Отчёт по лабораторной работе №2

Дисциплина: Математическое моделирование

Лушин Артём Андреевич

Содержание

Список иллюстраций

1 Цель работы

Построение математической модели решения задачи о погоне катера за браконьерской лодкой.

2 Выполнение лабораторной работы

1) Я определил свой вариант: (1132226520 mod 70)+1 = 41.

2) Принимаем за t0=0, Хл = 0 - место нахождения лодки браконьеров в момент обнаружения. За Хк = 17.4 - место нахождения катера береговой охраны относительно лодки браконьеров в момент обнаружения. Вводим полярные координаты. Считаем, что полюс Хл0 - это точка обнаружения лодки браконьеров, а полярная ось r проходит через точку нахождения катера береговой охраны. Траектория движения катера должна быть такой, чтобы и катер, и лодка всё время были на одном расстоянии от полюса. Только в этом случае траектоория катера пересекается с траекторией лодки. Поэтому для начала катер береговой охраны должен двигаться прямолинейно, а затем двигаться вокруг полюса с той же скоростью что и лодка. Чтобы найти расстояние x, необходимо составить 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 (т.к. надо рассмотреть два варианта развития событий). Время за которое они пройдут это расстояние, вычисляется как x/v или к-x/4.7v. Так как время одно и тоже, то эти величины одинаковые. Тогда неизвестное расстояние x можно найти из следующего уравнения:

$$\frac{x}{v} = \frac{17.4 - x}{4.8v}$$

$$\frac{x}{v} = \frac{x + 17.4}{4.8v}$$

$$X_1 = 3$$

$$X_2 = \frac{87}{19}$$

Рис. 2.1: Нахождение расстояния

3) После того, как катер окажется на одном расстоянии, что и лодка, он должен сменить траекторию и начать двигаться вокруг полюса удаляясь от него со скоростью лодки. Для этого скорость катера раскладываем на две составляющие: v_r - радиальная скорость и v_t - это тангенциальная скорость. Радиальная скорость v_r = dr/dt. Тангенциальная скорость равна произведению угловой скорости на радиус. Решение исходной задачи сводится к решению системы из двух диф. уравнений:

$$Vr = \sqrt{4.8^2 v^2 - v^2} = \sqrt{22.04}\, v$$

$$\begin{cases} \dfrac{dr}{dt} = v \\ r\dfrac{d\theta}{dt} = \sqrt{22.04}\, v \end{cases}$$

$$\frac{dr}{d\theta} = \frac{r}{\sqrt{22.04}}$$

С начальным условием $\begin{cases} \theta_0 = 0 \\ r_0 = 3 \end{cases}$ или $\begin{cases} \theta_0 = -\pi \\ r_0 = \frac{87}{19} \end{cases}$ 72

Рис. 2.2: Решение диф. уравнений

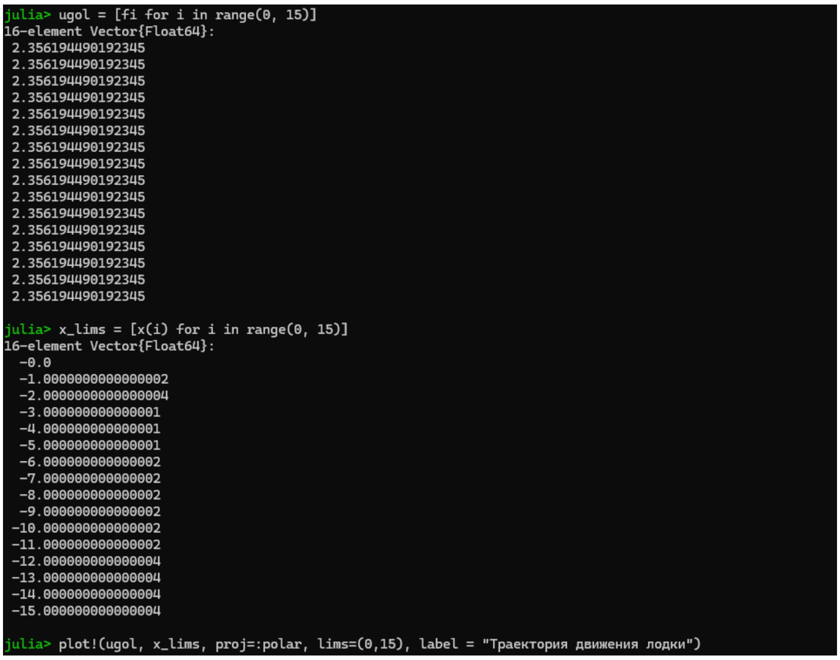
4) Строим траекторию движения катера первого случая.

```
julia> using DifferentialEquations, Plots

julia> k = 17.4
17.4

julia> r0 = k/5.8
3.0

julia> ro0_2 = k/3.8
4.578947368421052

julia> theta0 = (0.0, 2*pi)
(0.0, 6.283185307179586)

julia> theta0_2 = (-pi, pi)
(-3.141592653589793, π)

julia> fi = 3*pi/4
2.356194490192345

julia> fi = 3*pi/4;

julia> t = (0,50);

julia> x(t) = tan (fi) * t;
ERROR: ParseError:
# Error @ REPL[10]:1:11
x(t) = tan (fi) * t;
#         └ ── whitespace is not allowed here
Stacktrace:
 [1] top-level scope
   @ none:1

julia> x(t) = tan(fi) * t;

julia> f(r, p, t) = r/sqrt(22.04)
f (generic function with 1 method)

julia> prob = ODEProblem(f, r0, theta0)
ODEProblem with uType Float64 and tType Float64. In-place: false
Non-trivial mass matrix: false
timespan: (0.0, 6.283185307179586)
u0: 3.0

julia> sol = solve(prob, saveat=0.01)
retcode: Success
Interpolation: 1st order linear
t: 630-element Vector{Float64}:
 0.0
 0.01
 0.02
 0.03
 0.04
```

Рис. 2.3: Код для построения траектории катера

Рис. 2.4: Графическое построение траектории

5) Строим траекторию движения для катера и лодки в первом случае.



```
julia> ugol = [fi for i in range(0, 15)]
16-element Vector{Float64}:
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345
 2.356194490192345

julia> x_lims = [x(i) for i in range(0, 15)]
16-element Vector{Float64}:
  -0.0
  -1.0000000000000002
  -2.0000000000000004
  -3.000000000000001
  -4.000000000000001
  -5.000000000000001
  -6.000000000000002
  -7.000000000000002
  -8.000000000000002
  -9.000000000000002
 -10.000000000000002
 -11.000000000000002
 -12.000000000000004
 -13.000000000000004
 -14.000000000000004
 -15.000000000000004

julia> plot!(ugol, x_lims, proj=:polar, lims=(0,15), label = "Траектория движения лодки")
```

Рис. 2.5: Код для построения пересечения

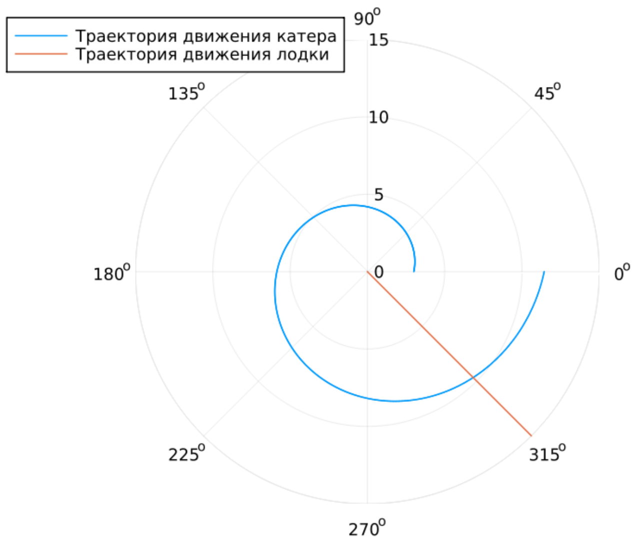


Рис. 2.6: Графическое пересечение лодки и катера

6) С помощью вычислительных мощностей находим точку пересечения катера и лодки. Для этого прописали функцию, которая является решение диф. уравнения.

```
julia> y(x) = 3*exp(5x/sqrt(551))
y (generic function with 1 method)

julia> y(fi)
4.955502566530996
```

Рис. 2.7: Точка пересечения лодки и катера

7) Расчёты и построение траектории для второго случая выполняются аналогично. Поэтому строим траекторию движения катера для второго случая.

```
julia> prob2 = ODEProblem(f, ro0_2, theta0_2)
ODEProblem with uType Float64 and tType Float64. In-place: false
Non-trivial mass matrix: false
timespan: (-3.141592653589793, 3.141592653589793)
u0: 4.578947368421052

julia> sol2 = solve(prob2, saveat=0.01)
retcode: Success
Interpolation: 1st order linear
t: 630-element Vector{Float64}:
 -3.141592653589793
 -3.1315926535897933
 -3.1215926535897935
 -3.1115926535897933
 -3.1015926535897935
 -3.0915926535897933
 -3.0815926535897935
 -3.0715926535897933
 -3.0615926535897935
 -3.0515926535897933
 -3.0415926535897935
 -3.0315926535897932
 -3.0215926535897935
  ⋮
  3.0384073464102066
  3.0484073464102064
  3.058407346410207
  3.068407346410207
  3.0784073464102066
  3.0884073464102064
  3.098407346410207
  3.108407346410207
  3.1184073464102067
  3.1284073464102065
  3.138407346410207
  3.141592653589793
u: 630-element Vector{Float64}:
  4.578947368421052
  4.588711249426708
  4.598495950366182
  4.6083015156323475
  4.618127989713791
  4.6279754171948175
  4.637843842755444
  4.6477333111714065
  4.657643867314154
  4.667575556150853
  4.6775284227443805
  4.687502512253336
  4.69749786993203
  ⋮
 17.079059312003423
 17.115477704449933
 17.151973753372175
 17.18854762436848
 17.22519948338818
 17.26192949673163
 17.29873783105018
 17.33562465334621
 17.372590130973087
 17.4096344316352
 17.44675772338796
 17.458599247318894

julia> plot(sol2.t, sol2.u, proj=:polar, lims=(0,15), label="Траектория движения катера")
```

Рис. 2.8: Код траектории катера во 2 случаи

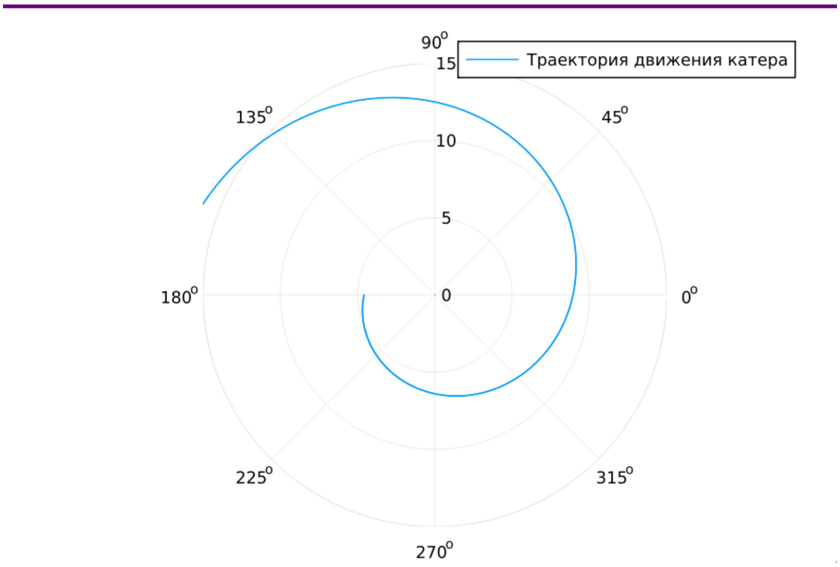


Рис. 2.9: Графическое построение катера для 2 случая

8) Затем построили пересечение катера и лодки для 2 случая.

```
julia> plot!(ugol,x_lims, proj=:polar, lims=(0,15), label="Траектория движения лодки")
```

Рис. 2.10: Код траектории пересечения во 2 случаи

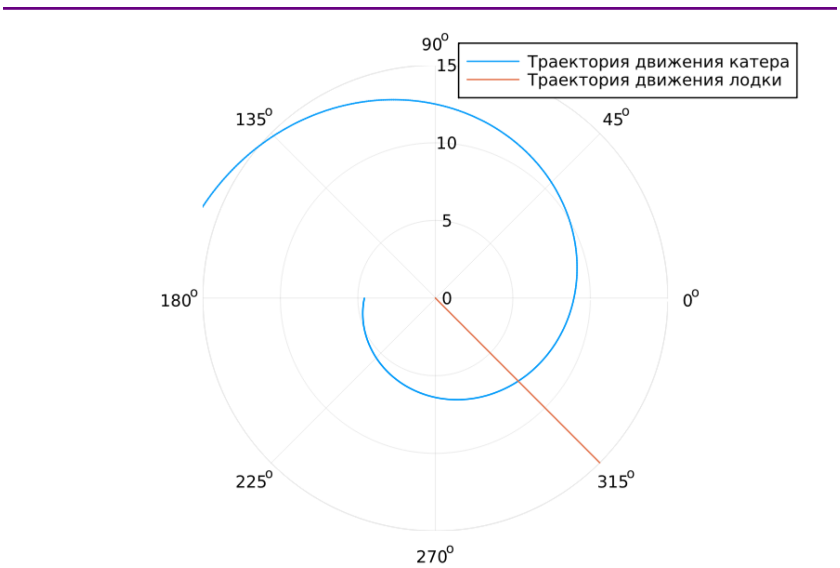


Рис. 2.11: Графическое пересечение катера и лодки

9) Как и в первом случаи, нашли точку пересечения катера и лодки.

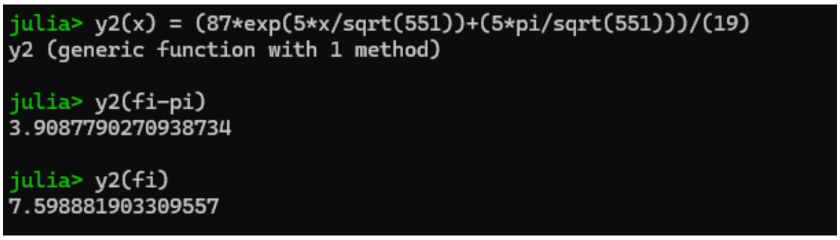


Рис. 2.12: Код для нахождения пересечения

3 Вывод

Я построил математическую модель решения задачи о погоне катера за браконьерской лодкой.