Лабораторная работа № 2

Задача о погоне

Покрас Илья Михайлович

Содержание

Список иллюстраций

Цели работы

Создать алгоритм построения математической модели на примере задачи о погоне. Провести теоретические рассуждение и вывести дифференциальные уравнения для определения точки пересечения лодки и катера из задачи.

Задание

- Изучить условия задачи;
- Вывести дифференциальное уравнение, соответствующее условиям задачи;
- Написать программу для расчета и построения модели траетории движения катера и лодки.
- Определить точку пересечения катера и лодки.

Теоретическое введение

Julia - высокоуровневый высокопроизводительный свободный язык программирования с динамической типизацией, созданный для математических вычислений.

Кривая погони — кривая, представляющая собой решение задачи о «погоне», которая ставится следующим образом. Пусть точка равномерно движется по некоторой заданной кривой. Требуется найти траекторию равномерного движения точки такую, что касательная, проведённая к траектории в любой момент движения, проходила бы через соответствующее этому моменту положение точки.

Выполнение лабораторной работы

Постановка задачи

1. Принимаем за $t_0 = 0, x_{b0} = 0$ - место нахождения лодки браконьеров в момент обнаружения, $x_0 = k$ - место нахождения катера береговой охраны относительно лодки браконьеров в момент обнаружения лодки.

2. Введем полярные координаты. Считаем, что полюс - это точка обнаружения лодки браконьеров x_{b0} $(theta = x_{b0} = 0)$, а полярная ось r проходит через точку нахождения катера береговой охраны (рис. @fig:001).

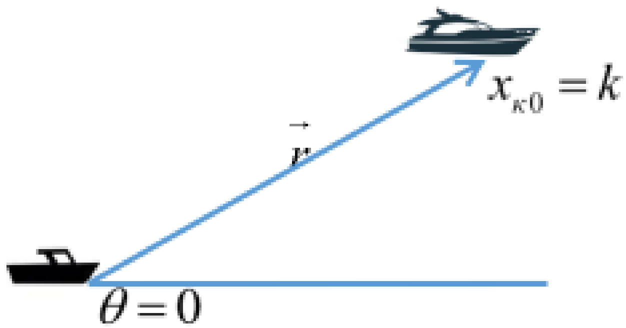

Рис. 1: Положение катера и лодки в начальный момент времени

3. Траектория катера должна быть такой, чтобы и катер, и лодка все время были на одном расстоянии от полюса theta$, 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 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/4.6v (во втором случае x+k/4.6v). Так как время одно и то же, то эти величины одинаковы. Тогда неизвестное расстояние x можно найти из следующего уравнения:

   $\frac{x}{v} = \frac{k-x}{4.6v}$ в первом случае или

   $\frac{x}{v} = \frac{x+k}{4.6v}$ во втором.

   Отсюда мы найдем два значения $x_1 = \frac{184}{46}$ и $x_2 = \frac{184}{26}$ 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 радиальную и тангенциальную скорости.

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

$$\begin{cases} \dfrac{dr}{dt} = v \\ r\dfrac{d\theta}{dt} = \sqrt{20{,}16}\,v \end{cases} \qquad \begin{cases} \theta_0 = 0 \\ r_0 = x_1 = \dfrac{184}{56} \end{cases} \quad \text{или} \quad \begin{cases} \theta_0 = -\pi \\ r_0 = x_1 = \dfrac{184}{36} \end{cases}$$

$$\frac{dr}{d\theta} = \frac{r}{\sqrt{20{,}16}}$$

$$\frac{dr}{r} = \frac{d\theta}{\sqrt{20{,}16}}$$

$$ln\, r = \int \frac{d\theta}{\sqrt{20{,}16}} = \frac{\theta}{\sqrt{20{,}16}}$$

$$r = Ce^{\frac{\theta}{\sqrt{20{,}16}}}$$

$$C_1 = \frac{65}{36} \qquad C_2 = \frac{65}{16}e^{\frac{\sqrt{20.16}}{-\pi}}$$

$$r = \frac{65}{36}e^{\frac{\theta}{\sqrt{20.16}}} \qquad r = \frac{65}{16}e^{\frac{\sqrt{20.16}}{-\pi}}e^{\frac{\theta}{\sqrt{20.16}}} = \frac{65}{16}e^{\frac{\theta+\pi}{\sqrt{20.16}}}$$

Рис. 2: Система из двух дифференциальных уравнений

7. Напишем программу для построения траектори движения катера береговой охраны и лодки с помощью Julia.(рис. @fig:004).

```julia
using Plots
using DifferentialEquations

r₀ = 184/56
θ₀ = 0.0
tspan = (0, 45.0)
boat_r = Float64[0.0, 45.0]
fi = Float64[3*π/4]

function ode_fn(du,u, p, t)
    du[1] = 1
    du[2] =  sqrt(20.16) / u[1]
end

prob = ODEProblem(ode_fn, [r₀, θ₀], tspan)
sol = solve(prob, dtmax=0.01)
r = [u[1] for u in sol.u]
θ = [u[2] for u in sol.u]
intersection_r = 0

for (i,θ) in enumerate(θ)
    if (round(θ, digits=2) == round(fi[1], digits=2))
        global intersection_r = r[i]
        break
    end
end

plt1 = plot(proj = :polar, aspect_ratio=:equal, dpi=300, title="Задача о погоне",legend=true)
plot!(plt1, θ, r, label="Траектория катера", color=:green)
plot!(plt1, [fi], boat_r, label="Траектория лодки", color=:red)
plot!( plt1, [fi], [intersection_r], seriestype = :scatter, label="Точка пересечения", color=:blue)
savefig(plt1, "model1.png")

r₀ = 184/36
θ₀ = -pi
prob = ODEProblem(ode_fn, [r₀, θ₀], tspan)
sol = solve(prob, dtmax=0.01)
r = [u[1] for u in sol.u]
θ = [u[2] for u in sol.u]

for (i,θ) in enumerate(θ)
    if (round(θ, digits=2) == round(fi[1], digits=2))
        global intersection_r = r[i]
        break
    end
end

plt2 = plot(proj = :polar, aspect_ratio=:equal, dpi=300, title="Задача о погоне",legend=true)
plot!(plt2, θ, r, label="Траектория катера", color=:green)
plot!(plt2, [fi], boat_r, label="Траектория лодки", color=:red)
plot!( plt2, [fi], [intersection_r], seriestype = :scatter, label="Точка пересечения", color=:blue)
savefig(plt2, "model2.png")
```

Рис. 3: Код программы

Рис. 4: Тракетория движения в 1 случае

Рис. 5: Тракетория движения в 2 случае

Вывод

Мы научились создавать алгоритмы построения математической модели на примере задачи о погоне.

Список литературы

[1] https://julialang.org/

[2] https://yamadharma.github.io/ru/post/2021/01/02/julia-differentialequations-callback-functions/

[3] https://docs.juliahub.com/DifferentialEquations/UQdwS/6.15.0/tutorials/ode_example/

[4] https://docs.juliaplots.org/stable/