Лабораторная работа №2. Задача о погоне

c/б 1032186063 | НФИбд-01-18

Доборщук Владимир Владимирович

27 марта 2021

Содержание

# Цели и задачи

**Цель:** изучить задачу о погоне, а также реализовать её модель программно.

**Задачи:**

* изучить теорию о задаче о погоне
* построить модель для 2 случаев траектории движения

# Программная реализация

## Подготовка к моделированию

Все данные соответствуют варианту 14 = .

**Инициализация библиотек**

import math  
import numpy as np  
from scipy.integrate import odeint  
import matplotlib.pyplot as plt

**Начальные данные и необходимые функции**

k=7.5  
n=3.1  
  
fi=math.pi/3  
x1=k/(n+1)  
  
r0=x1  
tetha0=0  
tetha=np.arange(tetha0, 2\*math.pi, 0.001)  
  
t=np.arange(0,6.284,0.001)  
t1=np.arange(r0, k, 0.01)

Объявим необходимые функции, исходя из данной нам информации в теоретической справке.

def dr(r,tetha):  
 dr = r/math.sqrt(n\*n-1)  
 return dr  
  
def f2(t):  
 xt=math.tan(fi+math.pi)\*t  
 return xt  
  
def f3(t1):  
 xt=math.tan(tetha0)\*t1  
 return xt  
  
def cart2pol(x,y):  
 rho = np.sqrt(x\*\*2 + y\*\*2)  
 phi = np.arctan2(y, x)  
 return rho, phi

Заложим в переменные решения для наших СДУ с помощью функции odeint модуля scipy.integrate, а также найдём точку пересечения траекторий.

## Построение графиков

### Модель первого типа

r=odeint(dr,r0,tetha)  
r1, tetha1 = cart2pol(t, f2(t))  
r2, tetha2 = cart2pol(t1, f3(t1))  
  
for i in range(len(tetha)):  
 if abs(tetha1[i]-tetha[i]) <= 0.001:  
 r\_ins = tetha[i]  
 tetha\_ins = r[i][0]  
  
plt.polar(tetha,r,'g')  
plt.polar(tetha2, r2, 'g')  
plt.polar(tetha1,r1,'b')  
plt.polar(r\_ins, tetha\_ins, color='firebrick', marker='o', label='crosspoint')  
plt.legend(loc='lower left')  
  
print('crosspoint: r='+str(r\_ins)+', tetha='+ str(tetha\_ins))

output: crosspoint: r=1.048, tetha=2.6145016420772618

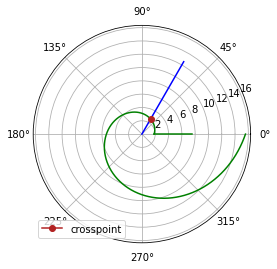


График первого случая

### Модель второго типа

x2 = k/(n-1)  
r0 = x2  
  
tetha0 = -math.pi  
tetha = np.arange(tetha0, -tetha0, 0.001)  
r=odeint(dr,r0,tetha)

t1=np.arange(-k, -r0, 0.01)  
t=np.arange(0,10,0.001)

r1, tetha1 = cart2pol(t, f2(t))  
r2, tetha2 = cart2pol(t1, f3(t1))  
  
for i in range(len(tetha)):  
 if abs(tetha1[i]-tetha[i]) <= 0.001:  
 r\_ins = tetha[i]  
 tetha\_ins = r[i][0]  
  
plt.polar(tetha,r,'g', label='police')  
plt.polar(tetha2, r2, 'g')  
plt.polar(tetha1,r1,'b', label='pirates')  
plt.polar(r\_ins, tetha\_ins, color='firebrick', marker='o', label='crosspoint')  
plt.legend(loc='lower left')  
  
print('crosspoint: r='+str(r\_ins)+', tetha='+ str(tetha\_ins))

output: crosspoint: r=1.047407346409745, tetha=14.888261569723666

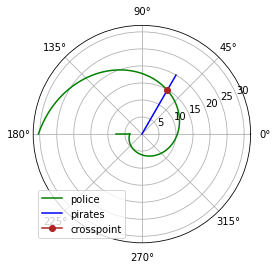


График второго случая

# Выводы

Мы изучили теорию о задаче о погоне, а также успешно реализовали 2 модели для решения этой задачи.