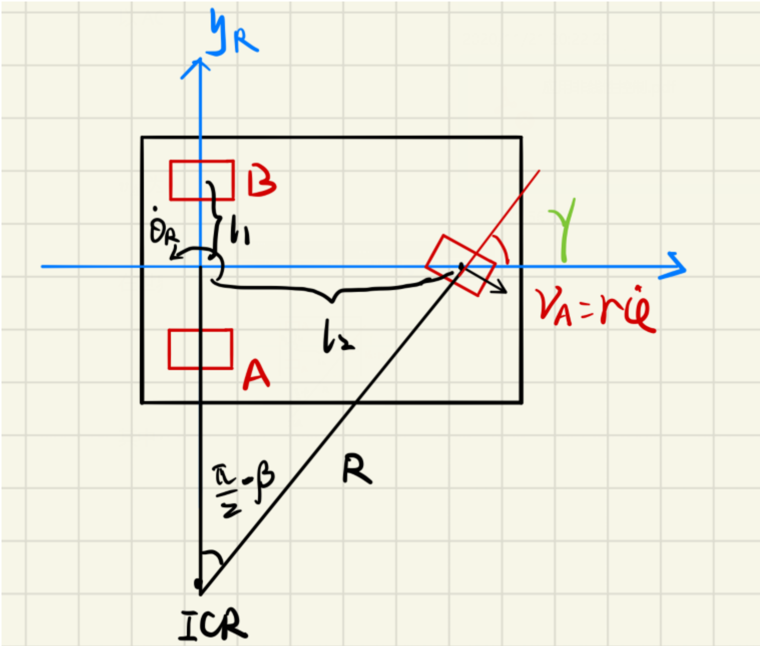
SZ170410221 朱方程

Homework3



运动学模型

以叉车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

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

位姿描述

假设平面移动机器人的位姿可以在世界坐标系下用如下方式表示

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{4}$$

利用二阶龙格库塔法来求取里程计模型，叉车的位姿可以用里程计积分（离散为差分）来估计：

$$\begin{gathered}
\Delta x = \Delta s \sin(\gamma + \frac{\Delta\gamma}{2}) \cos(\theta + \frac{\Delta\theta}{2}) \\
\Delta y = \Delta s \sin(\gamma + \frac{\Delta\gamma}{2}) \sin(\theta + \frac{\Delta\theta}{2}) \\
\Delta\theta = -\Delta s \frac{\cos(\gamma + \frac{\Delta\gamma}{2})}{l_2} \\
\Delta s = r\Delta\varphi
\end{gathered} \tag{5}$$

更新后的位置 p' 为

$$p' = p + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = f(x, y, \theta, \Delta s, \Delta\gamma) \tag{6}$$

协方差估计和更新

对于叉车，我们认为转向带来的不确定度为常数，转动的距离不确定度和距离大小成正比，即得到

$$\Sigma_\Delta = \begin{bmatrix} k_\gamma & 0 \\ 0 & k_s|\Delta s| \end{bmatrix} \tag{7}$$

其中 k_γ, k_s 为常数。

利用误差传播模型可以得到

$$\Sigma'_p = \nabla_p \Sigma_p \nabla_p^T + \nabla_\delta \Sigma_\Delta \nabla_\delta^T \tag{8}$$

其中

$$\begin{aligned}
\nabla_p = \nabla_p^T &= \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & -\Delta s \sin(\gamma + \frac{\Delta\gamma}{2}) \sin(\theta - \Delta s \frac{\cos(\gamma + \frac{\Delta\gamma}{2})}{2l_2}) \\ 0 & 1 & \Delta s \sin(\gamma + \frac{\Delta\gamma}{2}) \sin(\theta - \Delta s \frac{\cos(\gamma + \frac{\Delta\gamma}{2})}{2l_2}) \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{9}$$

$$
\nabla_\delta = \nabla_\delta^T = \begin{bmatrix} \frac{\partial f}{\partial \Delta s} & \frac{\partial f}{\partial \Delta \gamma} \end{bmatrix} \tag{10}
$$

$$
= \begin{bmatrix}
\sin\Gamma\cos(\theta - \frac{\Delta s\cos\Gamma}{2l_2}) + \frac{\Delta s\sin\Gamma\sin(\theta - \frac{\Delta s\cos\Gamma}{2l_2})}{2l_2}\cos\Gamma & \Delta s\cos(\theta - \frac{\Delta s\cos\Gamma}{2l_2})\cos\Gamma - \frac{\Delta s^2\sin^2\Gamma\sin(\theta - \frac{\Delta s\cos\Gamma}{2l_2})}{2l_2} \\
\sin\Gamma\sin(\theta - \frac{\Delta s\cos\Gamma}{2l_2}) - \frac{\Delta s\sin\Gamma\cos(\theta - \frac{\Delta s\cos\Gamma}{2l_2})}{2l_2}\cos\Gamma & \Delta s\cos(\theta - \frac{\Delta s\cos\Gamma}{2l_2})\cos\Gamma + \frac{\Delta s^2\sin^2\Gamma\cos(\theta - \frac{\Delta s\cos\Gamma}{2l_2})}{2l_2} \\
-\frac{\cos\Gamma}{l_2} & \frac{\Delta s\sin\Gamma}{l_2}
\end{bmatrix}
$$

其中 $\Gamma = \gamma + \frac{\Delta\gamma}{2}$

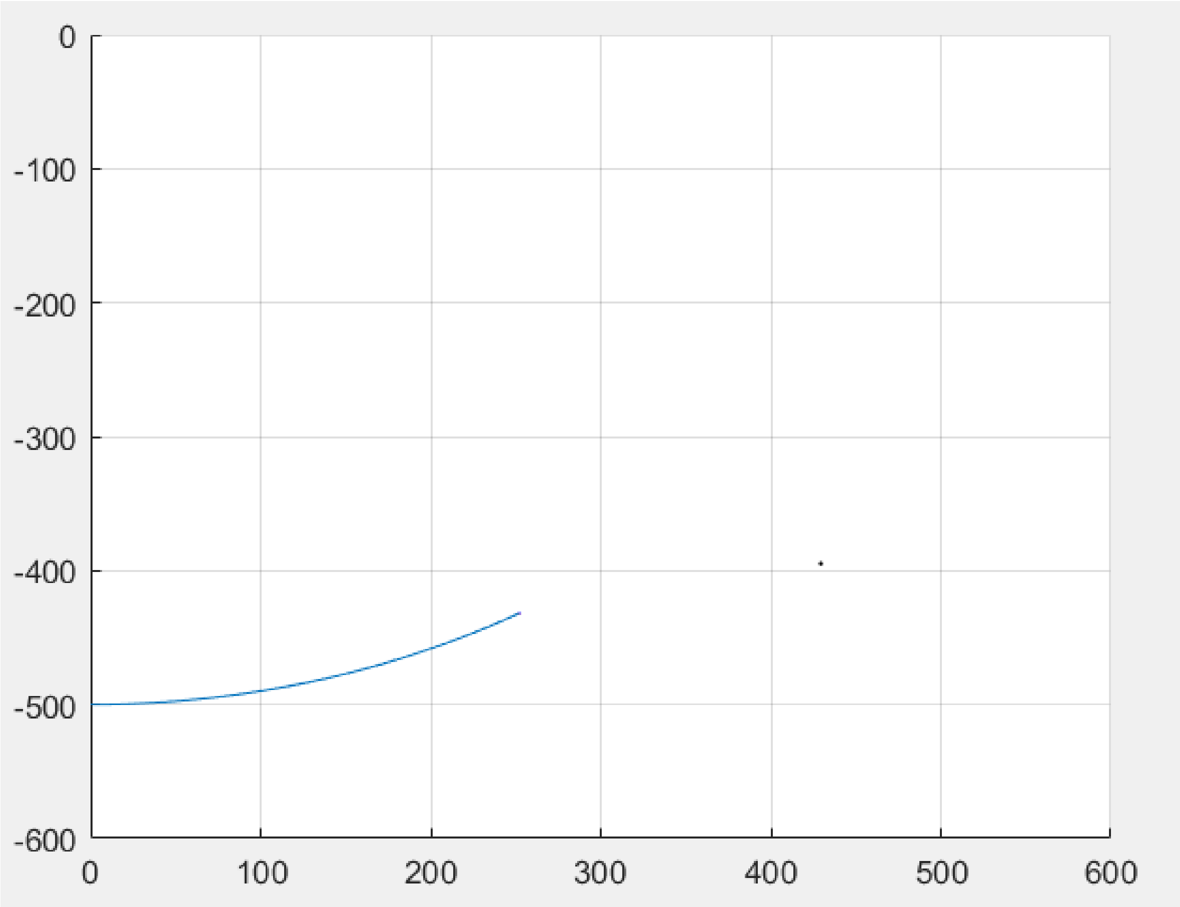
MATLAB仿真

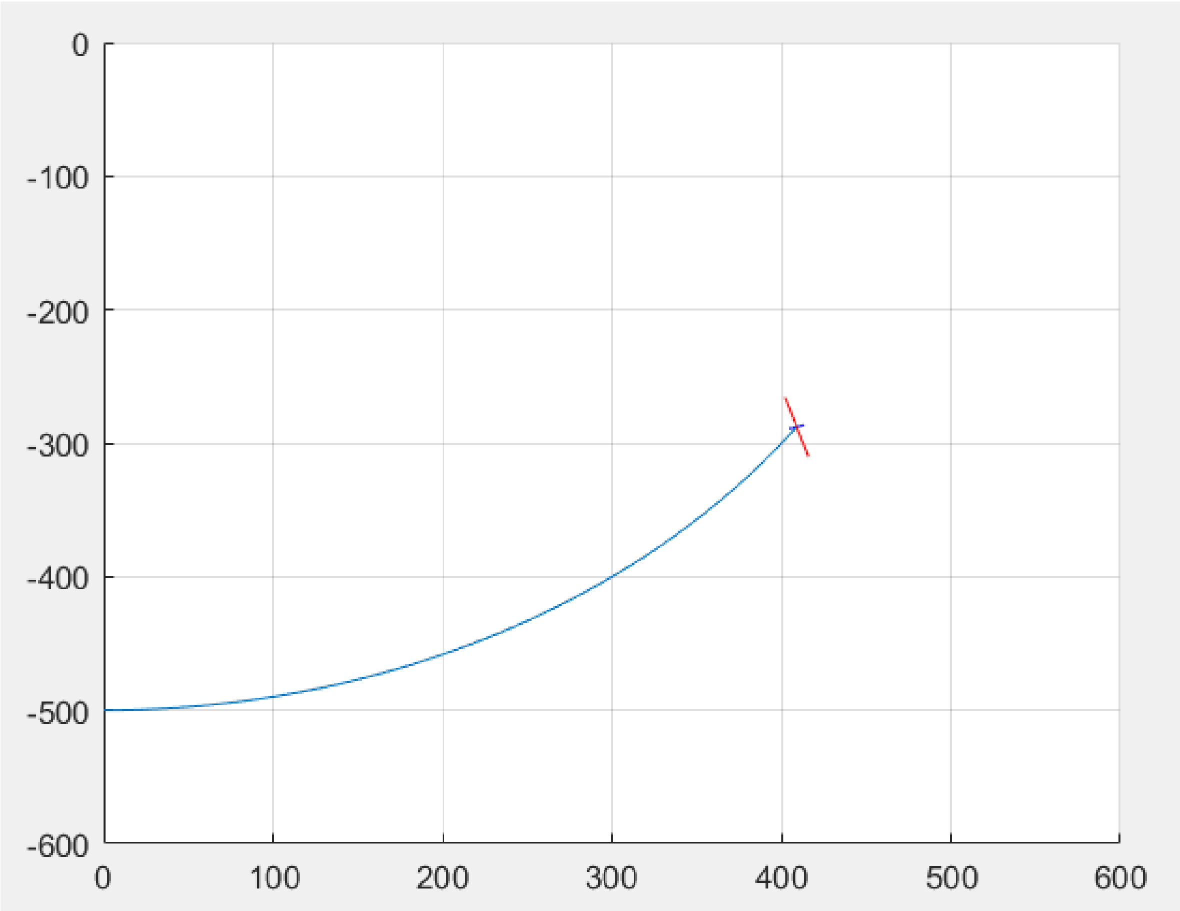
代码如下

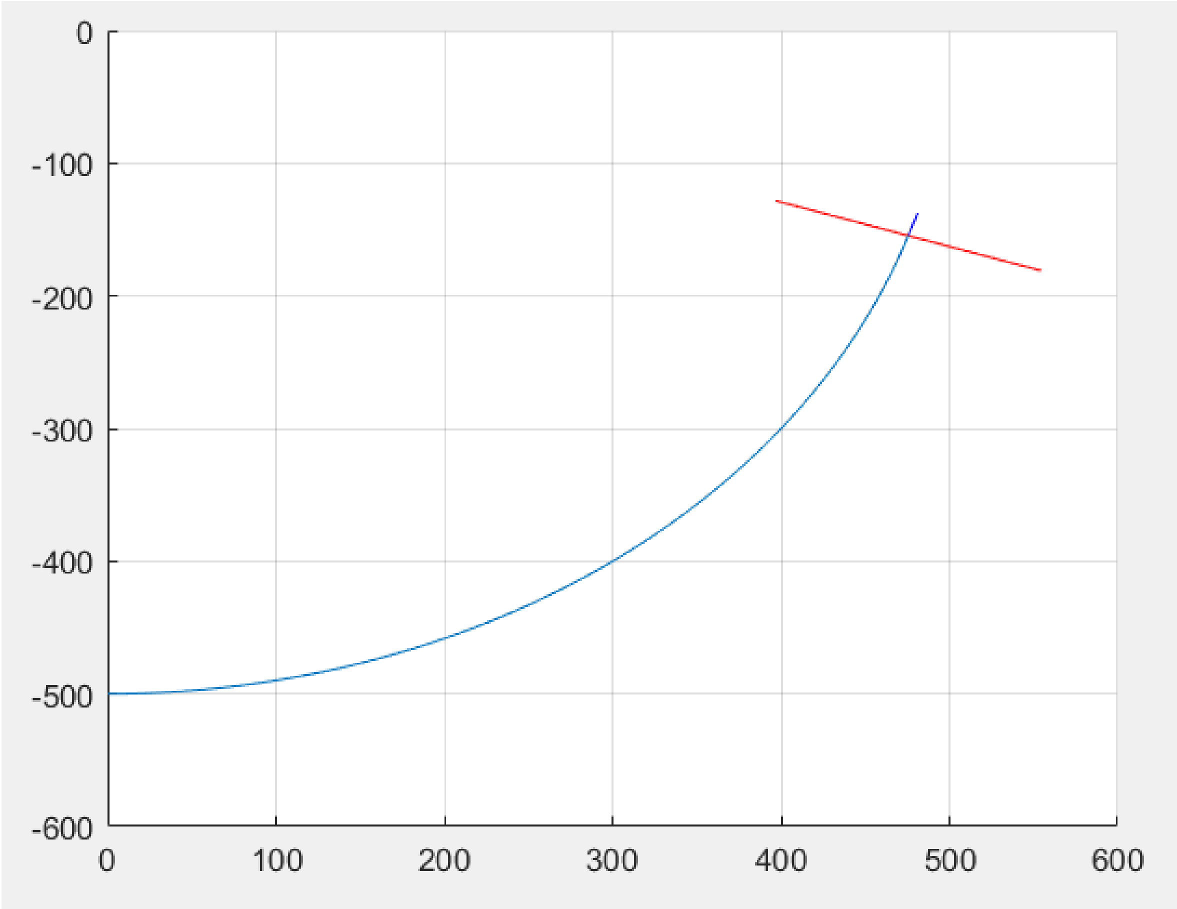
```
close all
clear all
%初始化位置协方差
CovP = zeros(3,3);
t = 5;
%位置增量和角度增量
ds=0;dg=0;
cur = 0;
x=[0;0;0];
plotx=[];
ax=subplot(1,1,1);
%循环
while cur<t
    cla(ax);
   dx = [cur;0;0]
    x = [500*sin(cur/t*2*pi/5);-500*cos(cur/t*2*pi/5);0];

    b = pi/2*cur/t;
    ds = norm(dx);
    %更新协方差
    CovP = UpdateCovP(CovP,x(3),ds,dg,b);
    [v,vd] = pcacov(CovP);
    for i = 1:3
    v(:,i) = v(:,i)/norm(v(:,i))
    end
    cur = cur + 0.01;
    plotx=[plotx;x'];

    hold on;
    grid on;
    sc=3;
    p1 = x - v(:,1) * vd(1)*sc/15
```

```
    p2 = x - v(:,2) * vd(2)*sc ;
    p3 = x + v(:,1) * vd(1)*sc/15 ;
    p4 = x + v(:,2) * vd(2)*sc ;
    line([p1(1), p3(1)],[p1(2), p3(2)],'color', 'b');
    line([p2(1), p4(1)],[p2(2), p4(2)],'color', 'r');
    xlim([0,600])
    ylim([-600,0])
    plot(plotx(:,1),plotx(:,2));
    pause(0.01);
end

%计算误差协方差矩阵
function CovX = GetCovX(ds,b,ks,kg)
    CovX = [ks*abs(ds) 0;0 kg];
end

%计算雅可比矩阵
function [Fp,Fd] = CalculateJocbian(a,ds,dg,b)
    L2 = 500;
    t = ds*sin(b+dg/2);
    t1 = a-ds*(cos(b+dg/2)/2/L2);
    Fp = [1 0 -t*sin(t1);0 1 t*cos(t1);0 0 1];

    t = a-ds*cos(b)/2/L2;
    Fd = [cos(t)*sin(b)+ds*sin(t)*cos(b)*sin(b)/2/L2, ds*cos(t)*cos(b) - ds^2*sin(t)*sin(b)^2/2/L2 ;
           sin(t)*sin(b)-ds*cos(t)*cos(b)*sin(b)/2/L2, ds*sin(t)*cos(b) + ds^2*cos(t)*sin(b)^2/2/L2 ;
           -cos(b)/L2                                , ds*sin(b)/L2];
end

%计算位置协方差
function CovP = UpdateCovP(CovP,a,ds,dg,b)
    Fp=[];Fd=[];
    [Fp,Fd] = CalculateJocbian(a,ds,dg,b);
    CovX = GetCovX(ds,b,0.02,0.02);
    CovP = Fp*CovP*Fp' + Fd*CovX*Fd'
end
```

用红色的线代表协方差：


0
-100
-200
-300
-400
-500
-600
0
100
200
300
400
500
600


0
-100
-200
-300
-400
-500
-600
0
100
200
300
400
500
600



0
-100
-200
-300
-400
-500
-600
0
100
200
300
400
500
600