Отчёт по лабораторной работе №2

Вариант 40

Аминов Зулфикор Мирзокаримович

Содержание

# 1 Цель работы

Цель данной работы - научиться выполнять построения математических моделей для выбора правильной стратегии при решении задач поиска.

# 2 Теоретическое введение

**Задача о погоне**

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 *k* км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 2 раза больше скорости браконьерской лодки.

Необходимо определить по какой траектории необходимо двигаться катеру, чтоб нагнать лодку.

**Постановка задачи**

1. Принимает за , - 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 (рис. 5.1)

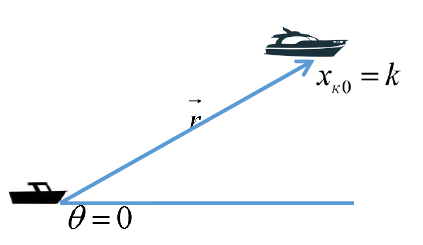


Figure 1: Положение катера и лодки в начальный момент времени

1. 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.
2. Чтобы найти расстояние x (расстояние 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 k-x/2v (во втором случае x+k/2v). Так как время одно и то же, то эти величины одинаковы. Тогда неизвестное расстояние x можно найти из следующего уравнения:
3. в первом случае или
4. во втором.

Отсюда мы найдем два значения и , задачу будем решать для двух случаев. 5. 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: - радиальная скорость и - тангенциальная скорость (рис. 2). 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 r,

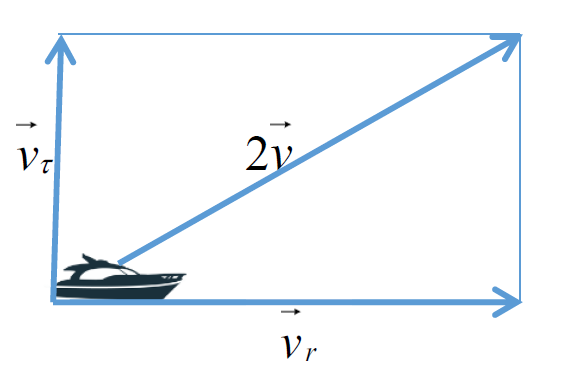


Figure 2: Разложение скорости катера на тангенциальную и радиальную составляющие

# 3 Задание

*Вариант 40*

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 15,5 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 3,5 раза больше скорости браконьерской лодки.

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.

# 4 Выполнение лабораторной работы и результат работы

**Код**

from math import \*  
import numpy as np  
from scipy.integrate import odeint  
import matplotlib.pyplot as plot  
  
n=3.5 #(в 3,5 раза больше)  
s=15.5 #(расстоянии 15,5 км)  
fi=pi\*3/4 #угол движения  
def f(tetha, r): #уравнение катера  
 dr=r/sqrt(n\*\*2 - 1)  
 return dr  
  
def f2(t): #лодка браконьеров  
 xt = tan(fi+pi)\*t  
 return xt  
r0=s/(n+1) #первый случай  
  
#решение диф уравнения для катера  
tetha = np.arange(0, 2\*pi, 0.01)  
r = odeint(f, r0, tetha)  
  
#вычисление траектории лодки  
t=np.arange(0.00000000000001, 20)  
r1=np.sqrt(t\*\*2 + f2(t)\*\*2)  
tetha1=np.arctan(f2(t)/t)  
  
plot.rcParams["figure.figsize"] = (10, 10)  
  
  
plot.polar(tetha, r, 'blue', label = 'катер')  
plot.polar(tetha1, r1, 'red', label = 'лодка')  
  
#вычисление точки пересечения  
tmp=0  
for i in range(len(tetha)):  
 if round(tetha[i], 2) == round(fi+pi, 2):  
 tmp=i  
print("Тета:", tetha[tmp], "r:", r[tmp][0])  
print("X:", r[tmp][0]/sqrt(2), "Y:", -r[tmp][0]/sqrt(2))  
  
plot.legend()  
plot.savefig("01.png",dpi=100)

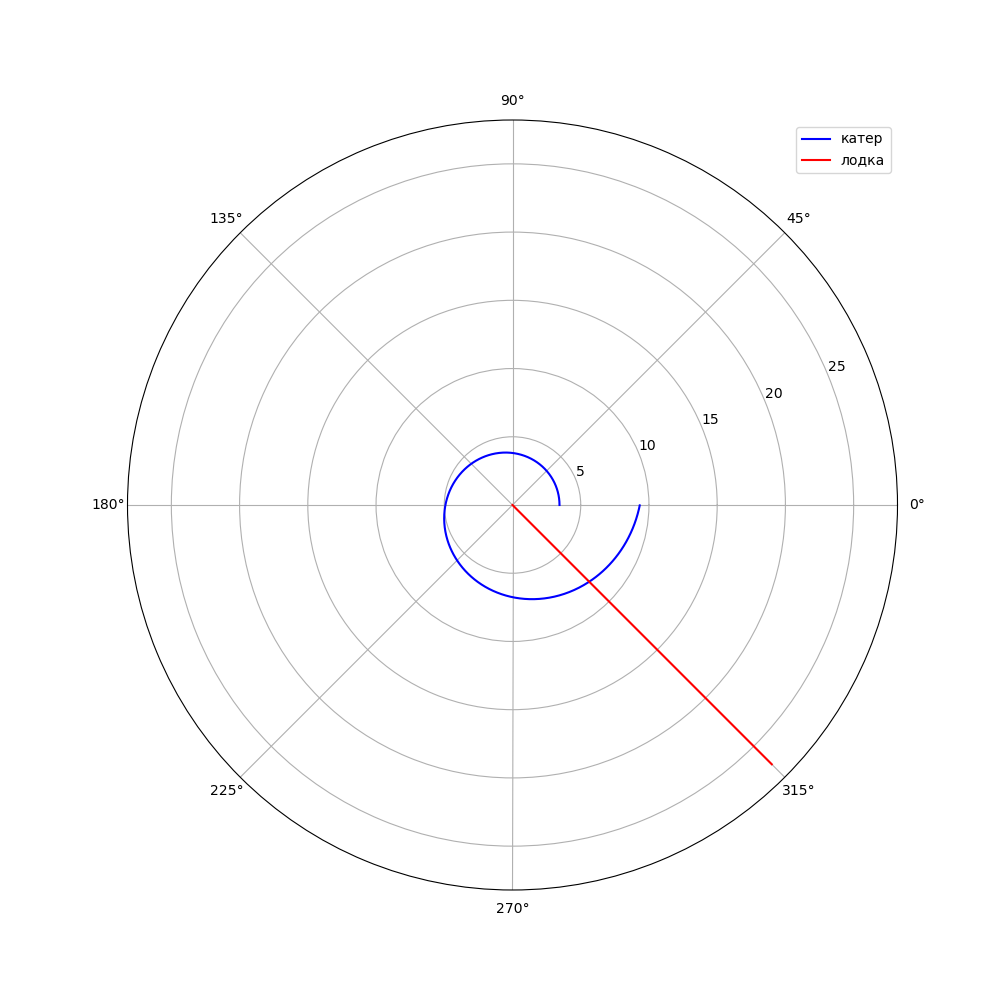


Figure 3: случай 1

///

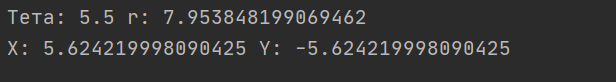


Figure 4: случай 1

from math import \*  
import numpy as np  
from scipy.integrate import odeint  
import matplotlib.pyplot as plot  
  
n=3.5 #(в 3,5 раза больше)  
s=15.5 #(расстоянии 15,5 км)  
fi=pi\*3/4 #угол движения  
def f(tetha, r): #уравнение катера  
 dr=r/sqrt(n\*\*2 - 1)  
 return dr  
  
def f2(t): #лодка браконьеров  
 xt = tan(fi+pi)\*t  
 return xt  
r0=s/(n-1) #второй случай  
  
#решение диф уравнения для катера  
tetha = np.arange(0, 2\*pi, 0.01)  
r = odeint(f, r0, tetha)  
  
#вычисление траектории лодки  
t=np.arange(0.00000000000001, 20)  
r1=np.sqrt(t\*\*2 + f2(t)\*\*2)  
tetha1=np.arctan(f2(t)/t)  
  
plot.rcParams["figure.figsize"] = (10, 10)  
  
  
plot.polar(tetha, r, 'blue', label = 'катер')  
plot.polar(tetha1, r1, 'red', label = 'лодка')  
  
#вычисление точки пересечения  
tmp=0  
for i in range(len(tetha)):  
 if round(tetha[i], 2) == round(fi+pi, 2):  
 tmp=i  
print("Тета:", tetha[tmp], "r:", r[tmp][0])  
print("X:", r[tmp][0]/sqrt(2), "Y:", -r[tmp][0]/sqrt(2))  
  
plot.legend()  
plot.savefig("02.png",dpi=100)

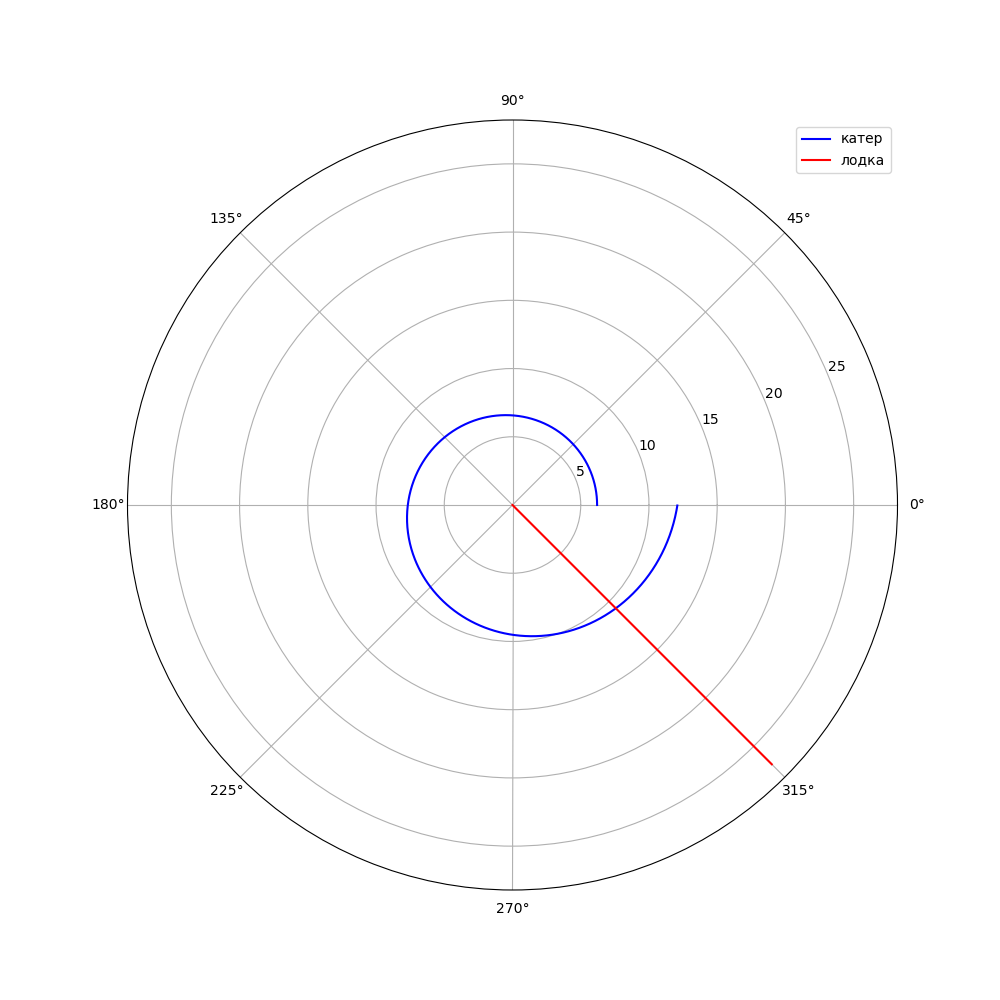


Figure 5: случай 2

///

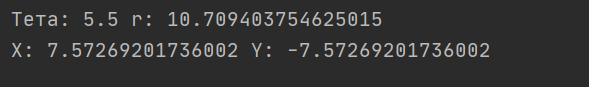


Figure 6: случай 2

# 5 Выводы

Научился выполнять построения математических моделей для выбора правильной стратегии при решении задач поиска.