Лабораторная работа №2

Задача о погоне. Вариант 39

Абдуллина Ляйсан Раисовна, НПИбд-01-21

Содержание

Список иллюстраций

Цель работы

Решить задачу о погоне. Изучить основы языков программирования OpenModelica и Julia.

Задачи

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

Теоретическое введение

Julia

Julia — высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков (например, MATLAB и Octave), однако имеет некоторые существенные отличия. Julia написан на Си, C++ и Scheme. Имеет встроенную поддержку многопоточности и распределённых вычислений, реализованные в том числе в стандартных конструкциях.

OpenModelica

OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. OpenModelica используется в академической среде и на производстве. В промышленности используется в области оптимизации энергоснабжения, автомобилестроении и водоочистке.

Выполнение лабораторной работы

Выбор варианта вычислялся из остатка деления студенческого билета на количесвто вариантов, плюс один. Таким образом Получили 39 вариант (Рис.1).

$$1032216538 \div 70 = 14745950 \text{ целых } 38 \text{ в остатке} = 14745950\frac{38}{70}$$

```
1032216538|70
 -70      |14745950
 332
-280
 522
-490
 321
-280
 416
-350
 665
-630
 353
-350
  38 + 1 = 39
```

Рис. 1: Вычисление варианта

Произведение расчетов

1. Примем за момент отсчета времени момент первого рассеивания тумана. Введем полярные координаты с центром в точке нахождения браконьеров и осью, проходящей через катер береговой охраны. Тогда начальные координаты катера (21; 0). Обозначим скорость лодки v.

2. 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.

3. Чтобы найти расстояние x (расстояние после которого катер начнет двигаться вокруг полюса), необходимо составить следующие уравнение. Пусть через время t катер и лодка окажутся на одном расстоянии x от полюса. За это время лодка пройдет x, а катер $21 + x$ (или $21 - x$, в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как x/v или $(21 - x)/5, 5v$, $((21 + x)/5, 5v)$. Так как время должно быть одинаковым, эти величины тоже будут друг другу равны. Из этого получаем объединение из двух уравнений (двух из-за двух разных изначальных позиций катера относительно полюса):

$$\left[\begin{array}{l} x/v = (21 - x)/5, 5v \\ x/v = (21 + x)/5, 5v \end{array}\right.$$

Из данных уравнений можно найти расстояние, после которого катер начнёт раскручиваться по спирали. Для данных уравнений решения будут следующими(Рис.2-3): $x_1 = 42/13, x_2 = 14/3$.

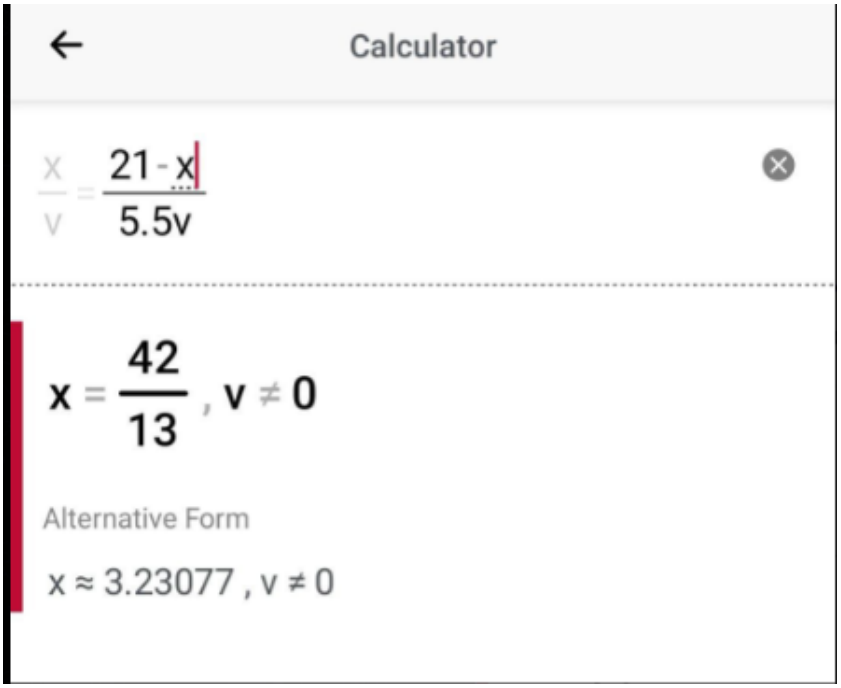


Рис. 2: Вычисления

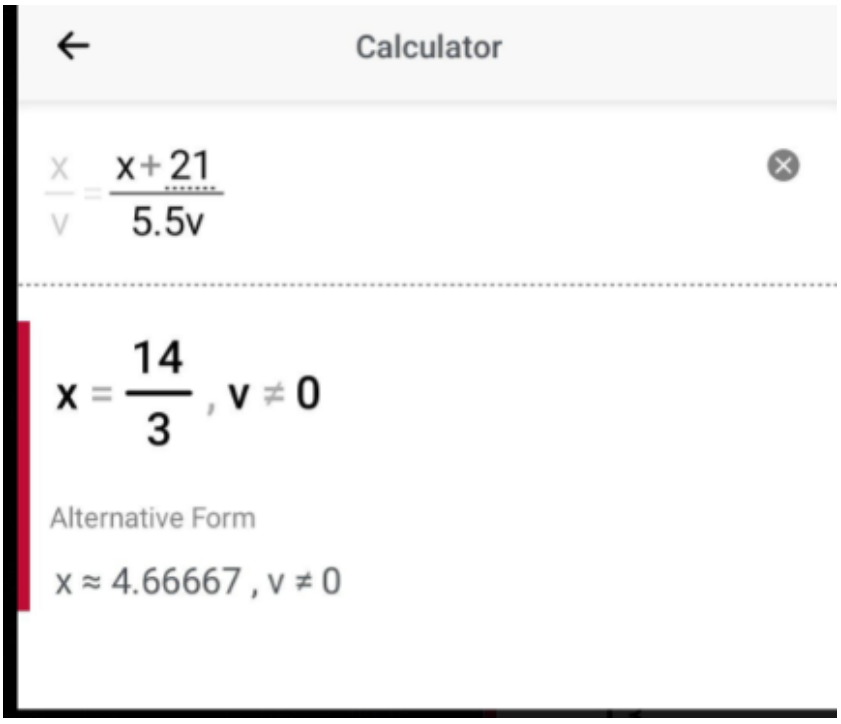


Рис. 3: Вычисления

Задачу будем решать для двух случаев. 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. (Рис.4)

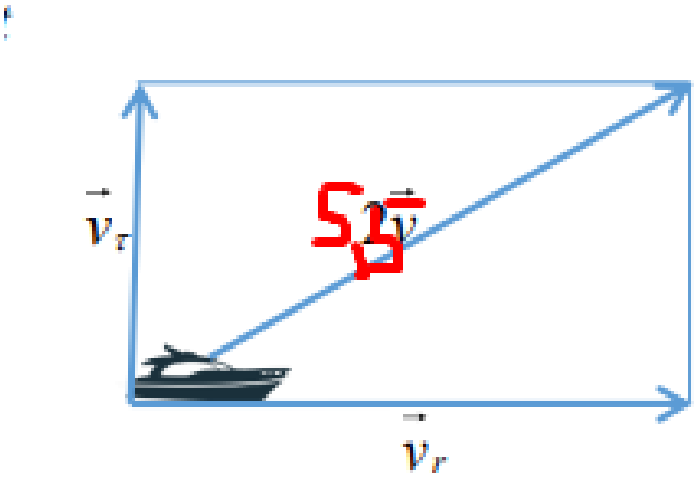


Рис. 4: Разложение скорости катера на тангенциальную и радиальную составляющие

Для этого скорость катера раскладываем на две составляющие: $v_r = dr/dt = v$ - радиальная скорость и $v_\tau = rd\theta/dt$ - тангенциальная скорость (Рис.5).

$$v_\tau = 3\sqrt{13}v/2$$

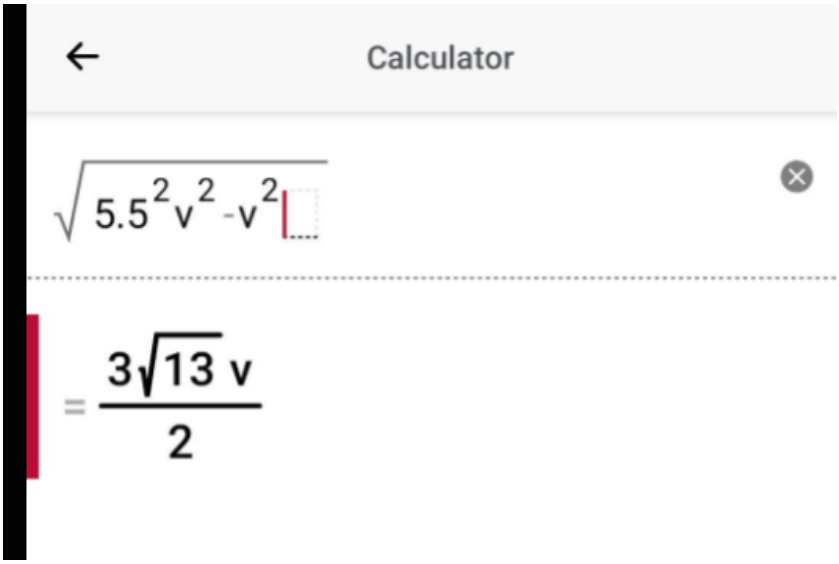


Рис. 5: Вычисления

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

$$\begin{cases} dr/dt = v \\ rd\theta/dt = 3\sqrt{13}v/2 \end{cases}$$

с начальными условиями

$$\begin{cases} \theta_0 = 0 \\ r_0 = x_1 = 42/13 \end{cases}$$

или

$$\begin{cases} \theta_0 = -\pi \\ r_0 = x_2 = 14/3 \end{cases}$$

Исключая из полученной системы производную по t, можно перейти к следующему уравнению (с неизменными начальными условиями):

$$dr/d\theta = 2r/3\sqrt{13}$$

Решением этого уравнения с заданными начальными условиями и будет являться траектория движения катера в полярных координатах.

Моделирование

OpenModelica не может быть использована для этой задачи, так как здесь используются полярные координаты.

Установка Julia и необходимх для нее пакетов (Рис.6).

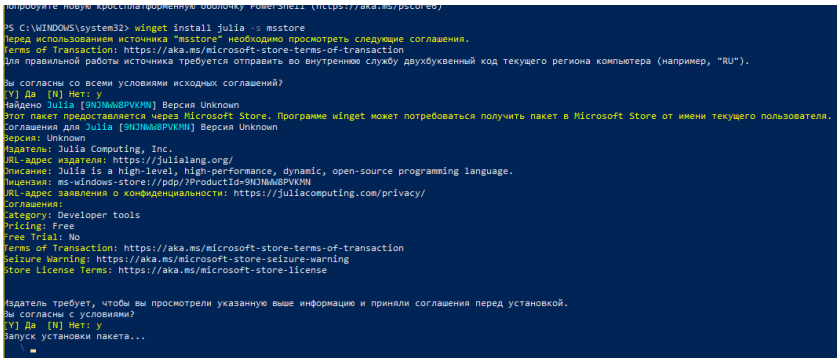


Рис. 6: Установка Julia

Исходный код программы:

using Plots using DifferentialEquations

#расстояние от лодки до катера const a = 21 const n = 5.5

#расстояние начала закругления-спирали const r0 = a/(n + 1) const r0_2 = a/(n - 1)

#интервал const T = (0, 2*pi) const T_2 = (-pi, pi)

function F(u, p, t) return u / sqrt(n*n - 1) end

#задача ОДУ problem = ODEProblem(F, r0, T)

#решение result = solve(problem, abstol=1e-8, reltol=1e-8) @show result.u @show result.t

dxR = rand(1:size(result.t)[1]) rAngles = [result.t[dxR] for i in 1:size(result.t)[1]]

#холст1 plt = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)

#параметры для холста plot!(plt, xlabel="theta", ylabel="r(t)", title="Абдуллина. Вар 39. Задача о погоне. Случай 1", legend=:outerbottom) plot!(plt, [rAngles[1], rAngles[2]],

[0.0, result.u[size(result.u)[1]]], label="Путь лодки", color=:blue, lw=1) scatter!(plt, rAngles, result.u, label="", mc=:blue, 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, "lab2_01.png")

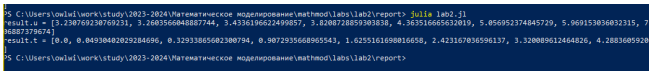
problem = ODEProblem(F, r0_2 , T_2) result = solve(problem, abstol=1e-8, reltol=1e-8) dxR = rand(1:size(result.t)[1]) rAngles = [result.t[dxR] for i in 1:size(result.t)[1]]

#холст2 plt1 = plot(proj=:polar, aspect_ratio=:equal, dpi = 1000, legend=true, bg=:white)

#параметры для холста plot!(plt1, xlabel="theta", ylabel="r(t)", title="Абдуллина. Вар 39. Задача о погоне. Случай 2", legend=:outerbottom) plot!(plt1, [rAngles[1], rAngles[2]], [0.0, result.u[size(result.u)[1]]], label="Путь лодки", color=:blue, lw=1) scatter!(plt1, rAngles, result.u, label="", mc=:blue, ms=0.0005) plot!(plt1, result.t, result.u, xlabel="theta", ylabel="r(t)", label="Путь катера", color=:green, lw=1) scatter!(plt1, result.t, result.u, label="", mc=:green, ms=0.0005)

savefig(plt1, "lab2_02.png")

Результаты работы

Запуск программы и получение результатов (Рис.7-9): 

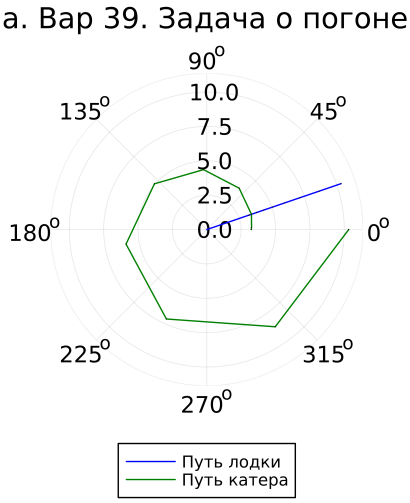


Рис. 7: Результат запуска программы - график №1

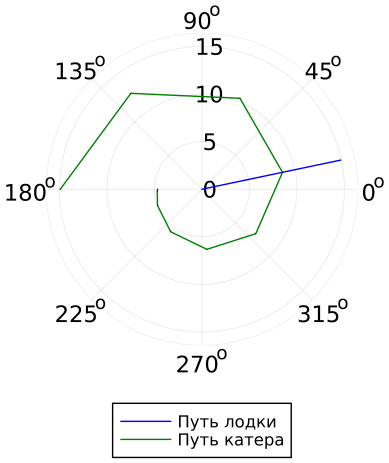


Рис. 8: Результат запуска программы - график №2

Выводы

Мы смогли решить задачу о погоне, изучить основы языков программирования Julia, а также выполнили все поставленные задачи: были построены графики для обоих случаев, где получилось отрисовать трактерию катера, траекторию лодки и получилось наглядно найти их точки пересечения.