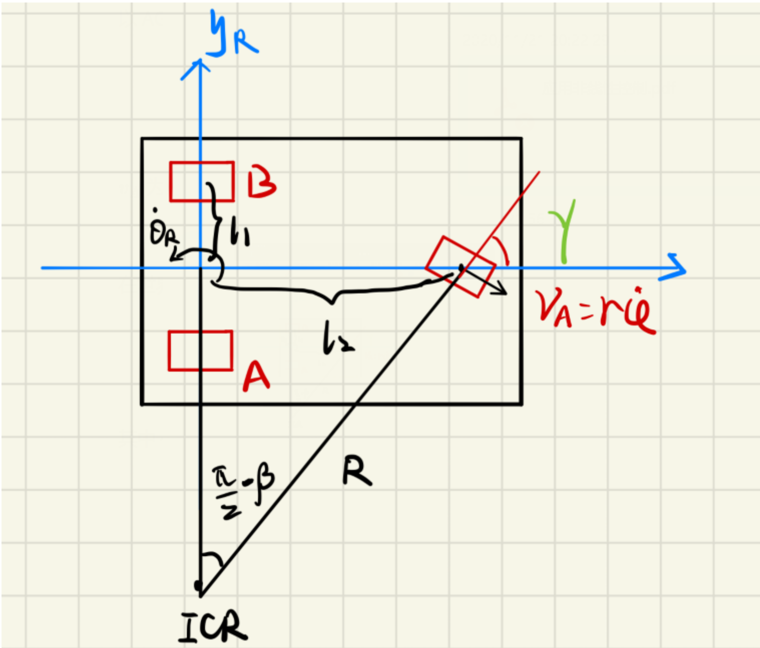
SZ170410221 朱方程

Homework2



运动学模型

以 AGV 后两轮中心作为机器人系参考点，其运动学模型为

$$\begin{cases} \dot{x}_R = v_A \sin\gamma \\ \dot{y}_R = 0 \\ \dot{\theta}_R = \omega = \dfrac{-v_A\cos\gamma}{l_2} \end{cases} \tag{1}$$

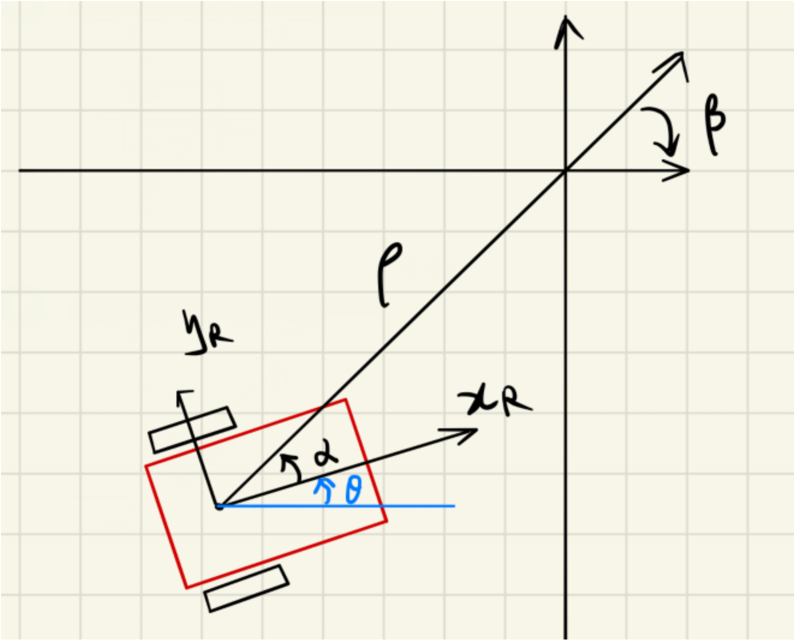
v_A 为舵轮的线速度。有

$$v_A = r\dot{\varphi} \tag{2}$$

在世界坐标系中观察，有

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r\dot{\varphi}\sin\gamma \\ \dfrac{-r\dot{\varphi}\cos\gamma}{l_2} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

其中 v 为叉车线速度，ω 为角速度。



极坐标描述

假设目标位置为世界坐标系的原点，转换到极坐标

$$
\begin{aligned}
\rho &= \sqrt{\Delta x^2 + \Delta y^2} \\
\beta &= -\arctan\frac{\Delta y}{\Delta x} \\
\alpha &= -\theta - \beta
\end{aligned} \tag{4}
$$

对ρ, β, α 求导可得

误差分析可得

$$
\begin{aligned}
e_c = p - p_r = [x - x_r, y - y_r, \theta - \theta_r] = [\tilde{x}, \tilde{y}, \tilde{\theta}] \\
\dot{e_p} = [\dot{\rho}, \dot{\alpha}, \dot{\beta}]
\end{aligned} \tag{5}
$$

其中

$$
\begin{aligned}
\dot{\rho} = \frac{\dot{\tilde{x}}\tilde{x}}{\rho} + \frac{\dot{\tilde{y}}\tilde{y}}{\rho} = v\cos\tilde{\theta}(-\cos(-\beta)) + v\sin\tilde{\theta}(-\sin(-\beta)) = -v\cos\alpha \\
\dot{\alpha} = \frac{\sin\alpha}{\rho}v - \omega \\
\dot{\beta} = -\frac{1}{1+(\frac{\tilde{y}}{\tilde{x}})^2}(\frac{\tilde{y}}{\tilde{x}})' = -\frac{\sin\alpha}{\rho}v
\end{aligned} \tag{6}
$$

结论为

$$\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}-\cos\alpha & 0\\ \frac{\sin\alpha}{\rho} & -1\\ -\frac{\sin\alpha}{\rho} & 0\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \tag{7}$$

控制器设计

$$\begin{aligned} v &= k_\rho \rho \\ \omega &= k_\alpha \alpha + k_\beta \beta \end{aligned} \tag{8}$$

运动学逆解求得输入的舵轮转速及方向角 γ 为

$$\begin{aligned} \gamma &= \arctan 2(-\frac{v}{\omega l_2}) \\ \dot{\varphi} &= \frac{v}{r\sin\gamma} \end{aligned} \tag{9}$$

控制器稳定性分析

将控制器代入状态方程，可得

$$\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}-k_\rho \rho\cos\alpha\\ k_\rho \sin\alpha - k_\alpha\alpha - k_\beta\beta\\ -k_\rho\sin\alpha\end{bmatrix} \tag{10}$$

系统在$(\rho, \alpha, \beta) = (0，0，0)$ 有唯一平衡点

围绕平衡点 $(0, 0, 0)$ 作线性化近似，即 $\cos\alpha \to 1, \sin\alpha \to 0$，当 $\alpha \to 0$.

$$\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}-k_\alpha & 0 & 0\\ 0 & -(k_\alpha - k_\rho) & k_\beta\\ 0 & -k_\rho & 0\end{bmatrix}\begin{bmatrix}\rho\\ \alpha\\ \beta\end{bmatrix} \tag{11}$$

系统矩阵的特征多项式为

$$\Delta(\lambda) = (\lambda + k_\alpha)[\lambda^2 + \lambda(k_\alpha - k_\rho) - k_\rho k_\beta] \tag{12}$$

如果参数满足

$$\begin{aligned} &\beta \geq 0 \\ &k_\beta \leq 0 \\ &k_\alpha - k_\rho \geq 0 \end{aligned} \tag{13}$$

所有特征根均具有负实部，该系统可局部稳定。

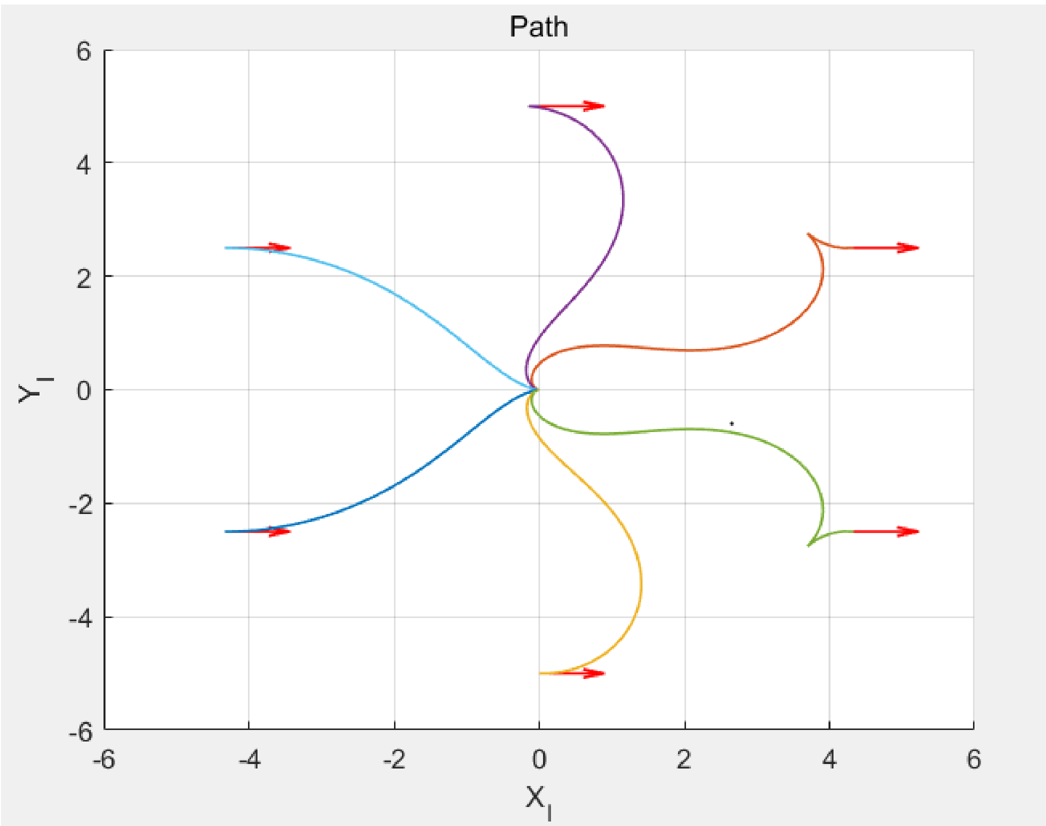
仿真代码

```
clc
clear all
for j=1:6
    r(j,1)=5;
    beta(j,1)=-pi/3*j+7/6*pi;
    theta=0;
    alpha(j,1)=-beta(j,1)-theta;
    x(j,1)=r(j,1)*cos(-beta(j,1)-pi);
    y(j,1)=r(j,1)*sin(-beta(j,1)-pi);
end
k_r=3;
k_a=7;
k_b=-1.5;
T=0.01;
for j=1:6
    for i=1:500
         if alpha(j,i)>pi/2 || alpha(j,i)<=-pi/2
             k_r=-3;
         else
             k_r=3;
         end
        r(j,i+1)=r(j,i)+T*(-k_r*r(j,i)*cos(alpha(j,i)));
        alpha(j,i+1)=alpha(j,i)+T*(k_r*sin(alpha(j,i))-k_a*alpha(j,i)-k_b*beta(j,i));
        beta(j,i+1)=beta(j,i)+T*(-k_r*sin(alpha(j,i)));
        w(j,i+1)=k_a*alpha(j,i+1)+k_b*beta(j,i+1);
        x(j,i+1)=r(j,i+1)*cos(-beta(j,i+1)-pi);
        y(j,i+1)=r(j,i+1)*sin(-beta(j,i+1)-pi);
    end
    figure(1)
    hold on
    grid on
    plot(0:0.01:5,alpha(j,:),'LineWidth',1);
    title('Alpha');
    xlabel('Time/s') ;
    ylabel('Alpha/rad') ;
    hold off

    figure(2)
    hold on
    grid on
    plot(0:0.01:5,beta(j,:),'LineWidth',1);
    title('Beta');
    xlabel('Time/s') ;
    ylabel('Beta/rad') ;
    hold off

    figure(3)
    hold on
```

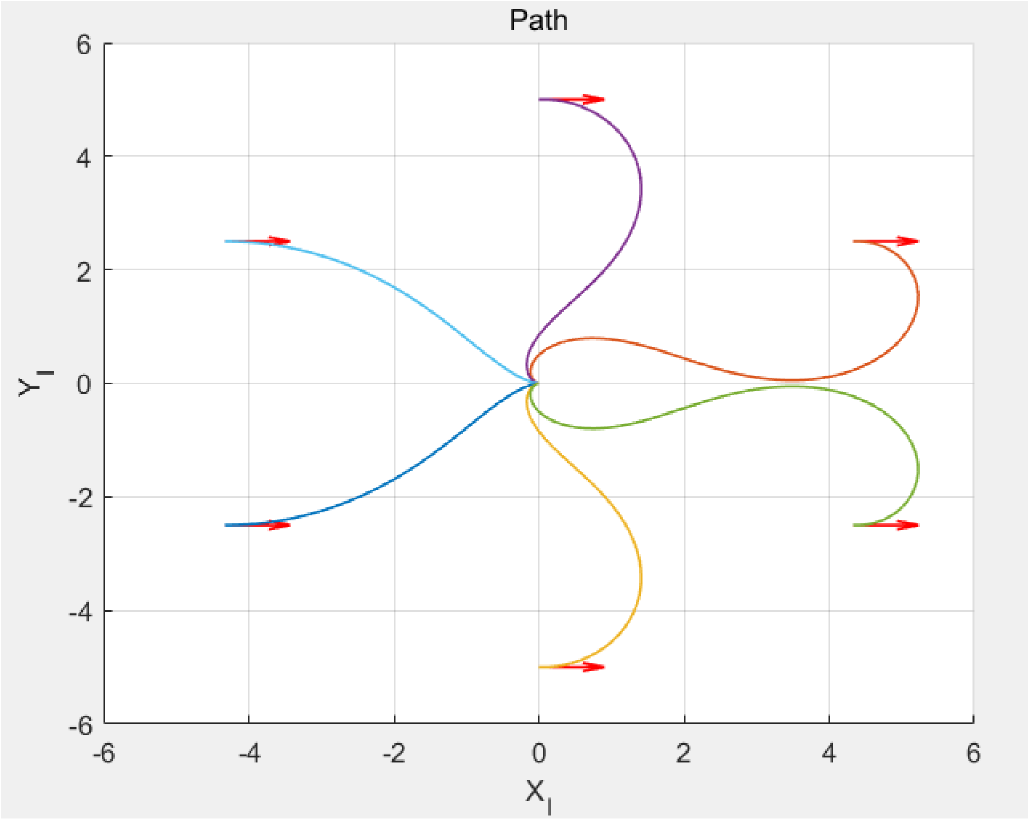
```
    grid on
    plot(0:0.01:5,r(j,:),'LineWidth',1);
    title('\rho');
    xlabel('Time/s') ;
    ylabel('\rho') ;
    hold off

    figure(4)
    hold on
    grid on
    q=quiver(x(j,1),y(j,1),1,0);
    q.Color = 'red';
    q.LineWidth=1;
    q.MaxHeadSize=1;
    plot(x(j,:),y(j,:),'LineWidth',1);
    hold off
    title('Path');
    xlabel('X_I') ;
    ylabel('Y_I') ;
    %legend('1', '2', '3', '4', '5', '6');
end

```

仿真结果

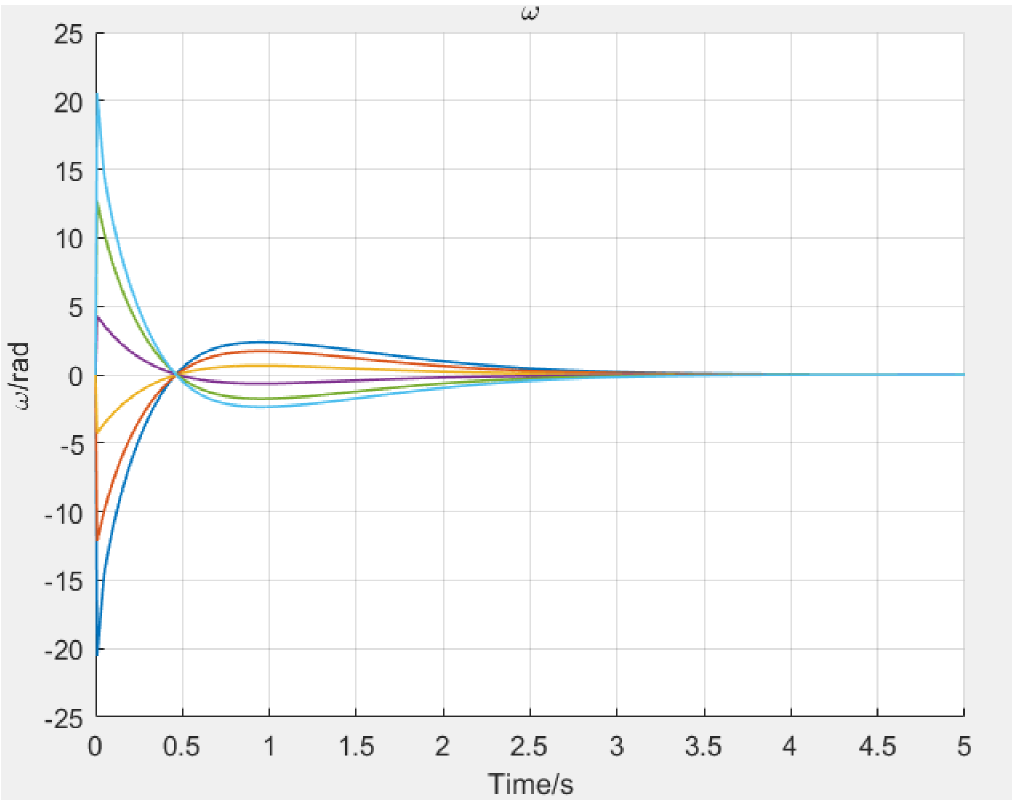
允许“倒车”：



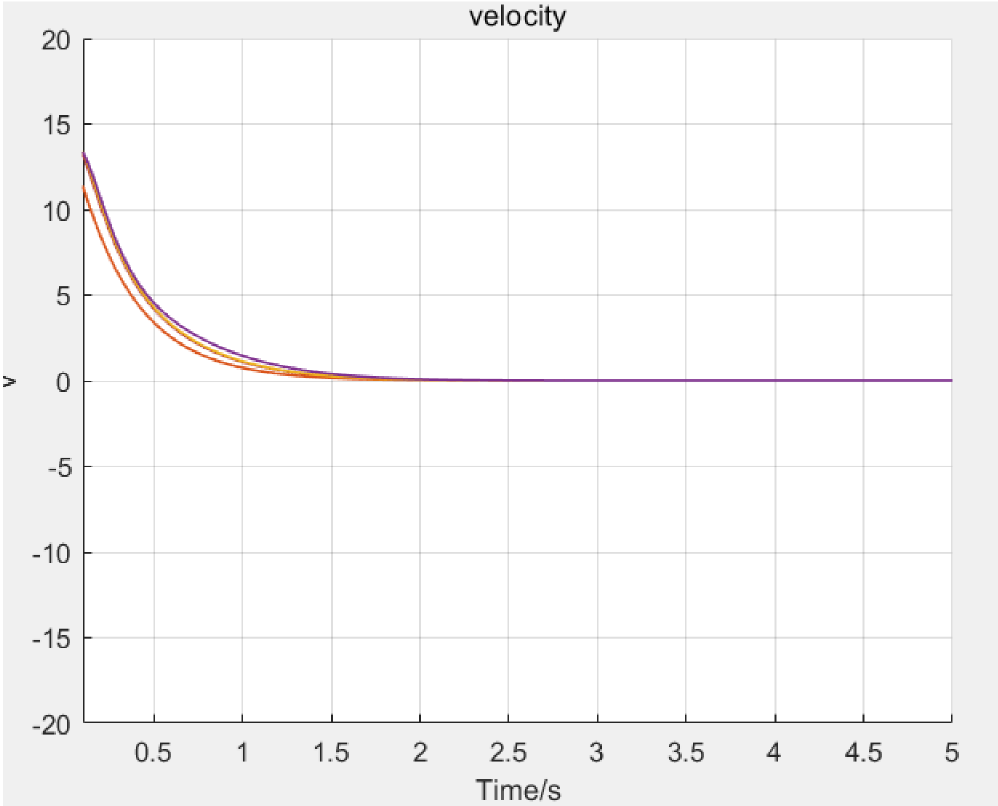
不允许“倒车”：

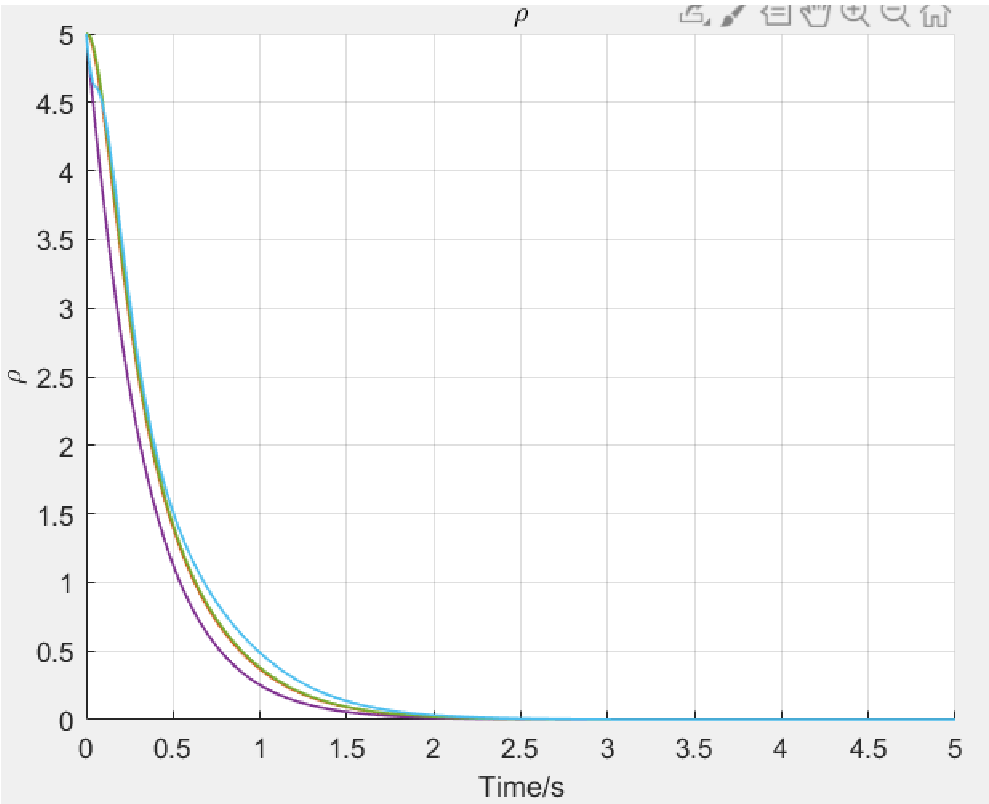


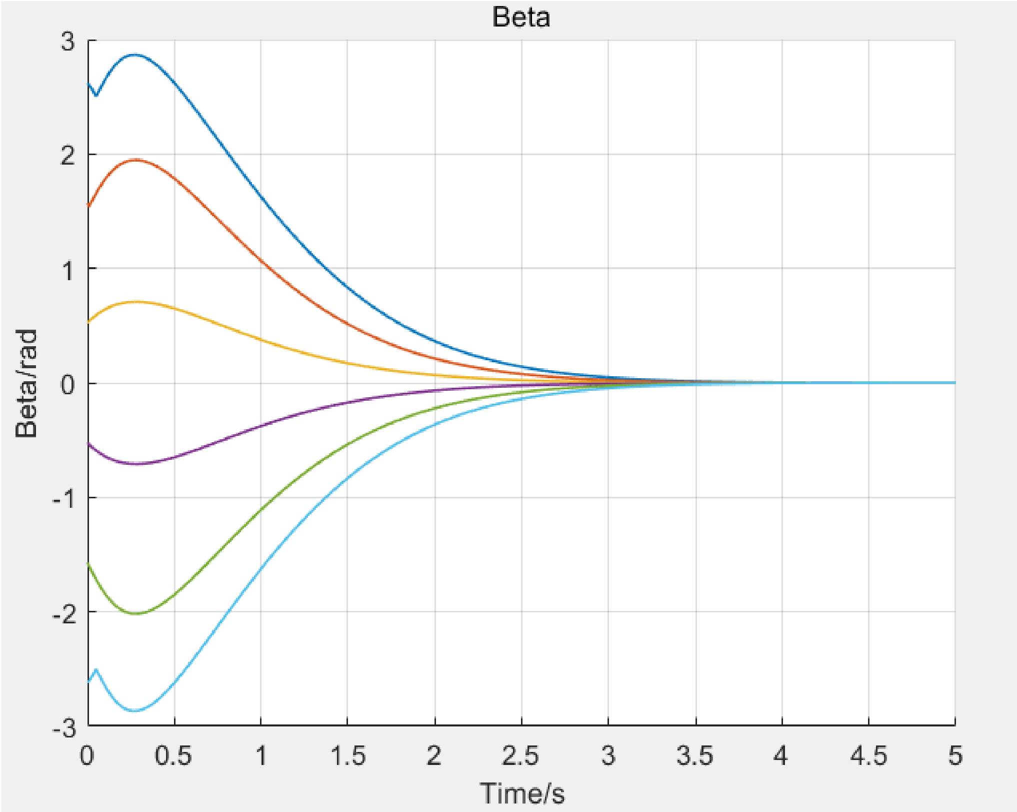
Path
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6
Y_I
X_I

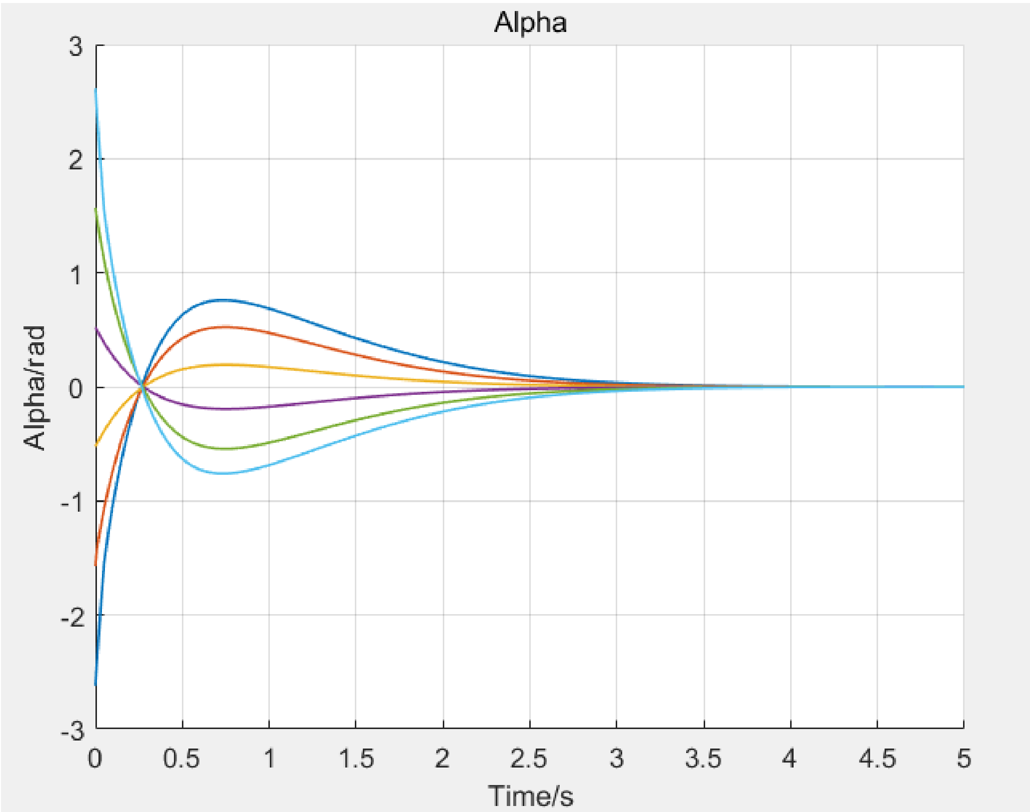


ω
25
20
15
10
5
0
-5
-10
-15
-20
-25
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
ω/rad
Time/s


velocity
v
Time/s
20
15
10
5
0
-5
-10
-15
-20
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


ρ
ρ
Time/s
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5



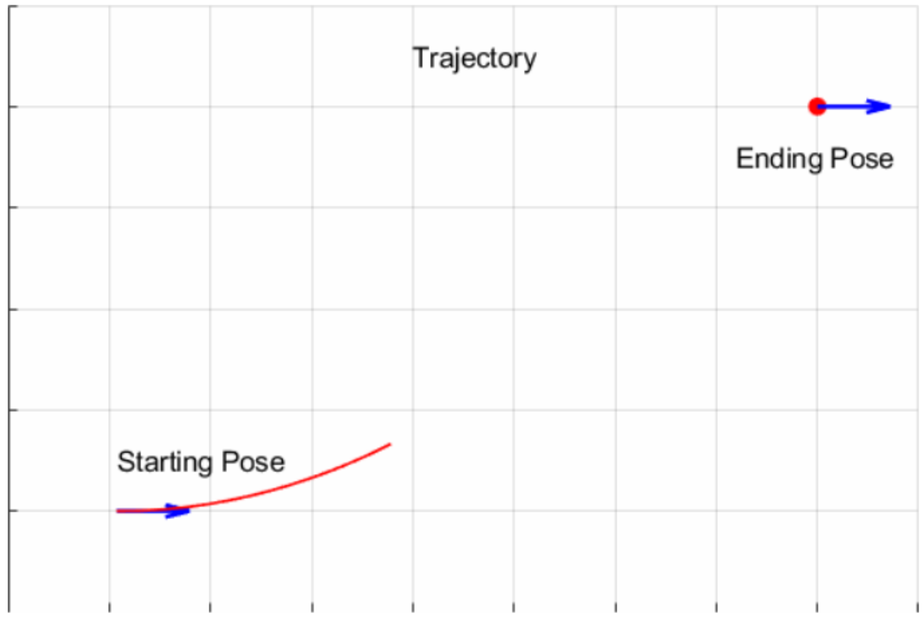


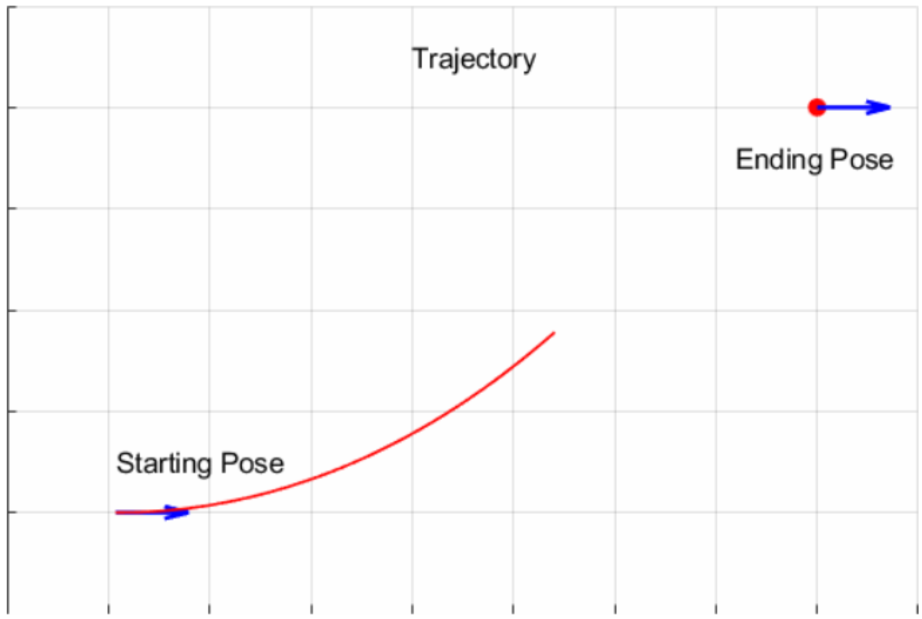
绘制gif代码

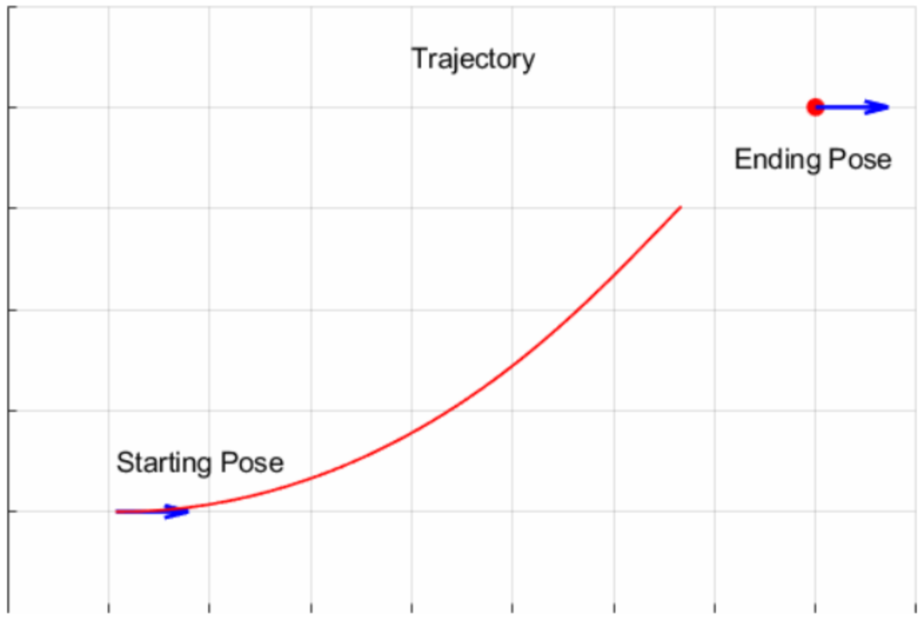
```
clc
clear all
for j=1:6
    r(j,1)=8;
    beta(j,1)=-pi/3*j+7/6*pi;
    theta=0;
```

```
    alpha(j,1)=-beta(j,1)-theta;
    x(j,1)=r(j,1)*cos(-beta(j,1)-pi);
    y(j,1)=r(j,1)*sin(-beta(j,1)-pi);
end
k_r=3;
k_a=7;
k_b=-1.5;
T=0.01;
for j=4:4
    for i=1:500
        r(j,i+1)=r(j,i)+T*(-k_r*r(j,i)*cos(alpha(j,i)));
        alpha(j,i+1)=alpha(j,i)+T*(k_r*sin(alpha(j,i))-k_a*alpha(j,i)-k_b*beta(j,i));
        beta(j,i+1)=beta(j,i)+T*(-k_r*sin(alpha(j,i)));
        w(j,i+1)=k_a*alpha(j,i+1)+k_b*beta(j,i+1);
        x(j,i+1)=r(j,i+1)*cos(-beta(j,i+1)-pi);
        y(j,i+1)=r(j,i+1)*sin(-beta(j,i+1)-pi);
    end
    figure(1)
    hold on
    grid on
    plot(x(j,:),y(j,:),'LineWidth',1);
    hold off
    title('Path');
    xlabel('X_I') ;
    ylabel('Y_I') ;

figure(2);
hold on
%添加文本
txt = 'Trajectory';
text(-4,0.5,txt)
txt = 'Starting Pose';
text(x(j,1),y(j,1)+0.5,txt)
txt = 'Ending Pose';
text(-0.8,-0.5,txt)
%画实心圆点
plot(0,0,'ro','MarkerFaceColor','r');
%画箭头
q=quiver(x(j,1),y(j,1),0.8,0);
q.Color = 'blue';
q.LineWidth=1.5;
q.MaxHeadSize=1.5;
q=quiver(0,0,0.8,0);
q.Color = 'blue';
q.LineWidth=1.5;
q.MaxHeadSize=1.5;
%画gif
h = plot(x(j,1),'r','LineWidth',1);
```

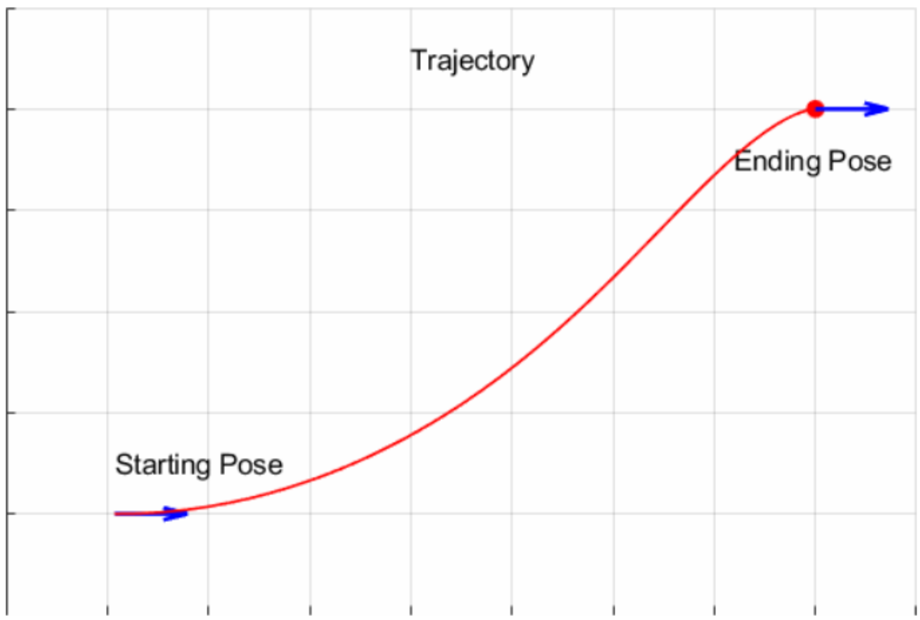
```
grid on
axis equal
axis([-8,1,-5,1]);
[A,map] = rgb2ind(frame2im(getframe),256);
imwrite(A,map,'1.gif','LoopCount',65535,'DelayTime',0.1);
for i = 1:length(x)
    h.XData(i) = x(j,i);
    h.YData(i) = y(j,i);
    [A,map] = rgb2ind(frame2im(getframe),256);
    imwrite(A,map,'1.gif','WriteMode','append','DelayTime',0.1);
    pause('on')
    pause(0.05)
end
end
```







Trajectory
Ending Pose
Starting Pose



Trajectory
Ending Pose
Starting Pose