轨迹生成算法实验

1. 实验目的

1. 理解贝塞尔曲线原理，掌握贝塞尔曲线轨迹生成的Matlab程序实现方法
2. 理解B-样条定义与设计原理，掌握B-样条轨迹生成的Matlab程序实现方法
3. 通过调整控制点向量和节点向量观察轨迹的变化
4. 分析控制点和节点对B-样条轨迹设计的影响

2. 实验要求

- 软件要求：Windows 10及以上版本；MATLAB R2022b及以上版本[1]。
- 硬件要求：笔记本/台式电脑1台[2]。

3.实验地址

例程目录：
[安装目录]\RflySimAPIs\5.RflySimFlyCtrl\1.BasicExps\e10-FixedWingCtrl\code_8\e6-2

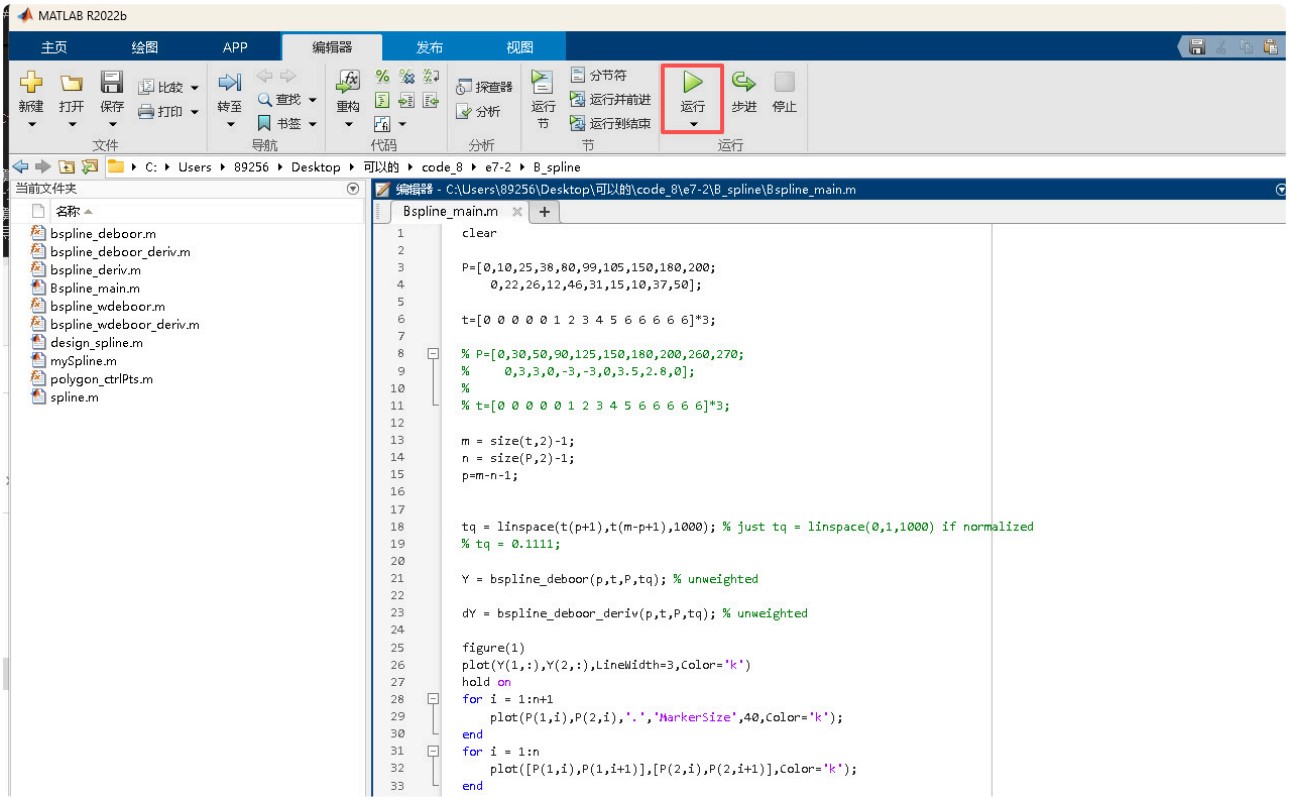
- B_spline/bspline_deboor.m：B样条De Boor迭代算法函数
- B_spline/bspline_deboor_deriv.m：B样条导数计算函数
- B_spline/bspline_wdeboor.m：加权B样条De Boor算法
- B_spline/bspline_wdeboor_deriv.m：加权B样条导数计算
- B_spline/bspline_deriv.m：B样条求导函数
- B_spline/Bspline_main.m：B样条曲线生成主程序
- B_spline/design_spline.m：样条设计辅助函数
- B_spline/mySpline.m：自定义样条函数
- B_spline/polygon_ctrlPts.m：多边形控制点生成
- B_spline/spline.m：验证控制点和节点向量对轨迹影响的程序
- bezir/bezir_n.m：贝塞尔曲线生成函数

- bezir/generate_poly.m：控制点设计和轨迹生成脚本

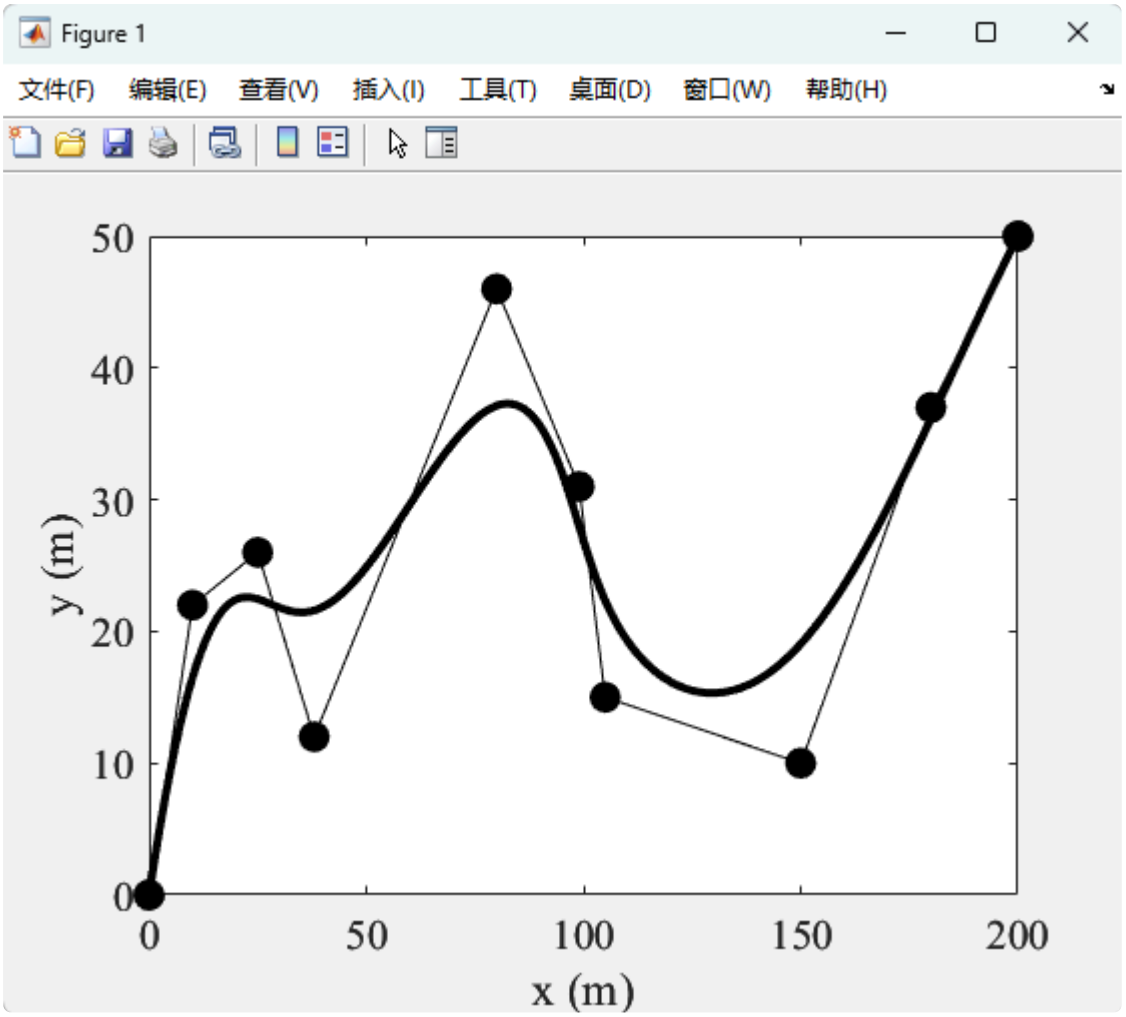
4. 实验内容或步骤

4.1 步骤1：贝塞尔曲线实验

进入e7-2文件夹，运行 "B_spline" 文件夹内 "Bspline_main.m"文件，可以完成B-样条轨迹的生成与绘制，如图所示；



可以发现绘制所得曲线为 clamped B-样条，验证参数 m = 14，n = 9，p = 4，且节点重复度所示为 p+1 = 5，满足 clamped B-样条条件。



4.2 步骤2：B样条实验

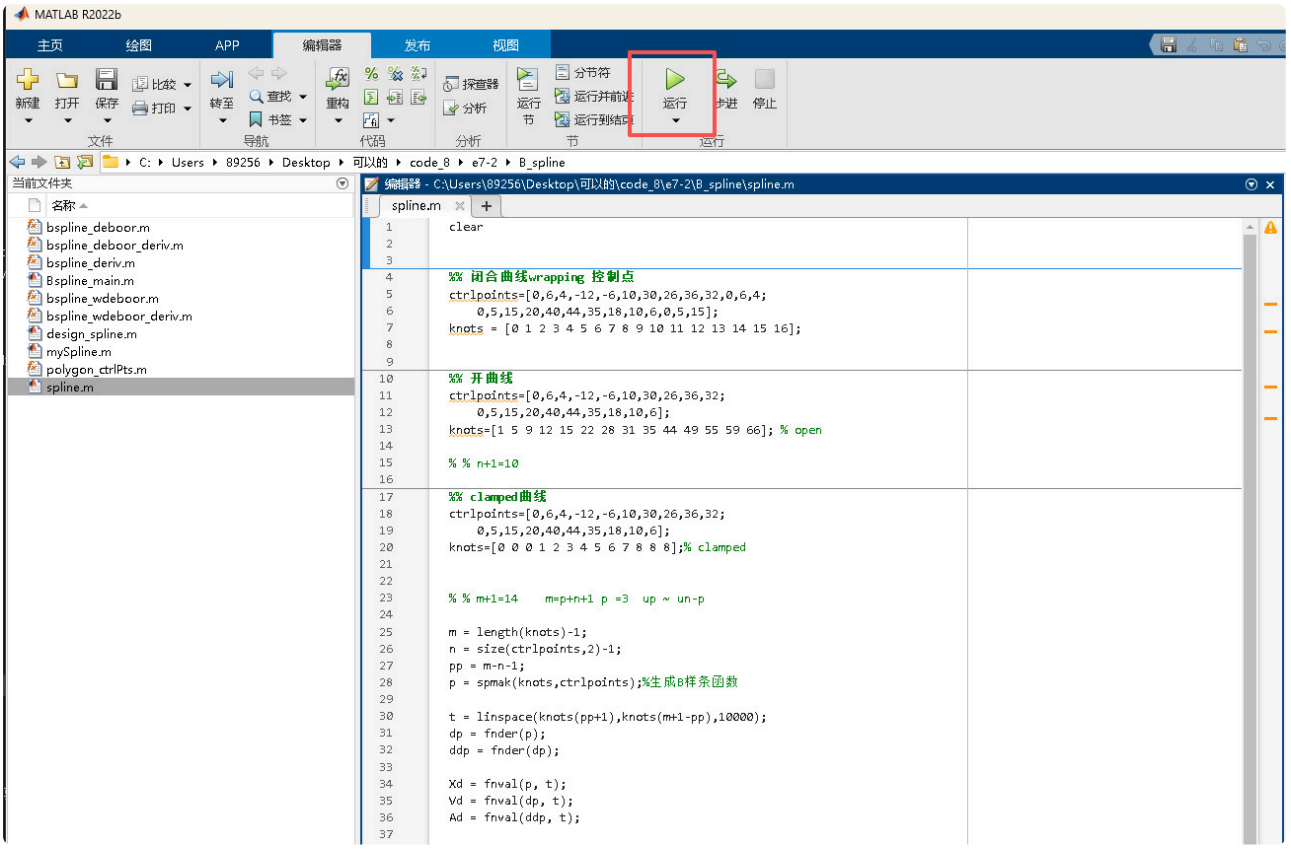
更改控制点和节点，分析不同控制点向量和节点向量对 B-样条轨迹的影响。使用MATLAB/Curve Fitting Toolbox 中的 "spmak"函数可以快速根据控制点向量和节点向量生成B-样条轨迹。这里 "spmak"函数的输入为控制点向量和节点向量，输出为 B-样条结构体。之后通过函数"fnder"实现求导功能、通过函数 "fnval"可以根据输入的时间节点得到样条轨迹值。示例代码见代码段，"knots"表示节点向量，"ctrlpoints" 表示控制点向量，"pp" 为 B-样条次数，"p"、"dp" 和"ddp" 分别为生成的B-样条轨迹及其导数轨迹和二阶导轨迹，见文件 "spline.m"。

```
m = length(knots)-1;
n = size(ctrlpoints,2)-1;
pp = m-n-1;
p = spmak(knots,ctrlpoints);%生成B样条函数
t = linspace(knots(pp+1),knots(m+1-pp),10000);
dp = fnder(p);
ddp = fnder(dp);
Xd = fnval(p, t);
xt = Xd(1,:);
yt = Xd(2,:);
```

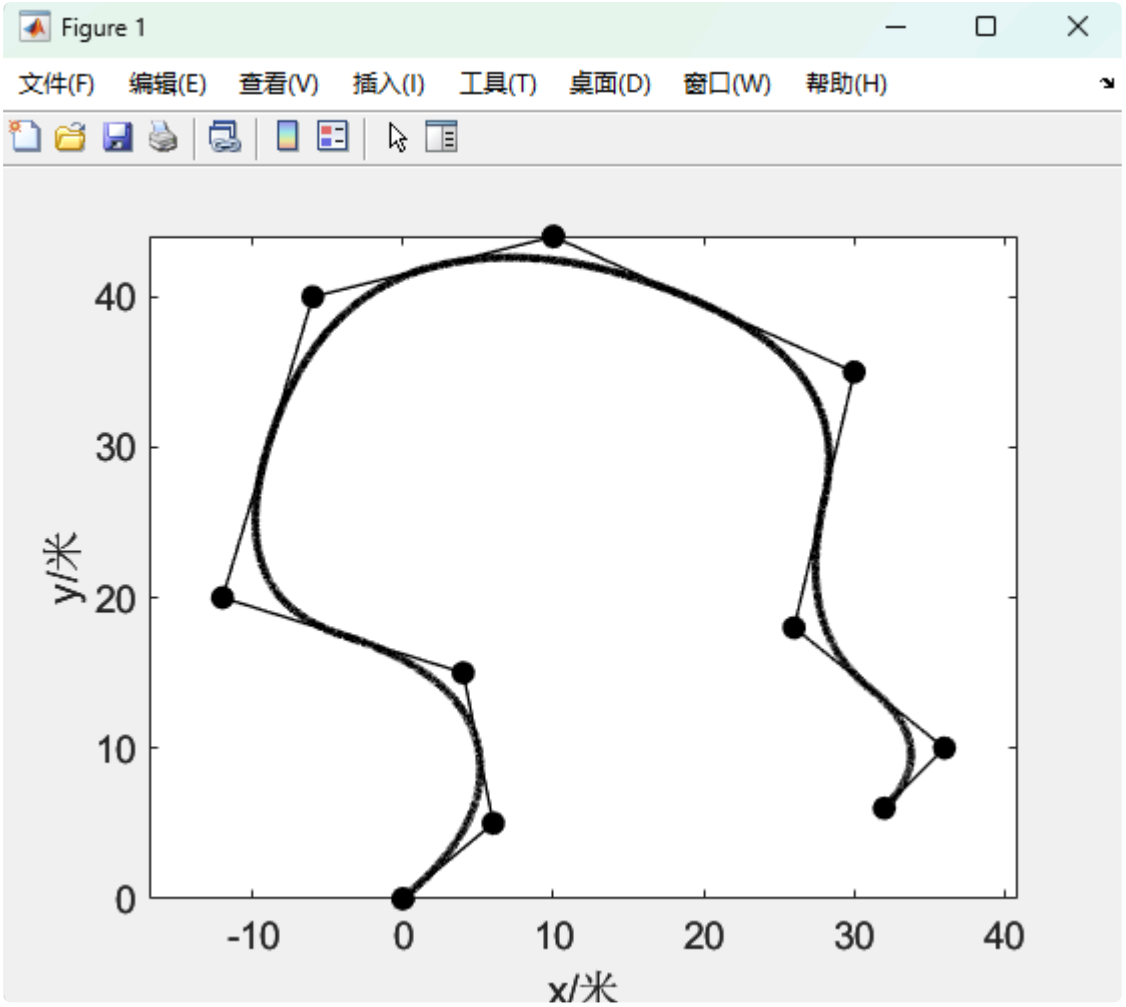
如代码段所示，为探究 clamped B-样条的生成条件，更改控制点向量和节点向量，绘制如图所示曲线的 clamped B-样条。

```
ctrlpoints=[0,6,4,-12,-6,10,30,26,36,32,0,6,4;
0,5,15,20,40,44,35,18,10,6,0,5,15];
knots = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16];
```

进入e7-2\B_spline文件夹，找到spline.m文件，运行该文件；



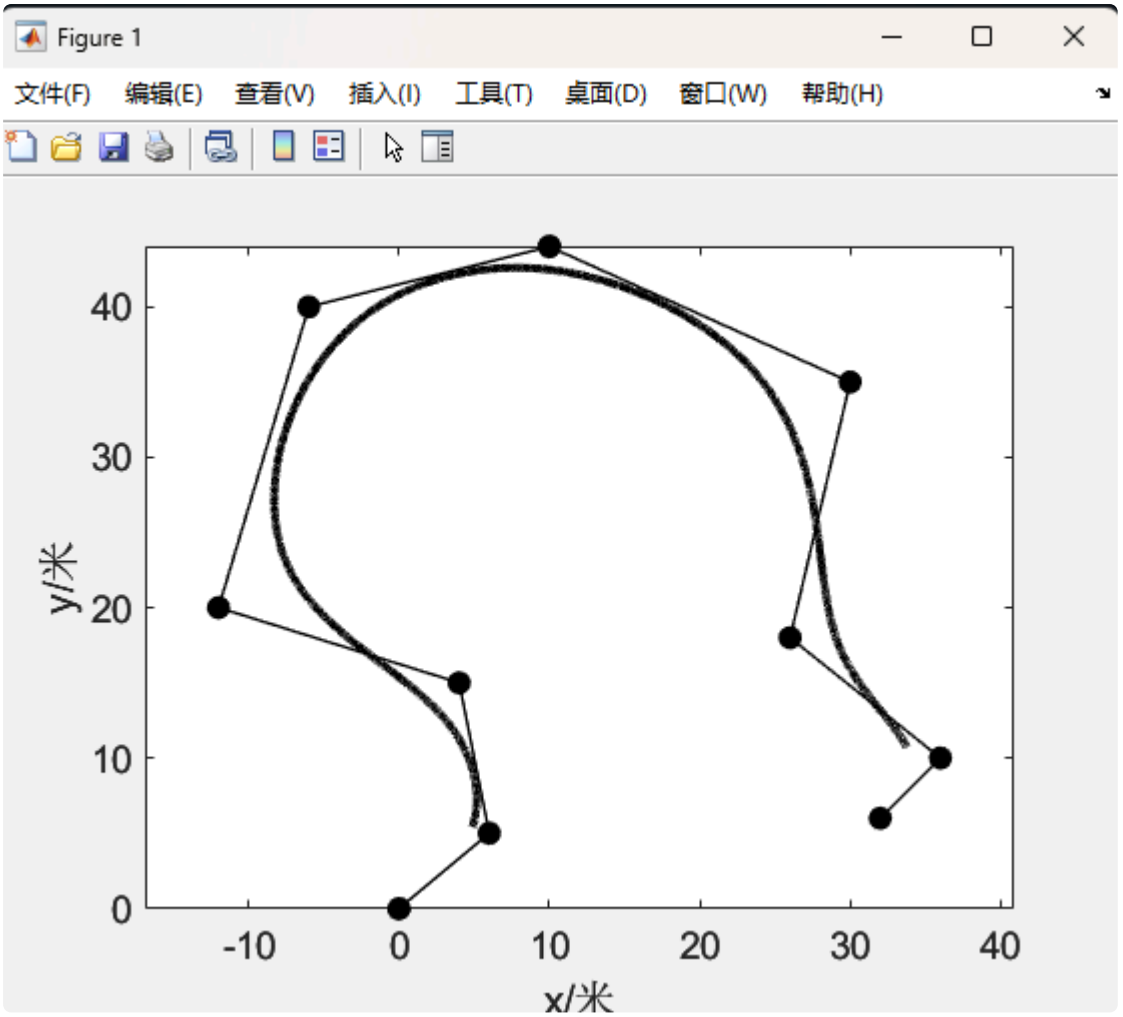
可以绘制 clamped B-样条曲线，如图所示。



上述示例控制点向量和节点向量中，m = 12，n = 9，p = m−n−1 =2。可以看出，节点重复度为 p+1 = 3，满足 clamped 条件。因此，clamped B-样条曲线经过第一个和最后一个控制点。

如代码段所示，为探究 open B-样条的生成条件，更改控制点向量和节点向量，绘制如图所示曲线的 open B-样条。

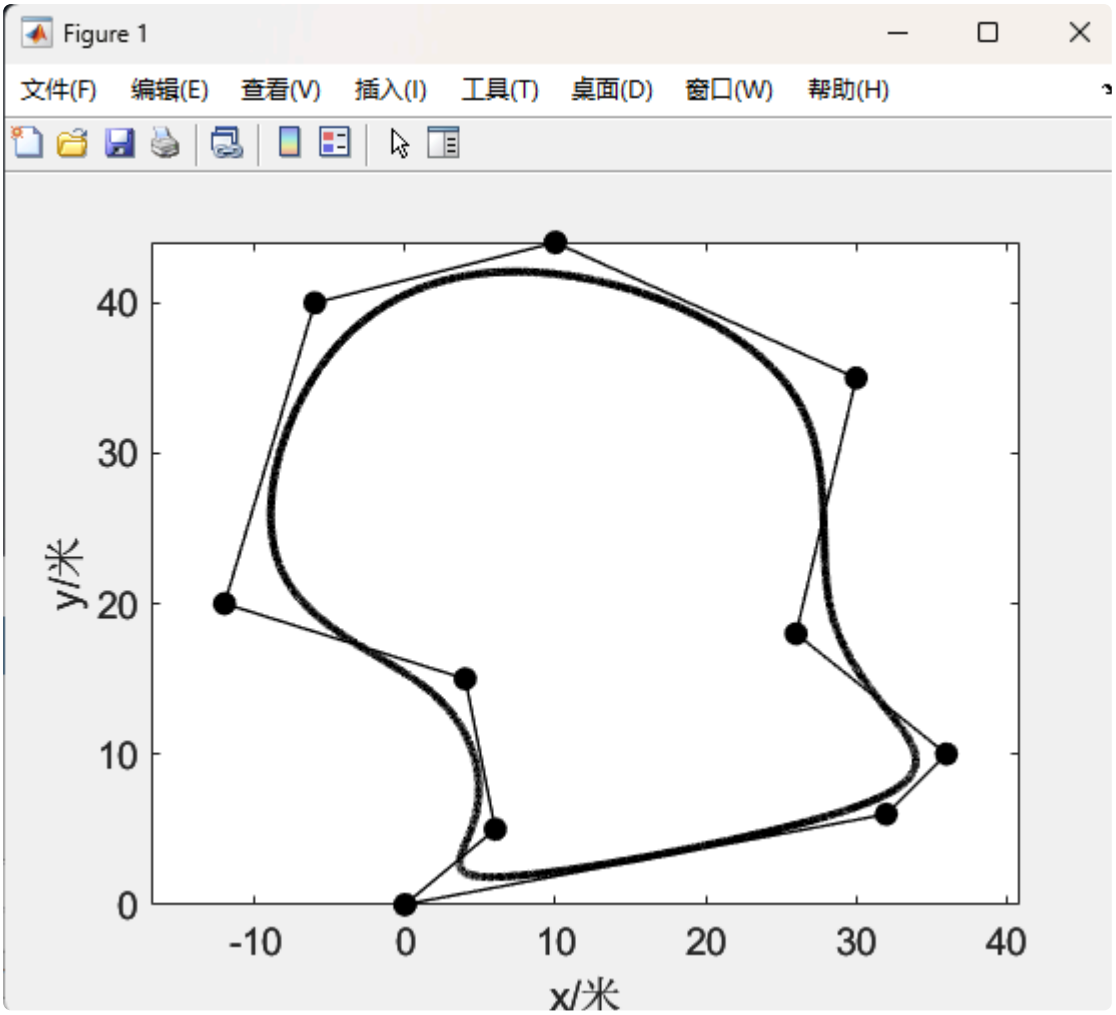
```
ctrlpoints=[0,6,4,-12,-6,10,30,26,36,32;
0,5,15,20,40,44,35,18,10,6];
knots=[1 5 9 12 15 22 28 31 35 44 49 55 59 66]; % open
```



上述示例控制点向量和节点向量中，m = 14，n = 9，p = m−n−1 = 4，与 open B-样条的生成条件一致，因此可以得到 open B-样条。

如代码段所示，为探究 closed B-样条的生成条件，更改控制点向量和节点向量，绘制如图所示曲线的 closed B-样条。有许多方法来产生闭曲线。简单的一种就是 "wrapping" 控制点。"wrapping"控制点的具体操作为：首先设计一个均匀的含 m+1 个节点序列，然后 "wrapping" 前 p个和最后 p 个控制点，如下代码中一样；

```
ctrlpoints=[0,6,4,-12,-6,10,30,26,36,32,0,6,4;
0,5,15,20,40,44,35,18,10,6,0,5,15];
knots=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16];% clamped
```



通过上述过程可以看出，给定控制点向量和控制点向量长度 n+1 后，当确定B-样条轨迹的次数 p，即可得到节点向量的长度 m+1 = n+ p+2。对于长度为 m+1的节点向量，如果开始和结束节点的重复度为p+1，则可以设计得到 clamped B-样条。将开始 p 个控制点同时作为最后 p 个控制点，即可得到 closed B-样条。

4.3 .....

5. 关键知识点

关键知识点1：贝塞尔曲线轨迹生成函数实现

在bezir_n.m中实现贝塞尔曲线生成函数。函数的输入为贝塞尔曲线的控制点向量"points"，函数的输出为贝塞尔曲线轨迹"x"、"y"，与时间参数"t"。

```
function [x, y, t] = bezir_n(points)
    syms t;
    % check points size
    n = length(points) - 1;

    % calculate k
    k = [1 1];
    for i = 1 : (n - 1)
        a = zeros(1, i+2);
        a = conv(k, [1 1]);
        clear k;
        k = a;
    end

    % 计算t
    for i = 1 : (n + 1)
        if i == 1
            j = i - 1;
            ft = (1 - t).^(n - j).*(t.^j);
            x = points(1, i) * k(i) * ft;
            y = points(2, i) * k(i) * ft;
        else
            j = i - 1;
            ft = (1 - t).^(n - j).*(t.^j);
            x = x + points(1, i) * k(i) * ft;
            y = y + points(2, i) * k(i) * ft;
        end
    end
end
```

设计得到贝塞尔曲线的控制点后，调用 "bezir_n.m" 即可生成对应的贝塞尔轨迹。在程序 "e7-2" 内"bezir" 的generate_poly.m 文件中，设计控制点向量，并作为函数 "bezir_n.m " 的输入，过程如代码段所示

```
points = [0 30 50 90 125 150 180 200 260 270;
        0 3 3 0 -3 -3 0 3.5 2.8 0];
[x, y, t] = bezir_n(points);
```

使用 "diff" 函数即可实现贝塞尔函数的求导过程。为了便于计算和时间戳索引，使用 "matlabFunction"函数转化生成的 "sym" 形式函数为形如@(t)t^2的匿名函数（括号内部为自变量，后面为相应的表达式），此后输入时间戳 t 即可得到相应输出，过程如代码段所示。

```
dy = diff(y,t);
dx = diff(x,t);
fx = matlabFunction(x);
fy = matlabFunction(y);
```

关键知识点2：B样条De Boor迭代算法实现

在 "e7-2" 的 "B_spline" 文件夹内bspline_deboor.m 文件中，编写 B-样条De Boor迭代算法函数。函数的输入为所设计B样条的次数"n"，输入节点向量"t"，控制点向量"P"，采样时间序列"U"。函数的输出为对应采样时间序列"U"的轨迹位置序列"C"。

```
function [C,U] = bspline_deboor(n,t,P,U)
d = n;
m = size(P,1);  % dimension of control points
t = t(:).';     % knot sequence
U = U(:);
S = sum(bsxfun(@eq, U, t), 2);  % multiplicity of u in t (0 <= s <= d+1)
I = bspline_deboor_interval(U,t);
Pk = zeros(m,d+1,d+1);
a = zeros(d+1,d+1);
C = zeros(size(P,1), numel(U));
for j = 1 : numel(U)
    u = U(j);
    s = S(j);
    ix = I(j);
    Pk(:) = 0;
    a(:) = 0;
    % identify d+1 relevant control points
    Pk(:, (ix-d):(ix-s), 1) = P(:, (ix-d):(ix-s));
    h = d - s;

    if h > 0
        % de Boor recursion formula
        for r = 1 : h
            q = ix-1;
            for i = (q-d+r) : (q-s)
                a(i+1,r+1) = (u-t(i+1)) / (t(i+d-r+1+1)-t(i+1));
                Pk(:,i+1,r+1) = (1-a(i+1,r+1)) * Pk(:,i,r) + a(i+1,r+1) *
Pk(:,i+1,r);
            end
end
C(:,j) = Pk(:,ix-s,d-s+1);  % extract value from triangular computation scheme
elseif ix == numel(t)  % last control point is a special case
C(:,j) = P(:,end);
else
C(:,j) = P(:,ix-d);
end
end
```

设计控制点和节点向量后，调用"bspline_deboor.m"可以实现 B-样条轨迹的生成。如代码段所示，生成 B-样条曲线，并获得轨迹导数。其中示例代码的第 2-4 行为定义控制点向量与节点向量；第6-8行为计算多项式次数；第10行对应计算采样时间；第11、12行对应调用函数计算 B-样条轨迹及导数。

```
clear
P=[0,10,25,38,80,99,105,150,180,200;
0,22,26,12,46,31,15,10,37,50];
t=[0 0 0 0 0 1 2 3 4 5 6 6 6 6 6]*3;
m = size(t,2)-1;
n = size(P,2)-1;
p=m-n-1;
tq = linspace(t(p+1),t(m-p+1),1000); % just tq = linspace(0,1,1000) if normalized
Y = bspline_deboor(p,t,P,tq); % unweighted
dY = bspline_deboor_deriv(p,t,P,tq); % unweighted
```

更多详细实验原理可见：全权,高文瀚,刘润潇,陈鑫泉,戴训华,吕书礼,徐琳,李悦.微小型固定翼无人机飞行控制设计与实践. 北京, 2025

6.参考资料

1. 全权,高文瀚,刘润潇,陈鑫泉,戴训华,吕书礼,徐琳,李悦.微小型固定翼无人机飞行控制设计与实践. 北京, 2025.
2. RflySim官方文档
3. PX4飞控固件官方文档
4. 飞思实验室官网
5. Farin G. Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide[M]. 5th ed. San Diego: Academic Press, 2002.

7.常见问题

Q1：生成的曲线不平滑怎么办？

A1：检查控制点的数量和分布是否合理。增加控制点数量或调整控制点位置可以改善曲线平滑度。另外，可以尝试调整节点向量的分布，使节点间隔更均匀，或者使用更高次的样条曲线。

Q2：B样条曲线没有通过端点怎么办？

A2：这通常是因为节点向量设置不正确。对于clamped B样条，需要确保开始和结束节点的重复度为p+1（p为样条次数）。例如，如果样条次数为3，则首末节点应重复4次。

Q3：贝塞尔曲线形状不符合预期怎么办？

A3：贝塞尔曲线的形状由控制点决定，曲线始末点为第一个和最后一个控制点。调整中间控制点的位置可以改变曲线形状。另外，增加控制点数量可以提供更多形状控制自由度。如果需要更复杂的形状，建议使用B样条曲线。

---

1. https://rflysim.com/ ↩
2. 推荐配置请见： https://rflysim.com/doc/zh/HowToInstall.pdf ↩