

Лабораторная работа №2

Задача о погоне

Кузнецов Юрий Владимирович

Содержание

Цель работы	1
Задачи.....	1
Термины	1
Теоретическая справка.....	2
Выполнение лабораторной работы.	2
Код.....	5
Вывод.....	6
Ресурсы.....	7

Цель работы

Решить задачу о погоне. Смоделировать кривую погони при помощи Julia и OpenModelica.

Задачи

1. Провести аналогичные 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

Термины

- Задача о погоне — классическая задача из области дифференциальных уравнений.
- Кривая погони — кривая, представляющая собой решение задачи о «погоне».
- Julia – это открытый свободный высокопроизводительный динамический язык высокого уровня, созданный специально для технических

(математических) вычислений. Его синтаксис близок к синтаксису других сред технических вычислений, таких как Matlab и Octave.

- OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica.

Теоретическая справка

Для построения кривой погони, нам необходимо знать начальные условия и уравнения кривой. Ориентируясь на рассуждения из пособия к лабораторной работы можно вывести общие формулы. Благодаря общим формулам можно будет написать программу, которая будет строить разные кривые погони в зависимости от исходных данных (расстояния и разницы в скорости).

Пусть

- n - разница в скорости, то есть скорость катера в n раз больше лодки;
- a - расстояние между катером и лодкой в момент рассеивания тумана.

Тогда общие начальные условия для первого случая выглядят следующим образом:

$$\begin{cases} \theta_0 = 0 \\ r_0 = \frac{a}{(n+1)} \end{cases}$$

Для второго случая:

$$\begin{cases} \theta_0 = -\pi \\ r_0 = \frac{a}{(n-1)} \end{cases}$$

Уравнение кривой в общем случае выглядит следующим образом:

$$r(\theta) = r_0 e^{\frac{\theta}{\sqrt{n^2-1}}}$$

Выполнение лабораторной работы.

1. Установим пакет в Julia необходимый для построения графика.

```
import Pkg Pkg.add("Plots")
```

```
uvkuznetsov — julia — julia — 125x40
Last login: Sat Apr 15 11:31:17 on ttys000
/Applications/Julia-1.8.app/Contents/Resources/julia/bin/julia ; exit;
uvkuznetsov@MacBook-Air-Urij ~ % /Applications/Julia-1.8.app/Contents/Resources/julia/bin/julia ; exit;

Documentation: https://docs.julialang.org
Type "?" for help, "j?" for Pkg help.
Version 1.8.5 (2023-01-08)
Official https://julialang.org/ release

julia> import Pkg
julia> Pkg.add("Plots")
Installing known registries into '~/.julia'
Updating registry at '~/.julia/registries/General.toml'
Resolving package versions...
Installed Wayland_protocols_jll v1.25.0+0
Installed JpegTurbo_jll v2.1.91+0
Installed FFMPEG v0.4.1
Installed Scratch v1.2.0
Installed Xorg_xcb_util_renderutil_jll v0.3.9+1
Installed ColorTypes v0.11.4
Installed x265_jll v3.5.0+0
Installed IrrationalConstants v0.2.2
Installed libfdk_aac_jll v2.0.2+0
Installed Graphite2_jll v1.3.14+0
Installed libass_jll v0.15.1+0
Installed Showoff v1.0.3
Installed Pixman_jll v0.40.1+0
Installed Libmount_jll v2.35.0+0
Installed XML2_jll v2.10.3+0
Installed GR_jll v0.72.0+0
Installed Xorg_libXext_jll v1.3.4+4
Installed Plots v1.38.9
Installed LERC_jll v3.0.0+1
Installed JSON v0.21.4
Installed Gettext_jll v0.21.0+0
Installed Preferences v1.3.0
```

Установка пакета 'Plots'

- Для подключения графики импортируем пакет Plots при помощи команды using. Константы a и n - входные данные, thetadeg - направление лодки, dTheta - шаг для равномерного разбиения периода, maxTheta - длина периода построения, cases - содержит два кейса, по которым мы будем итерироваться.

```
using Plots

const a = 18.5
const n = 6.1
const thetadeg = 295
const dTheta = 0.02
const maxTheta = 4pi
const cases = [" 1st ", " 2nd "]
```

Начало написания скрипта

Следующие итерации будут проходить в цикле следующим образом:

for item in list

end Также будем использовать условные операторы:

if condition

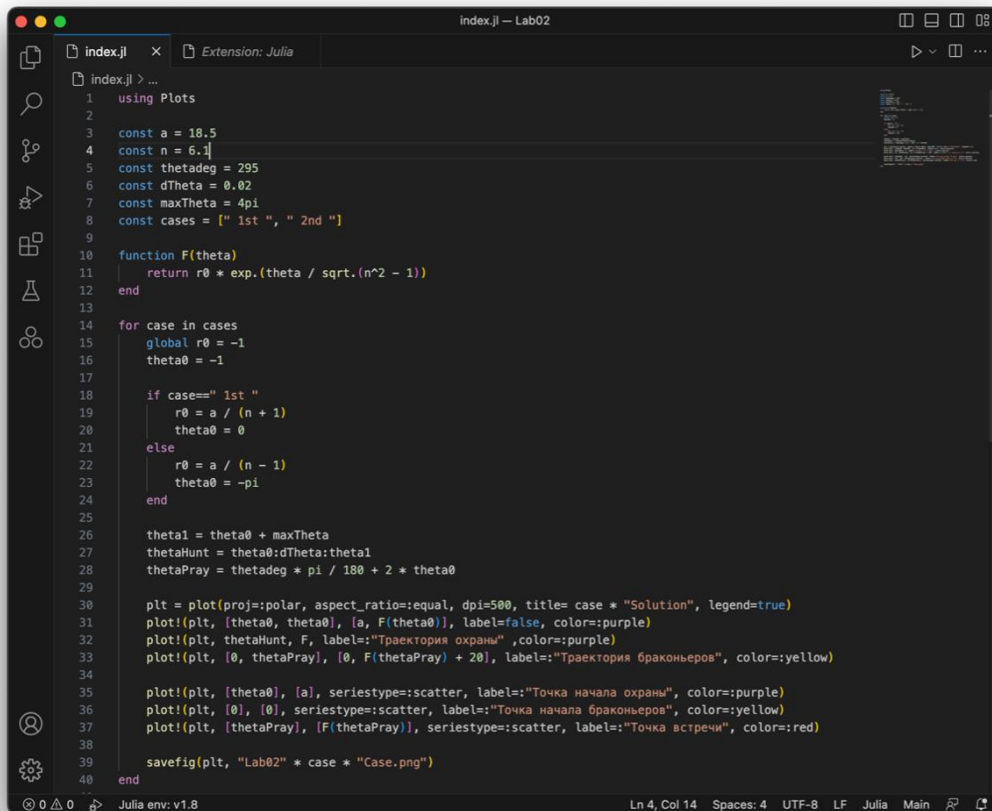
else

end

```
for case in cases
    global r0 = -1
    theta0 = -1

    if case==" 1st "
        r0 = a / (n + 1)
        theta0 = 0
    else
        r0 = a / (n - 1)
        theta0 = -pi
    end
end
```

Скрипт кейсы

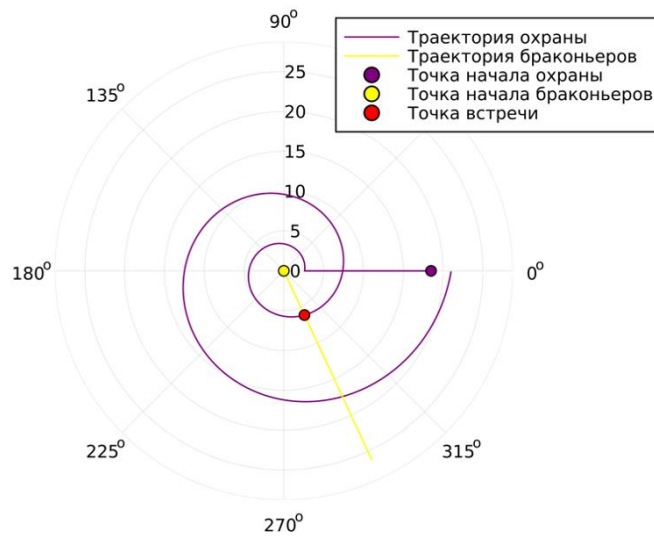
The image shows a screenshot of a Julia IDE window titled "index.jl - Lab02". The code defines constants for parameters like 'a', 'n', 'thetadeg', 'dTheta', 'maxTheta', and 'cases'. It includes a function 'F(theta)' and a main script that iterates over 'cases' to calculate initial conditions 'r0' and 'theta0'. The script then uses the 'Plots' package to create a polar plot with two trajectories: 'Траектория охраны' (purple) and 'Траектория браконьеров' (yellow). It also plots three specific points: the starting point of the guard (purple), the starting point of the poachers (yellow), and the meeting point (red). The plot is saved as a PNG file.

```
index.jl > ...
1 using Plots
2
3 const a = 18.5
4 const n = 6.1
5 const thetadeg = 295
6 const dTheta = 0.02
7 const maxTheta = 4pi
8 const cases = [" 1st ", " 2nd "]
9
10 function F(theta)
11     return r0 * exp.(theta / sqrt.(n^2 - 1))
12 end
13
14 for case in cases
15     global r0 = -1
16     theta0 = -1
17
18     if case==" 1st "
19         r0 = a / (n + 1)
20         theta0 = 0
21     else
22         r0 = a / (n - 1)
23         theta0 = -pi
24     end
25
26     theta1 = theta0 + maxTheta
27     thetaHunt = theta0:dTheta:theta1
28     thetaPray = thetadeg * pi / 180 + 2 * theta0
29
30     plt = plot(proj=:polar, aspect_ratio=:equal, dpi=500, title= case * "Solution", legend=true)
31     plot!(plt, [theta0, theta0], [a, F(theta0)], label=false, color=:purple)
32     plot!(plt, thetaHunt, F, label="Траектория охраны", color=:purple)
33     plot!(plt, [0, thetaPray], [0, F(thetaPray) + 20], label="Траектория браконьеров", color=:yellow)
34
35     plot!(plt, [theta0], [a], seriestype=:scatter, label="Точка начала охраны", color=:purple)
36     plot!(plt, [0], [0], seriestype=:scatter, label="Точка начала браконьеров", color=:yellow)
37     plot!(plt, [thetaPray], [F(thetaPray)], seriestype=:scatter, label="Точка встречи", color=:red)
38
39     savefig(plt, "Lab02" * case * "Case.png")
40 end
```

Полный скрипт

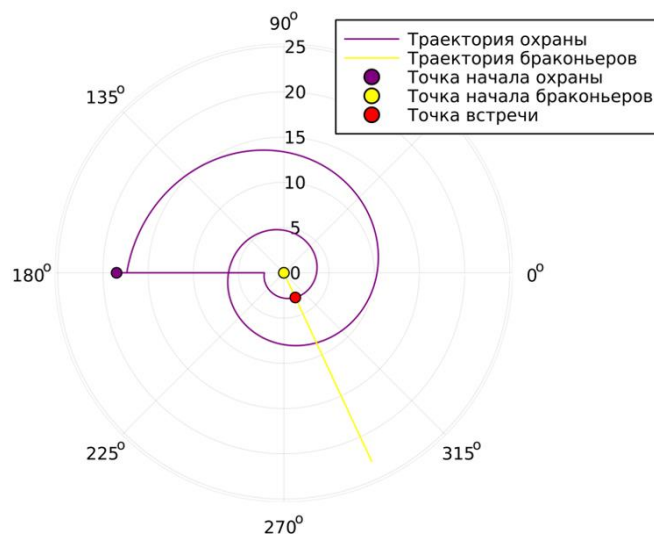
Вывод программы:

1st Solution



Кривая погони №1

2nd Solution



Кривая погони №2

На OpenModelica используя базовые настройки построение графика невозможно. В базовой настройке, графики могут выстраиваться исключительно в системе полярных координат.

Код
using Plots

```

const a = 18.5
const n = 6.1
const thetadeg = 295
const dTheta = 0.02
const maxTheta = 4pi
const cases = [" 1st ", " 2nd "]

function F(theta)
    return r0 * exp.(theta / sqrt.(n^2 - 1))
end

for case in cases
    global r0 = -1
    theta0 = -1

    if case==" 1st "
        r0 = a / (n + 1)
        theta0 = 0
    else
        r0 = a / (n - 1)
        theta0 = -pi
    end

    theta1 = theta0 + maxTheta
    thetaHunt = theta0:dTheta:theta1
    thetaPray = thetadeg * pi / 180 + 2 * theta0

    plt = plot(proj=:polar, aspect_ratio=:equal, dpi=500, title= case *
"Solution", legend=true)
    plot!(plt, [theta0, theta0], [a, F(theta0)], label=false, color=:purple)
    plot!(plt, thetaHunt, F, label="Траектория охраны", color=:purple)
    plot!(plt, [0, thetaPray], [0, F(thetaPray) + 20], label="Траектория
браконьеров", color=:yellow)

    plot!(plt, [theta0], [a], seriestype=:scatter, label="Точка начала
охраны", color=:purple)
    plot!(plt, [0], [0], seriestype=:scatter, label="Точка начала
браконьеров", color=:yellow)
    plot!(plt, [thetaPray], [F(thetaPray)], seriestype=:scatter,
label="Точка встречи", color=:red)

    savefig(plt, "Lab02" * case * "Case.png")
end

```

Вывод

Ознакомились с Julia подробнее – что дало возможность улучшить навыки в решении дифференциальных уравнений.

Ресурсы

- Задача о погоне (<https://esystem.rudn.ru/mod/resource/view.php?id=967233>)
- Кривая погони
(https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%B2%D0%B0%D1%8F_%D0%BF%D0%BE%D0%B3%D0%BE%D0%BD%D0%B8)
- Julia (http://www.unn.ru/books/met_files/JULIA_tutorial.pdf)
- OpenModelica (<https://ru.wikipedia.org/wiki/OpenModelica>)