Лабораторная работа №2

Задача о погоне

Дидусь К.В.

Содержание

[Цель работы 1](#_Toc68283341)

[Выполнение лабораторной работы 1](#_Toc68283342)

[Постановка задачи 1](#_Toc68283343)

[Код программы 3](#_Toc68283344)

[Результаты выполнения 4](#_Toc68283345)

[Выводы 6](#_Toc68283346)

# Цель работы

Ознакомиться с задачей о погоне. Научиться ее решать строить графики траектории движения, выводить уравнение, описывающее движение.

# Выполнение лабораторной работы

## Постановка задачи

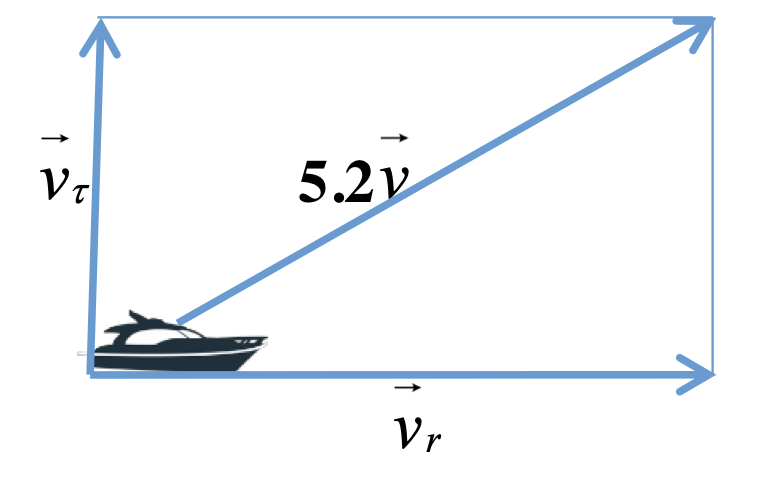
1. Место нахождения лодки браконьеров в момент обнаружения: . Место нахождения катера береговой охраны относительно лодки браконьеров в момент обнаружения лодки:
2. Введем полярные координаты. Считаем, что полюс - это точка обнаружения лодки браконьеров , а полярная ось r проходит через точку нахождения катера береговой охраны
3. Траектория катера должна быть такой, чтобы и катер, и лодка все время были на одном расстоянии от полюса, только в этом случае траектория катера пересечется с траекторией лодки. Поэтому для начала катер береговой охраны должен двигаться некоторое время прямолинейно, пока не окажется на том же расстоянии от полюса, что и лодка браконьеров. После этого катер береговой охраны должен двигаться вокруг полюса удаляясь от него с той же скоростью, что и лодка браконьеров.
4. Чтобы найти расстояние (расстояние, после которого катер начнет двигаться вокруг полюса), необходимо составить простое уравнение. Пусть через время t катер и лодка окажутся на одном расстоянии от полюса. За это время лодка пройдет , а катер — (или в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как или (во втором случае ). Так как время одно и то же, то эти величины одинаковы. Тогда неизвестное расстояние можно найти из следующего уравнения:
5. или

Отсюда мы найдем два значения и , задачу будем решать для двух случаев.

1. 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 . Для этого скорость катера раскладываем на две составляющие: — радиальная скорость и — тангенциальная скорость . Радиальная скорость - это скорость, с которой катер удаляется от полюса, . Нам нужно, чтобы эта скорость была равна скорости лодки, поэтому полагаем .

Тангенциальная скорость – это линейная скорость вращения катера относительно полюса. Она равна произведению угловой скорости на радиус

Из рисунка (рис. @fig:002) видно: (учитывая, что радиальная скорость равна ). Тогда получаем



Скорости

1. Решение исходной задачи сводится к решению системы из двух дифференциальных уравнений:

с начальными условиями

Исключая из полученной системы производную по , можно перейти к следующему уравнению:

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

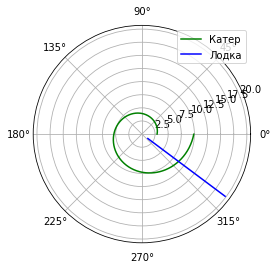
# Код программы

Приведу полный код программы (Python):  
Разработка велась в среде Spyder

import math  
import numpy as np  
from scipy.integrate import odeint  
import matplotlib.pyplot as plt  
  
t = np.arange(0, 15, 1)  
  
k = 17.9 # Растояние между катером и лодкой  
fi = 3\*math.pi/4 # Направление лодки  
tang\_vel = 5.2\*\*2-1 # Тангенциальная скорость  
  
# Движение катера  
def dr(r, tetha):   
 dr = r/math.sqrt(tang\_vel)  
 return dr  
  
#  
def tetha(v, tetha0):  
 r0 = k/v  
 tetha = np.arange(tetha0, 2\*math.pi, 0.01)  
 r = odeint(dr, r0, tetha)  
 return r0, tetha, r  
  
# Движение лодки  
def f2(t):   
 xt=math.tan(fi)\*t  
 return xt  
  
def plot(tetha, r, tetha2, r2):  
 plt.figure(1)  
 plt.polar(tetha, r, 'g')  
 plt.polar(tetha2, r2, 'b')  
 plt.legend(['Катер', 'Лодка'])  
  
  
ll = t\*t + f2(t)\*f2(t)  
r2 = np.sqrt(ll)  
tetha2 = (np.tan(f2(t)/t))\*\*(-1)  
  
r0, tetha, r = tetha(4.2, 0)  
plot(tetha, r, tetha2, r2)

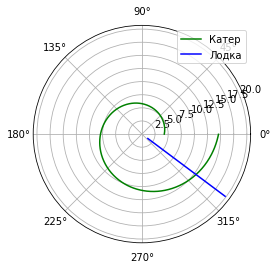
# Результаты выполнения

Случай первый (рис. @fig:001)



1

Случай второй (рис. @fig:002)



2

# Выводы

Ознакомился c задачей о погоне и нашел ее решение