数控技术上机实验报告

机械97 杨逢诜

**一、上机实验目的：**

上机实验的目的包括以下几点：

1.掌握MATLAB基本函数、语法、循环结构等一般算法环节的编制操作；

2.能够根据已知的MATLAB语法编制程序；

3.掌握逐点比较法、数据采样法两种常见插补算法的基本原理和实现方法；

4.明确逐点比较法、数据采样法在临界情况（F=0、过象限、过45°线）下的程序处理方法，知晓两种算法的基本区别；

5.了解插补算法误差的来源和插补算法精度的提高方法。

**二、上机实验的器材、语言类型和编程任务：**

（一）上机实验的器材：

上机实验的器材仅包括一台已安装MATLAB2021的笔记本计算机，以及一个有效的、可以完成数据转移功能的U盘。

（二）上机实验的语言类型和编程任务：

1.**上机实验的语言类型：**

本次上机采用MATLAB语言进行程序编制和调试，其代码平台为MATLAB自带的编译器。

**2.本次上机的编程任务：**

本次上机的编程任务包含以下三个插补程序的编制：

1）编写逐点比较法平面任一位置的直线插补程序，要求可设定起止位置和脉冲当量值、可以显示理论曲线和插补过程。

2）编写逐点比较法顺时针整圆插补程序，要求可以设定起始位置和圆心坐标、显示理论曲线和插补过程。

3）编制数据采样法逆时针整圆插补程序，要求可以设定起始位置和圆心坐标、显示理论曲线和插补过程。

**三、数控技术插补原理：**

**（一）逐点比较法：**

逐点比较法的基本原理在于，通过判断刀具实际位置与理论位置的相对方位，从而对刀具的进一步运动规律进行决策，这种决策往往需要依赖一定的判定函数。注意到：在逐点比较法中，刀具的运动方向只有+X、-X、+Y、-Y四种。显然地，这种插补算法并不能体现出进给轴的联动。

对于直线而言，逐点比较法的判别准则和运动决策模式为：





对于圆弧（顺时针插补）而言，逐点比较法的判别准则和运动决策模式为：





**（二）数据采样法：**

数据采样法基于这样的原理进行：依据刀具的当前位置和指定的进给速度、电机脉冲周期，推算出在下一次控制脉冲信号到达电机后刀具应当到达的位置，并计算相应各个进给轴的位移；根据计算得到的位移分别向相应的进给轴发送对应的脉冲信号使之执行相应的位移操作；循环进行这种操作直至刀具到达路径终点。这种插补算法可以有效地体现进给轴联动的特性。

注意到：对于直线插补而言，数据采样法往往能够保证刀具的理论路径与插补路径之间的一致性；而对于圆弧插补而言，数据采样法采用弦线逼近圆弧的方法使得刀具理论路径与实际插补路径之间保持一定的近似性。

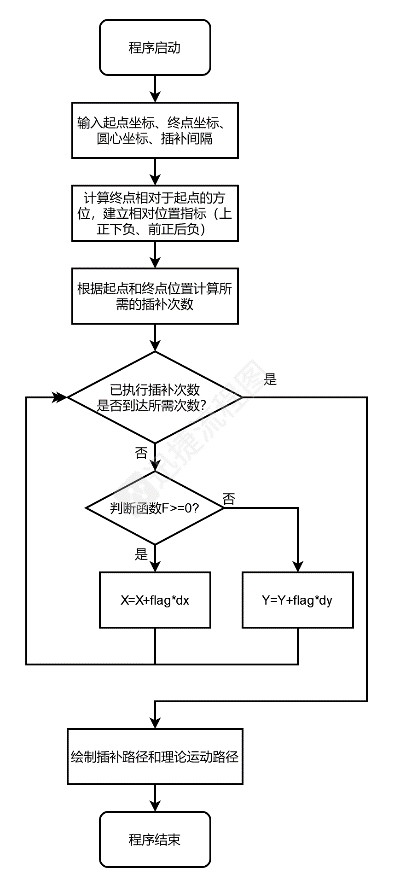
**四、插补程序编制原理：**

我们指出：在本章中，为力求简便，我们采取了以下两种规定以尽可能简化程序：①以简便原则重建插补坐标系：对于直线插补而言，我们基于直线起点建立坐标系；对于圆弧插补而言，我们基于圆弧圆心建立坐标系；②以简并原则考虑整个算法：将全部可以归纳具有统一形式的插补算法打包进行判断和分析，如：对于顺时针圆弧逐点比较法插补中的I,III象限合并处理、II、IV象限合并处理。基于上述两个原则，编制本次上机操作中的全部程序，并将程序框图置于下方以供参考：

（一）逐点比较法程序编制原理：

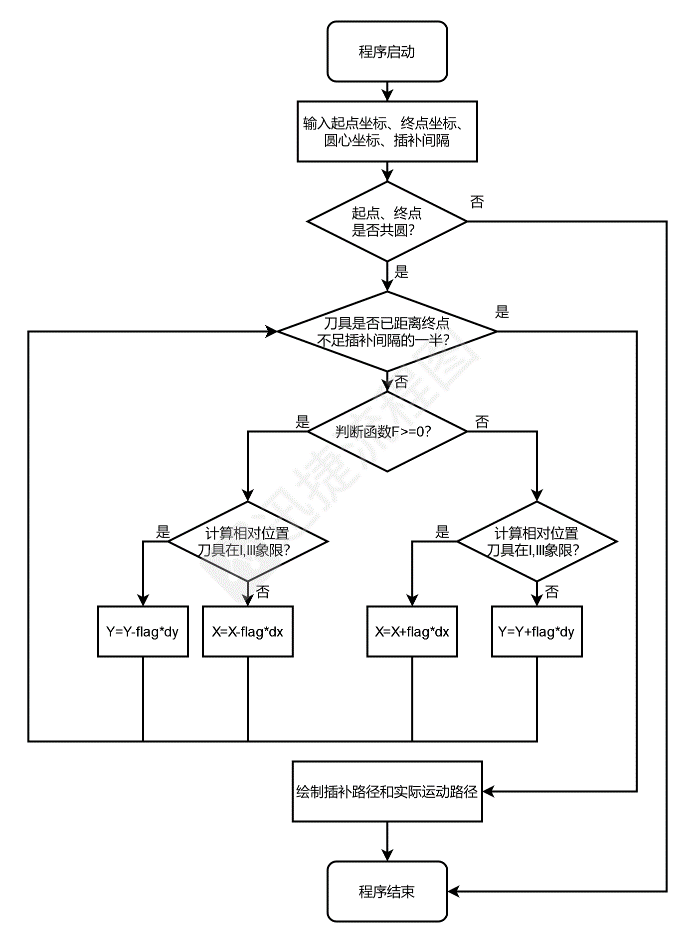
**1. 基于逐点比较法的直线插补程序原理：**

该插补程序的流程框图如下图所示：



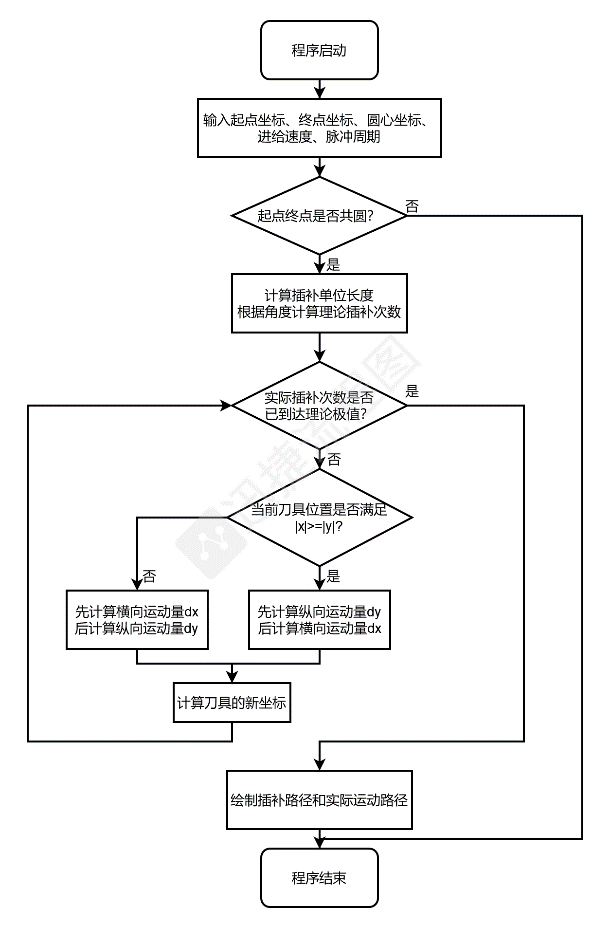
**2. 基于逐点比较法的顺时针圆弧插补程序：**

该插补程序的流程框图如下图所示：



（二）数据采样法程序编制原理：

基于数据采样法的逆时针圆弧插补算法之流程框图如下图所示：



**五、插补程序及运行实例：**

（一）插补程序形式：

依照插补程序编制原理和程序编制任务，得到以下三个MATLAB插补程序：

**1.基于逐点比较法的直线插补程序（G01代码原理）：**

clear;clc;

x0=input('请输入起点横坐标');

y0=input('请输入起点纵坐标');

x1=input('请输入终点横坐标');

y1=input('请输入终点纵坐标');

delta=input('请输入插补间隔');

sgn\_x=(x1-x0)/abs(x1-x0);

sgn\_y=(y1-y0)/abs(y1-y0);

n=abs(x1-x0)/delta+abs(y1-y0)/delta;

xn=ones(1,n+1)\*x0;

yn=ones(1,n+1)\*y0;

x=x0;y=y0;nc=1;

while nc<=n&&x1-x0~=0

nc=nc+1;

if abs((x1-x0)\*(y-y0))-abs((x-x0)\*(y1-y0))>=0

x=x+delta\*sgn\_x;

else

y=y+delta\*sgn\_y;

end

yn(nc)=y;

xn(nc)=x;

end

if x1-x0==0

yn=sgn\_y\*delta\*(0:n)+y0;

end

p1=plot([x0,x1],[y0,y1],'r-');

hold on

axis on

grid on

p2=plot(xn,yn,'b-');

title('逐点比较法直线插补程序示例')

xlabel('x轴坐标x\_{t}/mm')

ylabel('y轴坐标y\_{t}/mm')

legend([p1,p2],'Theoretical Routine','Interpolation Routine')

%-----------------------------------%

%原作者：机械97 杨逢诜 2193712613

%禁止在合理使用范围外使用此代码

%软件著作权所有，抄袭必究

**2.基于逐点比较法的顺时针圆弧插补程序（G02代码原理）：**

clear;clc;

x0=input('请输入起点横坐标');

y0=input('请输入起点纵坐标');

x1=input('请输入终点横坐标');

y1=input('请输入终点纵坐标');

xo=input('请输入圆心横坐标');

yo=input('请输入圆心纵坐标');

r=sqrt((x0-xo)^2+(y0-yo)^2);

if r-sqrt((x1-xo)^2+(y1-yo)^2)~=0

fprintf('无效输入：两点不能对定圆心共圆');

return;

end

delta=input('请输入插补间隔');

x=x0;y=y0;nc=1;

while sqrt((x-x1)^2+(y-y1)^2)>delta/3

yn(nc)=y;

xn(nc)=x;

if sqrt((x-xo)^2+(y-yo)^2)-r>=0

if signal(x-xo)\*signal(y-yo)>=0

y=y-delta\*signal(y-yo);

else

x=x-delta\*signal(x-xo);

end

else

if signal(x-xo)\*signal(y-yo)>=0

x=x+delta\*signal(y-yo);

else

y=y-delta\*signal(x-xo);

end

end

nc=nc+1;

end

syms u v t

u=xo+r\*cos(t);

v=yo+r\*sin(t);

if angelCalculate(xo,yo,x1,y1)<angelCalculate(xo,yo,x0,y0)

p1=fplot(u,v,[angelCalculate(xo,yo,x1,y1),angelCalculate(xo,yo,x0,y0)],'r-');

else

p1=fplot(u,v,[angelCalculate(xo,yo,x1,y1),2\*pi+angelCalculate(xo,yo,x0,y0)],'r-');

end

hold on

axis on

grid on

p2=plot(xn,yn,'b-');

title('逐点比较法圆弧插补程序示例')

xlabel('x轴坐标x\_{t}/mm')

ylabel('y轴坐标y\_{t}/mm')

legend([p1,p2],'Theoretical Routine','Interpolation Routine')

%-----------------------------------%

%原作者：机械97 杨逢诜 2193712613

%禁止在合理使用范围外使用此代码

%软件著作权所有，抄袭必究

**3.基于数据采样法的逆时针圆弧插补程序（G03代码原理）：**

clear;clc;

x0=input('请输入起点横坐标');

y0=input('请输入起点纵坐标');

x1=input('请输入终点横坐标');

y1=input('请输入终点纵坐标');

xo=input('请输入圆心横坐标');

yo=input('请输入圆心纵坐标');

r=sqrt((x0-xo)^2+(y0-yo)^2);

if r-sqrt((x1-xo)^2+(y1-yo)^2)~=0

fprintf('无效输入：两点不能对定圆心共圆');

return;

end

F=input('请输入进给速率');

t=input('请输入脉冲周期');

deltal=F\*t/(60\*1000);

if angelCalculate(xo,yo,x1,y1)>angelCalculate(xo,yo,x0,y0)

n=abs(angelCalculate(xo,yo,x1,y1)-angelCalculate(xo,yo,x0,y0))/(2\*asin(deltal/r/2));

else

n=abs(angelCalculate(xo,yo,x0,y0)-2\*pi-angelCalculate(xo,yo,x1,y1))/(2\*asin(deltal/r/2));

end

x=x0;y=y0;nc=1;

dx=0;dy=0;dx0=0;dy0=0;

while nc<=n+1

yn(nc)=y;

xn(nc)=x;

nc=nc+1;

theta=angelCalculate(xo,yo,x,y);

if tan(theta)>=-1&&tan(theta)<=1

dy=deltal\*(x-dx0/2)/r;

y=y+dy;

if abs(y-yo)>r

y=yo+signal\_ds(y-yo)\*r;

end

x=sqrt(r^2-(y-yo)^2)\*signal(x-xo)+xo;

else

dx=deltal\*(y+dy0/2)/r;

x=x-dx;

if abs(x-xo)>r

x=xo+signal\_ds(x-xo)\*r;

end

y=sqrt(r^2-(x-xo)^2)\*signal(y-yo)+yo;

end

dx0=(x-xn(nc-1));

dy0=(y-yn(nc-1));

end

yn(nc)=y1;

xn(nc)=x1;

syms u v t

u=xo+r\*cos(t);

v=yo+r\*sin(t);

if angelCalculate(xo,yo,x1,y1)>angelCalculate(xo,yo,x0,y0)

p1=fplot(u,v,[angelCalculate(xo,yo,x0,y0),angelCalculate(xo,yo,x1,y1)],'r-');

else

p1=fplot(u,v,[angelCalculate(xo,yo,x0,y0),2\*pi+angelCalculate(xo,yo,x1,y1)],'r-');

end

hold on

axis on

grid on

p2=plot(xn,yn,'b--');

title('数据采样法圆弧插补程序示例')

xlabel('x轴坐标x\_{t}/mm')

ylabel('y轴坐标y\_{t}/mm')

legend([p1,p2],'Theoretical Routine','Interpolation Routine')

%-----------------------------------%

%原作者：机械97 杨逢诜 2193712613

%禁止在合理使用范围外使用此代码

%软件著作权所有，抄袭必究

**4.插补代码辅助函数：符号函数sgn()**

\*符号函数的机制：对于正输入或零输入而言，输出正数；对于负输入而言，输出负数。这一函数主要用于对兼并后的插补情形进行细分判别，如：圆弧插补中

function tmp=signal(u)

if u>=0

tmp=1;

else

tmp=-1;

end

**5.插补代码辅助函数：角度计算函数AngelCalculate()**

\*角度计算函数的机制：对于给定起止点的线段，在[， ]范围内计算其与x轴正方向的夹角，用于判定起止点的角位置。

function theta=angelCalculate(x0,y0,x1,y1)

u=(y1-y0)/sqrt((x1-x0)^2+(y1-y0)^2);

v=(x1-x0)/sqrt((x1-x0)^2+(y1-y0)^2);

if u>0

theta=acos(v);

else

theta=-acos(v);

end

（二）插补程序的运行结论：

基于上述三个插补程序主函数和两个辅助函数对插补程序进行调试检查，运行检验结论如下：

\*在本节中，理论曲线全部以红色实线形式呈现，实际插补曲线全部以蓝色实线呈现；但由于G03插补历程中理论曲线-实际曲线重合度较高，因此在G03校验过程中实际插补曲线以蓝色虚线呈现。

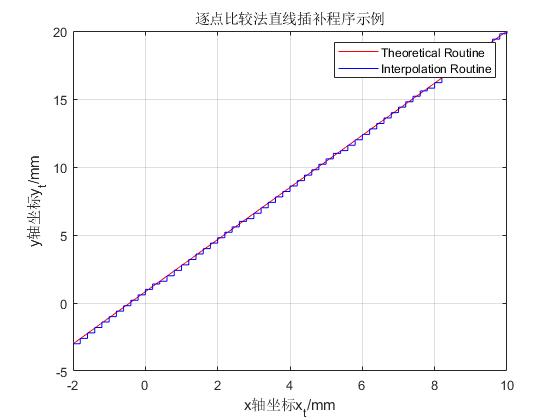
**1.基于逐点比较法的直线插补程序之检验（G01代码原理）：**

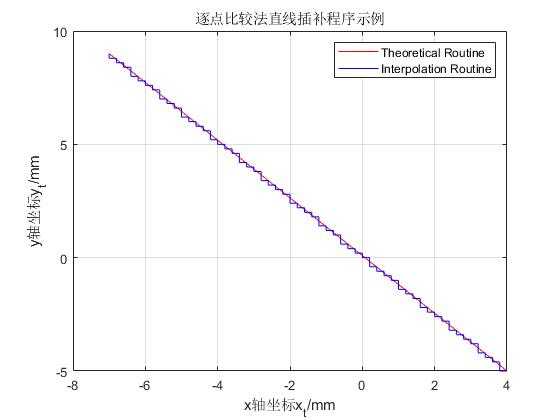
依次运行如下两个测试样例：

①由（-2，-3）插补至（10，20），插补间隔0.2；

②由（4，-5）插补至（-7，9），插补间隔0.2；

运行结论如下图所示：





两个较复杂的测试样例表明，程序运行结果符合预期，能够完成指定要求。程序**检验合格。**

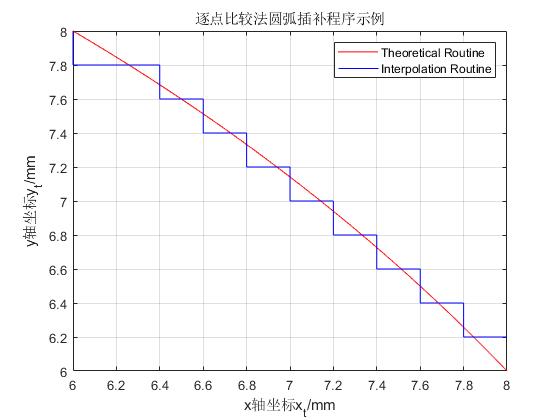
**2.基于逐点比较法的顺时针圆弧插补程序之检验（G02代码原理）：**

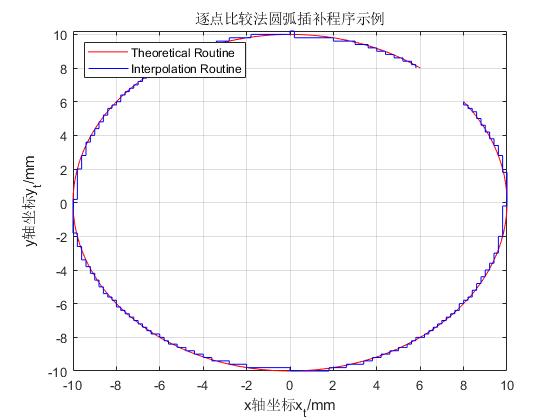
依次运行如下两个测试样例：

①由（6，8）插补至（8，6），圆心（0，0），插补间隔0.2；

②由（8，6）插补至（6，8），圆心（0，0），插补间隔0.2；

运行结论如下图所示：





两个全面覆盖复杂情况的测试样例表明，程序运行结果符合预期，能够完成指定要求。**程序检验合格。**

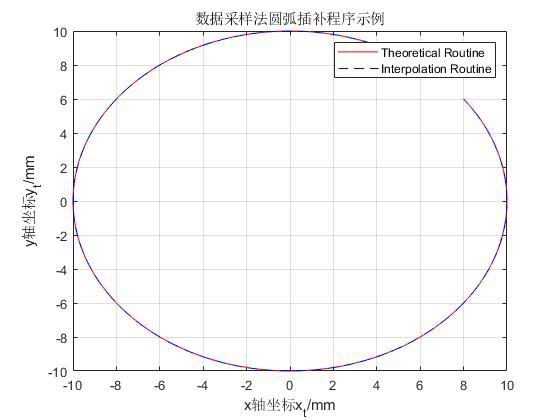
**3.基于数据采样法的逆时针圆弧插补程序之检验（G03代码原理）：**

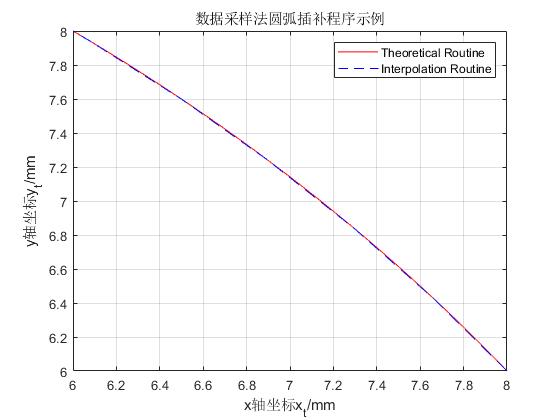
依次运行如下两个测试样例：

①由（6，8）插补至（8，6），圆心（0，0），脉冲周期20，进给速度1600；

②由（8，6）插补至（6，8），圆心（0，0），脉冲周期20，进给速度1600；

运行结论如下图所示：





两个全面覆盖复杂情况的测试样例表明，程序运行结果符合预期，能够完成指定要求。**程序检验合格。**

**六、插补程序误差分析**

注意到：在数控机床刀路插补过程中总是存在刀具插补运动轨迹与理论运动轨迹不重合的现象，这将在一定程度上降低数控机床的加工精度和零件的表面质量、形位误差精度。本节中将对编制的三个程序之插补路径误差进行分析：

（一）插补算法的固有误差：

显然地，根据逐点比较法进行插补产生的误差是相当大的——即使是采用了同等的插补间隔，其误差也显著大于数据采样法。这种误差主要来源于算法本身。注意到：逐点比较法中，刀具的运动轨迹相对于坐标轴不是平行就是正交，而理论路径的形态往往既非平行也非正交——这种力求保证位置精度而不保证速度方向精度的方法最终导致两个精度都无法完全兼顾，更何况这种插补的本质仍然是一种“事后补救”。

数据采样发的固有误差则相对而言要小很多——这一显著特征可以通过插补路径图样看出。这源于算法本身对于速度方向精度和位置精度的同时控制。用弦线近似代替切线大大降低了刀路的相对偏差——这一偏差的数量级是O(x3)，显著大于逐点比较法O(x)级别的精度。

（二）程序的数值计算误差：

程序的数值计算误差源于计算机处理离散数据过程中不可避免的舍入误差和计算机选择数值计算方法时该方法产生的误差。在本程序中，未显著观测到程序的数值计算误差——这可能与更高的单数据存储空间有关。实践中观测到，该数值计算误差远小于插补算法的固有误差，因此本节将不再详细陈述这一误差的量级。

**七、结论**

**1.完成了G01、G02、G03三种G代码对应插补算法的程序编制任务，经过检验指出三个插补程序分别符合相应的上机操作要求；**

**2.对逐点比较法、数据采样法两种插补算法的误差进行比较，并分析数控机床插补程序误差之来源。**