

Задача о погоне

Файзуллоев Шахрон НПИбд-02-19¹

18 февраля, 2022, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

Построения математических моделей для выбора правильной стратегии при решении задач поиска. Необходимо определить по какой траектории необходимо двигаться катеру, чтоб нагнать лодку.

Задание к лабораторной работе

1. Провести необходимые рассуждения и вывод дифференциальных уравнений, если скорость катера больше скорости лодки в n раз.
2. Построить траекторию движения катера и лодки для двух случаев.
3. Определить по графику точку пересечения катера и лодки.

Процесс выполнения лабораторной работы

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 9,8 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 3,8 раза больше скорости браконьерской лодки.

Решение исходной задачи сводится к решению системы из двух дифференциальных уравнений

$$\begin{cases} \frac{dr}{dt} = v \\ r \frac{d\theta}{dt} = v \sqrt{n^2 - 1} \end{cases}$$

с начальными условиями

$$\begin{cases} \theta_0 = 0 \\ r_0 = \frac{k}{n+1} \end{cases}$$

$$\begin{cases} \theta_0 = -\pi \\ r_0 = \frac{k}{n-1} \end{cases}$$

Исключая из полученной системы производную по t , можно перейти к следующему уравнению: $\frac{dr}{d\theta} = \frac{r}{\sqrt{n^2-1}}$

Начальные условия остаются прежними. Решив это уравнение, мы получим траекторию движения катера в полярных координатах.

The screenshot shows the JupyterLab interface. The top bar includes the Jupyter logo and a message: "lab2: Последнее изменение файла 3 минуты назад (автоматически)". Below this is a menu with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The main workspace displays a Jupyter Notebook with a single code cell. The code defines two functions, `cat` and `brak`, and performs numerical integration and plotting. The `cat` function calculates the integral of $\sqrt{c^2 - 1}$ using `odeint`. The `brak` function calculates $\tan(\pi \cdot f_1) \cdot t$. The code then sets parameters `d`, `c`, `f1`, and `x1`, and uses `scipy.integrate.odeint` to solve the differential equation. Finally, it uses `matplotlib.pyplot` to plot the results. The bottom status bar indicates "Memory: 200.8 MB / 2 GB".

```
from math import *
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plot

d = 9.8
c = 3.8

f1 = pi * 3/4

def cat(tetha,res):
    rt = res/sqrt(c**2 - 1)
    return rt

def brak(t):
    xt = tan(pi*f1)*t
    return xt

ex1 = d/(c-1)
tetha = np.arange(0,2*pi,0.01)
res = odeint(cat,ex1,tetha)
t = np.arange(0.000000000000000001,20)
lot = np.sqrt(t**2 + brak(t)**2)
tetha1 = np.arctan(brak(t)/t)

plot.rcParams["figure.figsize"] = (10,10)
plot.polar(tetha,res,'blue',label = "karep")
plot.polar(tetha1,lot,'red',label = "Asaka")

tmp = 0
for i in range(len(tetha)):
    if round(tetha[i],2) == round(f1*pi,2):
        tmp = i
print("tetha1 = ",tetha1[tmp])
```

7/10

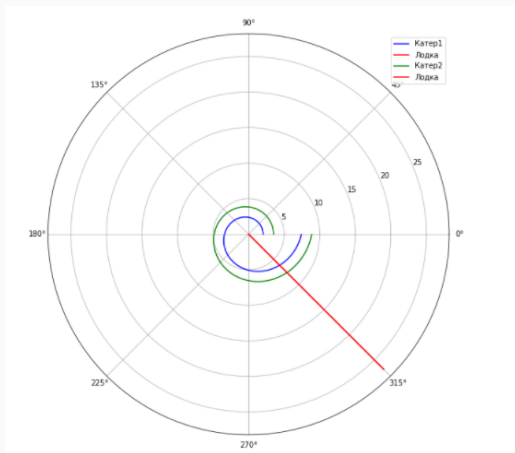


Figure 2: траектории движения для обеих случаев

Для первого случая точка пересечения красного и синего графиков - точка пересечения катера и лодки, исходя из

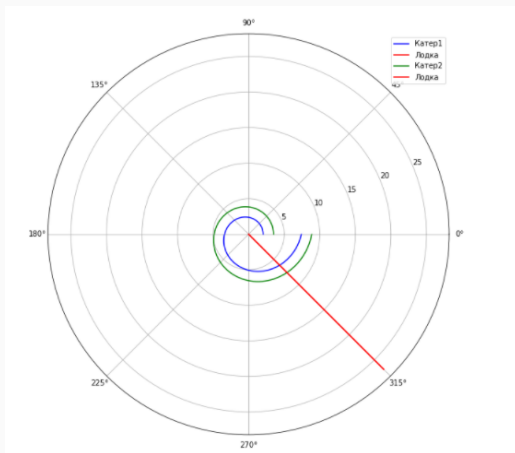


Figure 3: траектории движения для обеих случаев

Для второго случая точка пересечения красного и зеленого графиков - точка пересечения катера и лодки, исходя из

Выводы по проделанной работе

Наблюдаем, что при погоне «по часовой стрелке» для достижения цели потребуется пройти значительно меньшее расстояние

Построили математические модели для выбора правильной стратегии при решении задач поиска. Определить по какой траектории необходимо двигаться катеру, чтоб нагнать лодку.