Математическое моделирование

Лабораторная работа №2

Матюшкин Денис Владимирович (НПИбд-02-21)

Содержание

# 1 Цель работы

Построение математической модели для выбора правильной стратегии при решении задач поиска.

# 2 Задание

**Вариант 50**

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживаетсяна расстоянии 16,9 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 4,7 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.

# 3 Теоретическое введение

Julia - это высокопроизводительный язык программирования, который сочетает в себе скорость компилируемых языков с удобством использования скриптовых языков. Он предназначен для научных вычислений, анализа данных и создания высокопроизводительных приложений. Julia поддерживает многопоточность, имеет обширную экосистему библиотек и является проектом с открытым исходным кодом [1].

OpenModelica - это свободная и открытая среда для моделирования и анализа динамических систем. Она предоставляет инструменты для создания и симуляции моделей в различных областях, таких как инженерия, наука, экономика [2].

Дифференциальные уравнения (ДУ) - это уравнения, которые содержат производные неизвестной функции. Они используются для описания изменения величин в зависимости от времени или других независимых переменных [3].

# 4 Выполнение лабораторной работы

### 4.0.1 1. Математическая модель

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 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 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:

* в пером случае
* во втором случае

Отсюда мы найдем два значения и .

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 (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 , .

* (учитывая, что радиальная скорость равна ). Тогда получаем .

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

с начальными условиями или

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

### 4.0.2 2. Использование языков

Используем язык Julia для решения этой задачи.

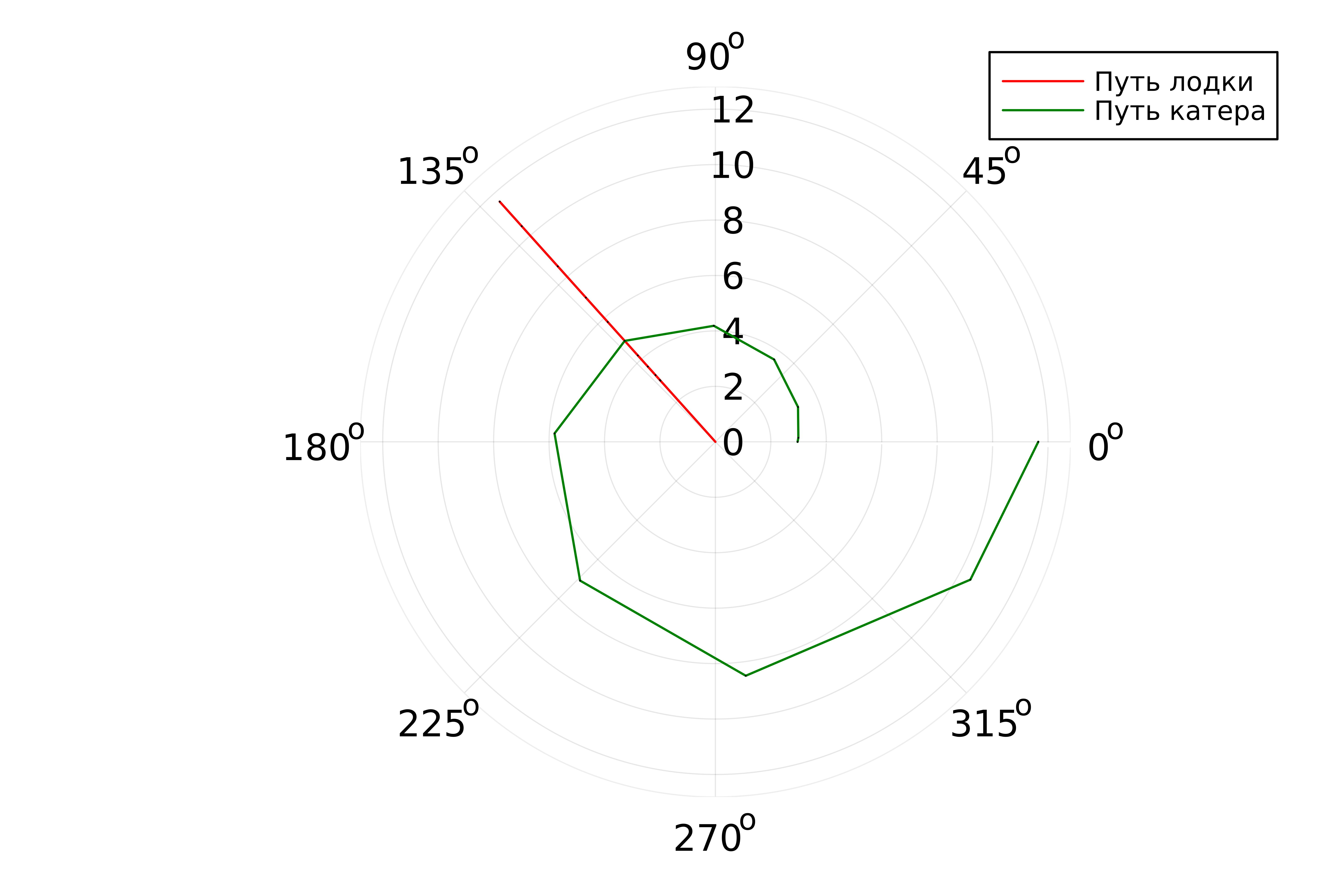
* Код программы для первого случая:

using DifferentialEquations  
using Plots  
  
const n = 16.9  
const v = 4.7  
const r = n / (v + 1)  
const t1 = (0, 2pi)  
  
function F(u, p, t)  
 return u / sqrt(v\*v - 1)  
end  
  
setup = ODEProblem(F, r, t1)  
result = solve(setup, abstol=1e-8, reltol=1e-8)  
index = rand(1:size(result.t)[1])  
rAngles = [result.t[index] for i in 1:size(result.t)[1]]  
plt = plot(proj=:polar, aspect\_ratio=:equal, dpi = 1000, legend=true, bg=:white)  
plot!(plt, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь лодки", color=:red, lw=1)  
scatter!(plt, rAngles, result.u, label="", mc=:red, ms=0.0005)  
plot!(plt, result.t, result.u, xlabel="theta", ylabel="r(t)", label="Путь катера", color=:green, lw=1)  
scatter!(plt, result.t, result.u, label="", mc=:green, ms=0.0005)  
savefig(plt, "case1.png")

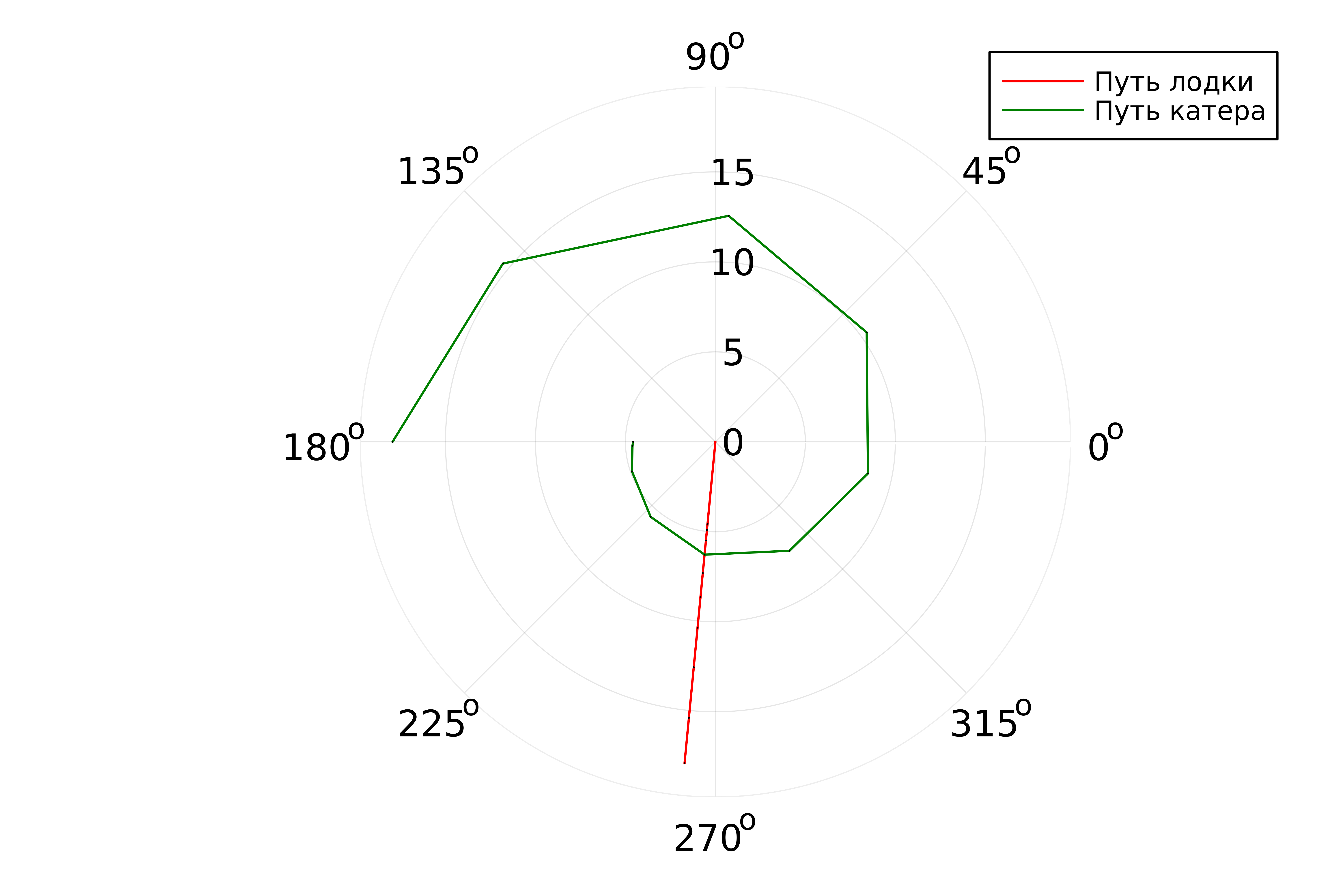
* Код программы для второго случая:

using DifferentialEquations  
using Plots  
  
const n = 16.9  
const v = 4.7  
const r = n / (v - 1)  
const t1 = (-pi, pi)  
  
function F(u, p, t)  
 return u / sqrt(v\*v - 1)  
end  
  
setup = ODEProblem(F, r, t1)  
result = solve(setup, abstol=1e-8, reltol=1e-8)  
index = rand(1:size(result.t)[1])  
rAngles = [result.t[index] for i in 1:size(result.t)[1]]  
plt = plot(proj=:polar, aspect\_ratio=:equal, dpi = 1000, legend=true, bg=:white)  
plot!(plt, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь лодки", color=:red, lw=1)  
scatter!(plt, rAngles, result.u, label="", mc=:red, ms=0.0005)  
plot!(plt, result.t, result.u, xlabel="theta", ylabel="r(t)", label="Путь катера", color=:green, lw=1)  
scatter!(plt, result.t, result.u, label="", mc=:green, ms=0.0005)  
savefig(plt, "case2.png")

Результаты сохраняются в виде картинки с расширешнием png (рис. ?? и ??).



Траекторию движения катера и лодки для первого случая



Траекторию движения катера и лодки для второго случая

В картинке видна точка пересечения лодки и катера.

# 5 Выводы

В ходе этой лабораторной работы ознакомились c языками программирования Julia и OpenModelica. Построили математическую модель для выбора правильной стратегии при решении задач поиска.

# Список литературы

1. Julia 1.10 Documentation [Электронный ресурс]. Matrix Laboratory, 2023. URL: <https://docs.julialang.org/en/v1/>.

2. User Documentation [Электронный ресурс]. Open Source Modelica Consortium, 2013. URL: <https://openmodelica.org/useresresources/userdocumentation/>.

3. Егоров Д.Л. Дифференциальные уравнения: учебное пособие. 1-е изд. Казань, 2020. 108 с.