

# 一种 HPGL 插补控制算法仿真研究

黄刚

(徐州工程学院信电工程学院 徐州 221111)

**摘要** 为使 HPGL 格式文件输出的折线光滑化,分析了该格式文件数据输出存在的弊端及通常的解决办法。在此基础上,提出了一种基于遗传算法的插值处理方法,对该算法进行了数学描述,并对插值处理过程进行了数学解析。仿真实验结果表明,在对 HPGL 格式文件输出的折线进行若干次的插值处理过程中,只要将插值控制在设定的精度范围内,最终的运动轨迹就会形成一条光滑的曲线。

**关键词** HPGL,遗传算法,插值处理,MATLAB 仿真

**中图分类号** TH164 **文献标识码** A

## Simulative Study on HPGL Interpolation Control Algorithms

HUANG Gang

(Department of Information & Electronic Engineering, Xuzhou Institute of Technology, Xuzhou 221111, China)

**Abstract** To smooth the polyline outputted by HPGL format file, the existing disadvantages of data output and corresponding solutions were analyzed. An interpolation processing method based on genetic algorithms was raised, and the mathematical description of the algorithms was given, in addition, the mathematical analysis of the interpolation processing was elaborated. As shown in the simulative experiment, during several interpolation treatments of polyline outputted by HPGL format file, as long as the interpolation is controlled in accurate range, the final motion path would become a smooth curve.

**Keywords** HPGL, Genetic algorithms, Interpolation processing, MATLAB simulation

### 1 引言

目前大量的 CAD/CAM 软件均支持 HPGL(惠普图形语言)格式文件的数据输出,这种格式输出的数据主要以坐标点的方式出现,即将原始的曲线采用逼近的折线方式进行描述,易于实现,对于普通应用已经足够,但对于精确的应用系统而言,这种输出方式存在着不光滑的通病<sup>[1,2]</sup>,目前解决方法主要有以下两种:

①曲线插补运动控制方法,即对原始曲线(一般为 B 样条曲线)直接进行控制<sup>[3]</sup>。这种方法虽彻底,但曲线底层的运动控制系统需要支持样条曲线的解释及运动控制。同时,还需要在上位机软件的标准接口下运行,算法复杂且运算量大,因此难以实现。

②提高折线逼近的精度方法,即将曲线分解的更精细的方法<sup>[4]</sup>。这种方法需要在 CAD/CAM 软件上进行改进,无法统一实现。即使实现了,也会导致数据量的剧增。

基于上述原因,本文提出一种基于遗传算法的插值处理方法,该方法能直接在运动控制系统底层快速地原始曲线实现精细化,即对 HPGL 格式文件输出的折线进行若干次的插值处理,最终使输出的曲线与原始的曲线完全拟合。

### 2 算法的数学描述

HPGL 节点图如图 1 所示。连接  $P_1, P_4$  并取其中点

$P_{1,4}$ , 连接  $P_2, P_3$  并取其中点  $P_{2,3}$ , 连接  $P_{1,4}, P_{2,3}$ , 并将其按一定比例  $k$  延长至  $A_1$  点, 则  $A_1$  点为  $P_2, P_3$  间的第一个插值点。同理, 可得  $A_2, A_3 \dots$ 。第一次插值完成后,  $A_1, A_2, A_3 \dots$  加入到  $P_1, P_2, P_3 \dots$  中, 统一作为新的节点, 节点数量增一倍, 若需要, 则进行下一轮的插值计算...

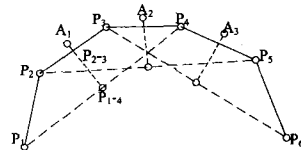


图1 HPGL 节点图

经无数次的插值后,在数学上就形成了一条连续光滑的曲线。当  $k \leq 0.25$  时,该曲线为  $N$  阶可导,对应运动轨迹而言,其速度、加速度、加速度变化率等均连续、完全光滑。而光滑曲线的运动可以完全避免在各原始节点处( $P_1, P_2, P_3 \dots$ )的运动方向突变引起的抖动,更利于运行速度的提高,同时提高运动及切割质量。

### 3 比例系数 k 的确定

在本文提出的 HPGL 插补控制算法中,比例系数  $k$  的大小对算法输出结果影响很大,以图 1HPGL 节点图为例,通过采用群体搜索策略和简单遗传算子的全局优化算法—遗传算法(Genetic Algorithm, GA)来计算比例系数  $k$  的最优值。

到稿日期:2010-03-19 返修日期:2010-06-29 本文受江苏省高校自然科学基金(09KJD120005)资助。

黄刚(1976-),男,硕士,讲师,主要研究方向为计算机算法、计算机应用, E-mail: gangzi@xzit.edu.cn.

### 3.1 遗传算法简介

遗传算法是一种宏观意义上的仿生算法,它模仿的机制是一切生命与智能的产生与进化过程。它以自然选择和遗传理论为基础,通过模拟达尔文“优胜劣汰、适者生存”的原理鼓励产生好的结构,通过模仿孟德尔遗传变异理论在迭代过程中保持已有的结构,同时寻找更好的结构,是一种将生物进化过程中适者生存规则与群体内部染色体的随机信息交换机制相结合的高效全局寻优搜索算法<sup>[5-8]</sup>。GA 摒弃了传统的搜索方式,模拟自然界生物进化过程,采用人工的方式对目标空间进行随机优化搜索。它将问题域中的可能解看作是群体的一个个体或染色体,并将每一个个体编码成符号串形式,模拟遗传选择、自然淘汰以及基因变异的生物进化过程,对群体反复进行基于遗传学的操作(遗传、交叉与变异)。根据预定的目标适应度函数对每个个体进行评价,依据适者生存、优胜劣汰的进化原则,不断得到更优的群体,同时不依赖于梯度信息,而以全局并行搜索方式来搜索优化群体中的最优个体,以求得满足要求的最优解。

基本遗传算法的算法流程如图 2 所示<sup>[9-12]</sup>。

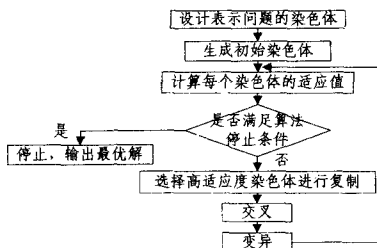


图 2 基本遗传算法流程图

### 3.2 仿真实验结果

以图 1HPGL 节点图为例,插值次数分别为 3,4,5,遗传算法交叉概率  $P_m=0.8$ ,变异概率  $P_r=0.01$ ,最大迭代次数为 200,仿真实验环境为 MATLAB Pentium M 1.60GHz CPU 760M RAM,仿真实验结果如图 3—图 5 所示。

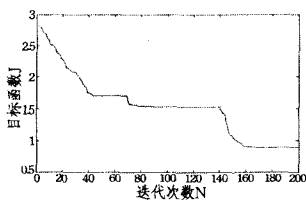


图 3 插值次数为 3 时算法收敛曲线图

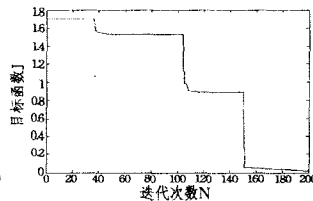


图 4 插值次数为 4 时算法收敛曲线图

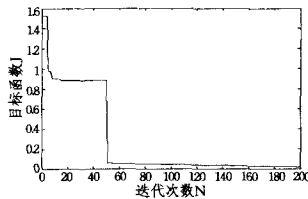


图 5 插值次数为 5 时算法收敛曲线图

由图 3—图 5 综合考虑,建议比例系数  $k$  取 0.125。

## 4 算法的数学解析

### 4.1 第一次插补

以  $P_1-P_4$  点为插补节点,定义  $A_1$  点为  $P_2P_3$  线段的中点, $B_1$  点为  $P_1P_4$  线段的中点,则由中点坐标公式可求出  $A_1$  和  $B_1$  两点的坐标分别为:

$$x_{a1} = \frac{x_2 + x_3}{2}, y_{a1} = \frac{y_2 + y_3}{2}$$

$$x_{b1} = \frac{x_1 + x_4}{2}, y_{b1} = \frac{y_1 + y_4}{2}$$

连接线段 BA 并将其延长至  $C_1$  点,使得  $A_1C_1/B_1A_1=1/8$ ,对应放大系统数  $k$  取值为 0.125,由线段的定比分点公式可得  $C_1$  点坐标为:

$$x_{c1} = \frac{9(x_2 + x_3) - x_1 - x_4}{16}, y_{c1} = \frac{9(y_2 + y_3) - y_1 - y_4}{16}$$

同理,以  $P_2, P_5, P_3, P_4$  为插补节点,插补出点  $C_2$ ,其坐标公式为:

$$x_{c2} = \frac{9(x_3 + x_4) - x_2 - x_5}{16}, y_{c2} = \frac{9(y_3 + y_4) - y_2 - y_5}{16}$$

以  $P_3, P_6, P_4, P_5$  为插补节点,插补出  $C_3$  点,其坐标公式为:

$$x_{c3} = \frac{9(x_4 + x_5) - x_3 - x_6}{16}, y_{c3} = \frac{9(y_4 + y_5) - y_3 - y_6}{16}$$

具体插补过程如图 6 所示。

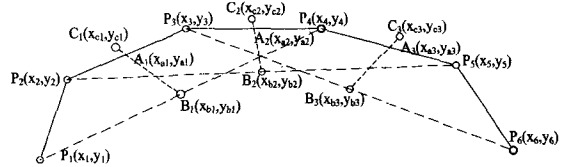


图 6 HPGL 算法第一次插补

### 4.2 第二次插补

第二次插补以加入第一次插补的点作为节点,对于起始第一段线段的插补点,以其中点为节点进行计算,而第一个线段不进行 HPGL 插补。

由于第一条线段需要作为节点,就会出现特殊情况,即对于第二条线段的第二次插补会没有插补中点,因此对于第二条线段的第二次插补中点采用第一条线段的中点进行插补。

第二条线段开始,插补节点变为第一段线段中点  $O, P_2, C_1, P_3$ 。其中  $O$  点为  $P_1P_2$  的中点,其坐标为  $(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$ 。定义  $E_1$  点为线段  $P_2C_1$  的中点, $D_1$  点为  $OP_3$  线段的中点,第二次插补后点为  $F_1$ 。

同样求  $E_1$  和  $D_1$  两点坐标分别为:

$$x_{e1} = \frac{9(x_2 + x_3) - x_1 - x_4}{16} + \frac{x_2}{2} = \frac{25x_2 + 9x_3 - x_1 - x_4}{32}$$

$$y_{e1} = \frac{9(y_2 + y_3) - y_1 - y_4}{16} + \frac{y_2}{2} = \frac{25y_2 + 9y_3 - y_1 - y_4}{32}$$

$$x_{d1} = \frac{\frac{x_1 + x_2}{2} + x_3}{2} = \frac{x_1 + x_2 + 2x_3}{4}$$

$$y_{d1} = \frac{\frac{y_1 + y_2}{2} + y_3}{2} = \frac{y_1 + y_2 + 2y_3}{4}$$

连接  $D_1E_1$ ,并按同样的放大系数  $k$  延长至  $F_1$  点,用同样的方式可求出其坐标为:

$$x_{f1} = \frac{9(25x_2 + 9x_3 - x_1 - x_4) - 8(x_1 + x_2 + 2x_3)}{256}$$

$$= \frac{-17x_1 + 217x_2 + 65x_3 - 9x_4}{256}$$

$$y_{f1} = \frac{9(25y_2 + 9y_3 - y_1 - y_4) - 8(y_1 + y_2 + 2y_3)}{256}$$

$$= \frac{-17y_1 + 217y_2 + 65y_3 - 9y_4}{256}$$

将插补节点变为  $P_2, C_2, C_1, P_3$ , 进行计算插补。定义  $E_2$  点为线段  $C_1P_3$  的中点,  $D_2$  点为  $P_2C_2$  线段的中点, 第二次插补后点为  $F_2$ 。

同样求  $E_2, D_2$  点坐标分别为:

$$x_{E2} = \frac{9(x_2 + x_3) - x_1 - x_4 + 16x_3}{32} = \frac{25x_3 + 9x_2 - x_1 - x_4}{32}$$

$$y_{E2} = \frac{9(y_2 + y_3) - y_1 - y_4 + 16y_3}{32} = \frac{25y_3 + 9y_2 - y_1 - y_4}{32}$$

$$x_{D2} = \frac{9(x_3 + x_4) - x_2 - x_5 + 16x_2}{32} = \frac{9x_3 + 9x_4 + 15x_2 - x_5}{32}$$

$$y_{D2} = \frac{9(y_3 + y_4) - y_2 - y_5 + 16y_2}{32} = \frac{9y_3 + 9y_4 + 15y_2 - y_5}{32}$$

定义  $F_3$  在  $A_3B$  线段往  $A_3$  点延长  $k$  点, 连接  $D_2E_2$ , 并按同样的放大系数  $k$  延长至  $F_2$  点, 用同样的方式可求出其坐标为:

$$x_{F2} = \frac{9(25x_3 + 9x_2 - x_1 - x_4) - (9x_3 + 9x_4 + 15x_2 - x_5)}{256} = \frac{216x_3 + 66x_2 - 9x_1 - 18x_4 + x_5}{256}$$

$$y_{F2} = \frac{9(25y_3 + 9y_2 - y_1 - y_4) - (9y_3 + 9y_4 + 15y_2 - y_5)}{256} = \frac{216y_3 + 66y_2 - 9y_1 - 18y_4 + y_5}{256}$$

同理可得  $P_3C_2, C_2P_4, P_1C_3, C_3P_5$  之间的插值点  $F_3, F_4, F_5$  和  $F_6$  的坐标。具体插补过程如图 7 所示。

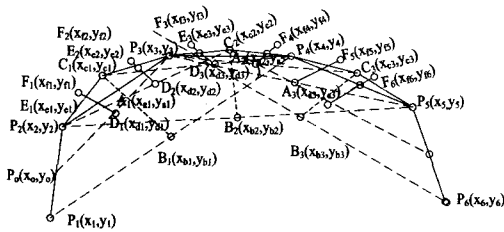


图 7 HPGL 算法第二次插补

在实际程序处理时, 为加快程序处理效率, 每一次插补取 5 个节点  $P_1 \sim P_5$ , 进行 3 个插值点计算。根据以上数学解析过程及结论, 在 MATLAB 上进行仿真后实现算法。仿真流程图如图 8 所示。

仿真结果如图 9 所示。

**结束语** 为使 HPGL 格式文件输出的折线光滑化, 提出一种基于遗传算法的 HPGL 插补控制算法, 其通过遗传算法确定最优的比例系数  $k$ 。仿真实验结果表明, 在对 HPGL 格式文件输出的折线进行若干次的插值处理过程中, 只要将插值控制在设定的精度范围内, 最终输出的曲线就会完全光滑。

## 参考文献

- [1] 杨开明, 石川, 叶佩清, 等. 数控系统轨迹段光滑衔接控制算法[J]. 清华大学报: 自然科学版, 2007, 47(8): 1295-1299
- [2] 叶伟, 王小椿. 一种连续小线段高速插补算法[J]. 南京理工大学学报: 自然科学版, 2008, 32(4): 443-448
- [3] 邓四清, 方遼, 等. 基于函数值的有理四次样条曲线的区域控制[J]. 计算机工程与应用, 2008, 44(20): 192-195
- [4] 张志强. 数控系统参数曲线、曲面插补算法及加减速控制研究[D]. 天津: 天津大学, 2008
- [5] Holland J H. Adaptation in natural and artificial systems[M]. Arbor: University of Michigan Press, 1975
- [6] Holland J H. Adaptation in natural and artificial systems[M]. Ann Arbor: University of Michigan Press, 1975
- [7] Dejong KA. The analysis of the behavior of a class of genetic adaptive systems[D]. Ann Arbor: University of Michigan, 1975
- [8] Oldberg G E. Genetic algorithms in search, optimization and machine learning[M]. Boston: Addison Wesley Longman Press, 1989
- [9] Fogel D B. An introduction to simulated evolutionary optimization[J]. IEEE Trans. Neural Networks, 1994, 5(1)
- [10] Goldberg D E. Genetic algorithm in search, optimization and machine learning[M]. Addison-Wesley Publishing Company Inc., 1985
- [11] Qin Ming-hao, Xu Ye-yi. Dynamic analysis and optimization design of six-rod mechanism in high-speed punching press[C]// The International Conference on Mechanical Dynamics, Shenyang, China, 1987
- [12] 谢安世, 周传华, 等. 基于 PK 模型的一种自适应遗传算法研究[J]. 计算机工程与应用, 2010, 46(7): 52-56
- [13] 田玉柱, 王丙参, 冉延平, 等. 基于遗传算法的广义指数分布参数估计[J]. 重庆工学院学报: 自然科学版, 2009, 23(10): 157-159

## 5 算法的仿真实现

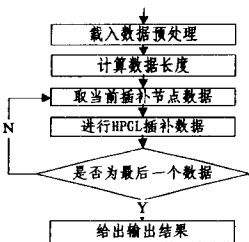


图 8 HPGL 算法流程图

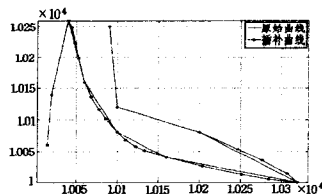


图 9 HPGL 算法仿真结果图

(上接第 245 页)

- [9] Zhang Li, Cui Yuquan. Outer P-sets and data internal-recovery[J]. An International Journal Advances in Systems Science and Applications, 2010, 10(2): 189-199
- [10] Huang Shunliang, Wang Wei, Geng Dianyou. P-sets and its internal p- memory characteristics[J]. An International Journal Advances in Systems Science and Applications, 2010, 10(2): 216-222
- [11] 李豫颖, 谢维奇, 史开泉.  $F$ -残缺数据的辨识与恢复[J]. 山东大学学报: 理学版, 2010, 45(9): 57-64
- [12] 周玉华, 张冠宇, 史开泉. P-集合与双信息规律生成[J]. 数学的实践与认识, 2010, 40(13): 71-80
- [13] 周玉华, 张冠宇, 张丽. 内外数据圆与动态数据-恢复[J]. 山东大学学报: 理学版, 2010, 45(8): 21-26
- [14] 张丽, 崔玉泉, 史开泉. 外 P-集合与数据内-恢复[J]. 系统工程与

- 电子技术, 2010, 32(6): 1233-1238
- [15] 于秀清. P-集合的识别与筛选[J]. 山东大学学报: 理学版, 2010, 45(1): 94-98
- [16] 张飞, 陈萍, 张丽. P-集合的 P-分离与应用[J]. 山东大学学报: 理学版, 2010, 45(3): 18-22
- [17] 汤积华, 陈保会, 史开泉. P-集合与  $(F, F)$ -数据生成-辨识[J]. 山东大学学报: 理学版, 2009, 44(11): 83-92
- [18] Zhang Li, Xu Ming, Shi Kaiquan. P-sets and applications of internal outer data circle[J]. Quantitative Logic and Soft Computing, 2010, 2: 581-591
- [19] Qiu Yufeng, Chen Baohui. f-model generated by P-sets[J]. Quantitative Logic and Soft Computing, 2010, 2: 613-620
- [20] Li Yuying, Zhang Li, Shi Kaiquan. Generation and recovery of compressed data and redundant data[J]. Quantitative Logic and Soft Computing, 2010, 2: 661-671