
Лабораторная работа №2. Задача о погоне

с/б 1032186063 | НФИбд-01-18

Доборщук Владимир Владимирович

27 марта 2021

Содержание

Цели и задачи	4
Программная реализация	5
Подготовка к моделированию	5
Построение графиков	6
Модель первого типа	6
Модель второго типа	7
Выводы	9

Список иллюстраций

1	График первого случая	7
2	График второго случая	8

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 $14 = (1032186063 \bmod 70) + 1$.

Инициализация библиотек

```
1 import math
2 import numpy as np
3 from scipy.integrate import odeint
4 import matplotlib.pyplot as plt
```

Начальные данные и необходимые функции

```
1 k=7.5
2 n=3.1
3
4 fi=math.pi/3
5 x1=k/(n+1)
6
7 r0=x1
8 tetha0=0
9 tetha=np.arange(tetha0, 2*math.pi, 0.001)
10
11 t=np.arange(0,6.284,0.001)
12 t1=np.arange(r0, k, 0.01)
```

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

```
1 def dr(r, tetha):
2     dr = r/math.sqrt(n*n-1)
3     return dr
4
5 def f2(t):
6     xt=math.tan( fi+math.pi)*t
7     return xt
8
9 def f3(t1):
10    xt=math.tan(tetha0)*t1
11    return xt
12
13 def cart2pol(x,y):
14     rho = np.sqrt(x**2 + y**2)
15     phi = np.arctan2(y, x)
16     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

```
1 r=odeint(dr,r0,tetha)
2 r1, tetha1 = cart2pol(t, f2(t))
3 r2, tetha2 = cart2pol(t1, f3(t1))
4
5 for i in range(len(tetha)):
6     if abs(tetha1[i]-tetha[i]) <= 0.001:
7         r_ins = tetha[i]
8         tetha_ins = r[i][0]
9
10 plt.polar(tetha,r,'g')
11 plt.polar(tetha2, r2, 'g')
12 plt.polar(tetha1,r1,'b')
13 plt.polar(r_ins, tetha_ins, color='firebrick', marker='o', label='
    crosspoint')
14 plt.legend(loc='lower left')
15
16 print('crosspoint: r='+str(r_ins)+' , tetha='+ str(tetha_ins))
```

output: crosspoint: $r=1.048$, $tetha=2.6145016420772618$

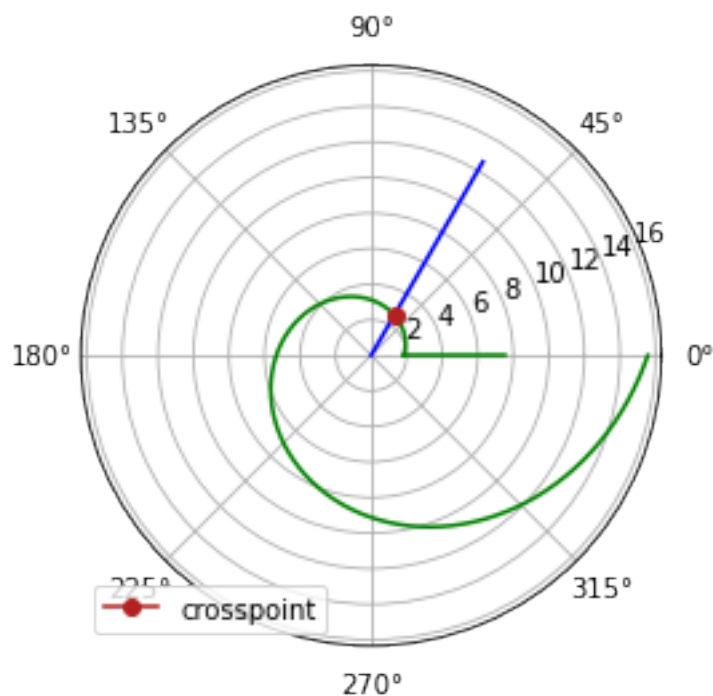


Рис. 1: График первого случая

Модель второго типа

```
1 x2 = k/(n-1)
2 r0 = x2
3
4 tetha0 = -math.pi
5 tetha = np.arange(tetha0, -tetha0, 0.001)
6 r=odeint(dr,r0,tetha)
```

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

```
1 r1, tetha1 = cart2pol(t, f2(t))
2 r2, tetha2 = cart2pol(t1, f3(t1))
3
4 for i in range(len(tetha)):
5     if abs(tetha1[i]-tetha[i]) <= 0.001:
6         r_ins = tetha[i]
7         tetha_ins = r[i][0]
8
9 plt.polar(tetha, r, 'g', label='police')
10 plt.polar(tetha2, r2, 'g')
11 plt.polar(tetha1, r1, 'b', label='pirates')
12 plt.polar(r_ins, tetha_ins, color='firebrick', marker='o', label='
    crosspoint')
13 plt.legend(loc='lower left')
14
15 print('crosspoint: r='+str(r_ins)+' , tetha='+ str(tetha_ins))
```

output: crosspoint: r=1.047407346409745, tetha=14.888261569723666

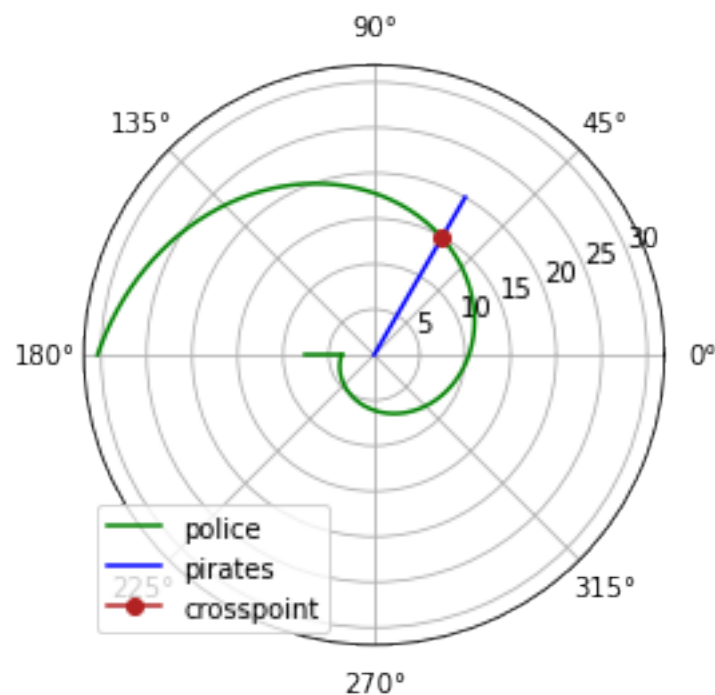


Рис. 2: График второго случая

Выводы

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