Лабораторная работа №2

Задача о погоне

Гаглоев Олег

Contents

List of Figures

List of Tables

1 Цель работы

Вариант 38 Приведем один из примеров построения математических моделей для выбора правильной стратегии при решении задач поиска. Например, рассмотрим задачу преследования браконьеров береговой охраной. На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 19 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 5,1 раза больше скорости браконьерской лодки. Необходимо определить по какой траектории необходимо двигаться катеру, чтоб нагнать лодку.

2 Задание

1. Запишите уравнение, описывающее движение катера, с начальными условиями для двух случаев (в зависимости от расположения катера относительно лодки в начальный момент времени).
2. Постройте траекторию движения катера и лодки для двух случаев.
3. Найдите точку пересечения траектории катера и лодки

3 Теоретическое введение

Файл в приложении к Лабораторной работе №2

4 Выполнение лабораторной работы

Я не разобрался в julia, а так как баллы терять не хочется, попробовал написать на питоне, поэтому скорее всего баллов не получу Принимаем за t_0=0 , X_0=0 - место нахождения лодки браконьеров в момент обнаружения, X_0=k - место нахождения катера береговой охраны относительно лодки браконьеров в момент обнаружения лодки. Введем полярные координаты. Считаем, что полюс - это точка обнаружения лодки браконьеров x_0=0(⊠=x_0=0), а полярная ось r проходит через точку нахождения катера береговой охраны.

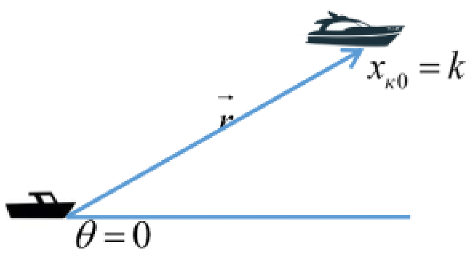

Figure 4.1: рис1

Чтобы найти расстояние x после которого катер начнет двигаться вокруг полюса , необходимо составить простое уравнение. Пусть через время t катер и лодка окажутся на одном расстоянии x от полюса. За это время лодка пройдет x , а катер k-x (или k+x , в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как x/v или (x+k/)v (или (x-k)/v во втором случае). Так как время одно и то же, то эти величины одинаковы. Тогда неизвестное расстояние x можно найти из следующего уравнения: x/v=(x+k)/v или (x-k)/v во втором случае Отсюда мы найдем два значения x1 и x2, задачу будем решать для двух случаев: x_1=k/n+1, при tetha=0 x_2=k/n-1, при tetha = -pi После того, как катер береговой охраны окажется на одном расстоянии от полюса, что и лодка, он должен сменить прямолинейную траекторию и начать двигаться вокруг полюса удаляясь от него со скоростью лодки v. Для этого скорость катера раскладываем на две составляющие: vr - радиальная скорость и vt- тангенциальная скорость. Радиальная скорость - это скорость, с которой катер удаляется от полюса V_r=dr/dt.Нам нужно, чтобы эта скорость была равна скорости лодки, поэтому полагаем v=dr/dt. Тангенциальная скорость – это линейная скорость вращения катера относительно полюса. Она равна произведению угловой скорости d tetha/dt на радиус r. vr=r*d tetha/dt

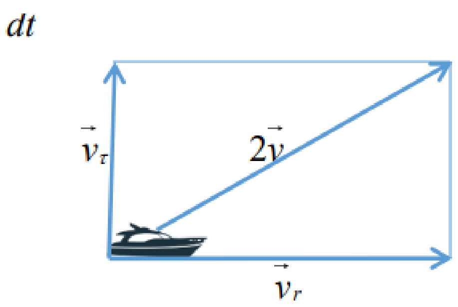

Figure 4.2: рис2

Используя теорему Пифагора получаем :v_t =sqrt (n^2 * v^2 - v^2)=v * sqrt(n^2-1). Теперь приравняем значения v_t и получаем: r * d tetha/dt=v * sqrt(n^2-1) Решение исходной задачи сводится к решению системы из двух дифференциальных уравнений: v=dr/dt и r * d tetha/dt=v * sqrt(n^2-1) Исключим из системы переменную t и получим следующее уравнение: dr/d tetha = r/sqrt(n^2-1) Теперь имея два разных начальных условия , посмотрим на результаты выполнения программы Сами начальные условия:

{ tetha=0

{r_0= s/(n+1)

и

{ tetha=−π

{r_0= s/ (n−1)

```
from math import *
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
# dr/d tetha = r/sqrt(n^2-1)
def f1(tetha, r):  # уравнение для катера
    dr = r / sqrt(n ** 2 - 1)
    return dr
def f2(t):
    xt = tan(fi + pi) * t
    return xt
s = 19
n = 5.1
fi = 3 * pi / 4
deltaR=s / (n - 1)
# решаем диф. уравнение для катера
tetha = np.arange(0, 2 * pi, 0.001)
r = odeint(f1, deltaR, tetha)
# траектория лодки
deltaT = np.arange(0.1, 50)
r1 = np.sqrt(deltaT**2 + f2(deltaT)**2)
tetha1 = np.arctan(f2(deltaT) / deltaT)
flag=0
for i in range(len(tetha)):
    if round(tetha[i],2)==round(fi+pi,2):
        flag=i
print("Theta:",tetha[flag],'r:',r[flag][0])

```

Figure 4.3: код

Графики построить не удалось из за какой то ошибки. В итоге ни на джулии не написал полностью, ни на питоне Получившиеся решения: Для первого случая

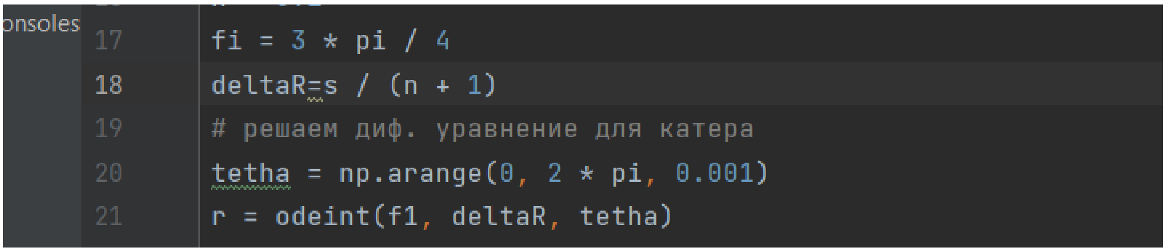

Figure 4.4: рис4

Figure 4.5: рис5

Для второго случая:

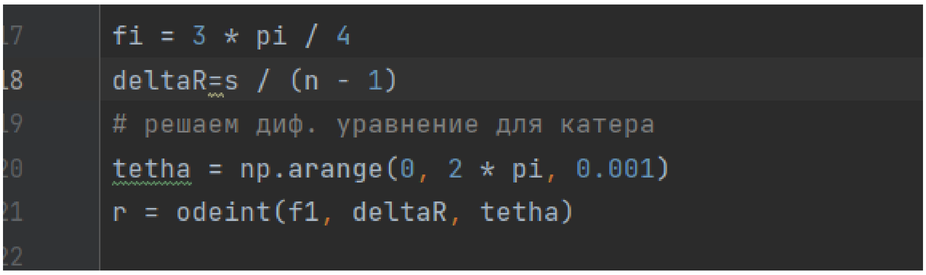

Figure 4.6: рис5

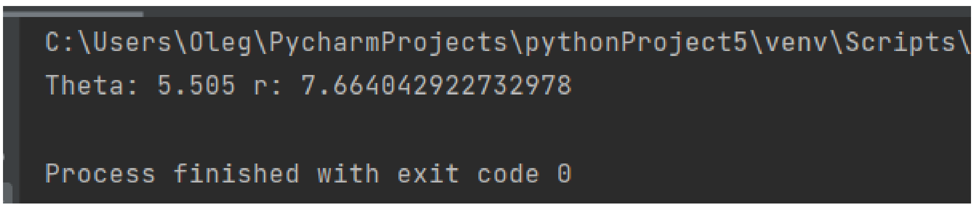

Figure 4.7: рис6

5 Выводы

Я смоделировал ситуацию, описанную в задаче и нашел необходимые значения

Список литературы

Текст к лабораторной работе №2 в ТУИС https://esystem.rudn.ru/pluginfile.php/1971721/mod_resource/content/2/Лабораторная%20работа%20№%201.pdf