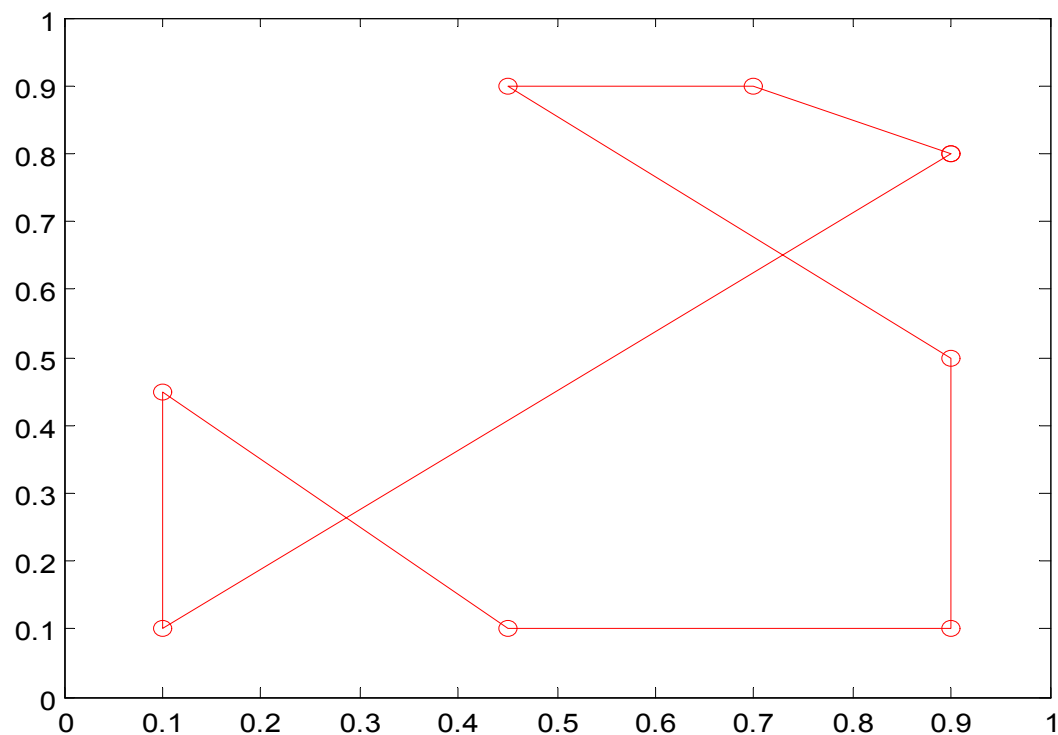


图10-16 进化代次数为300时的轨迹，距离  
 $L = 3.4173$

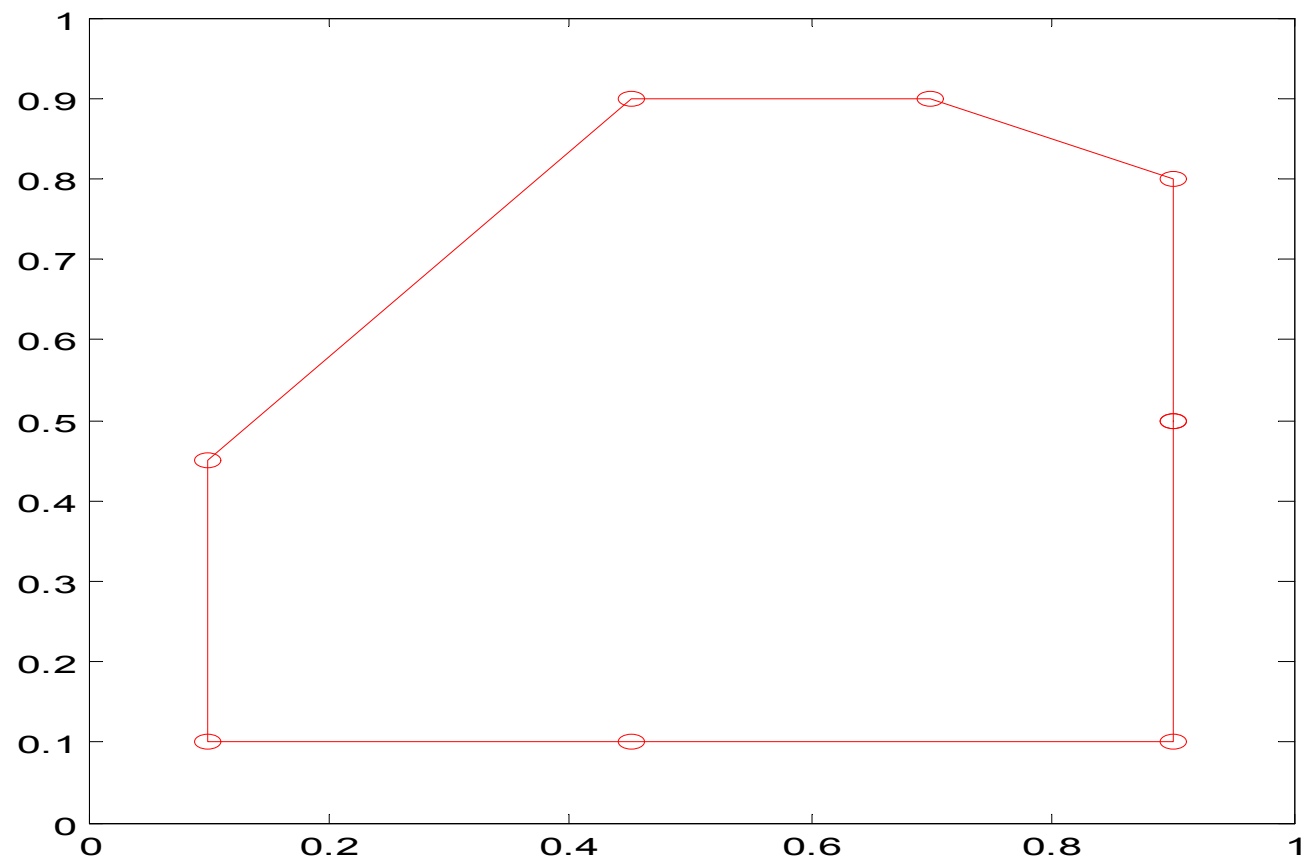


图10-17 进化代次数为500时的轨迹，距离  
 $L=2.8937$ 。