

Лабораторная работа №2

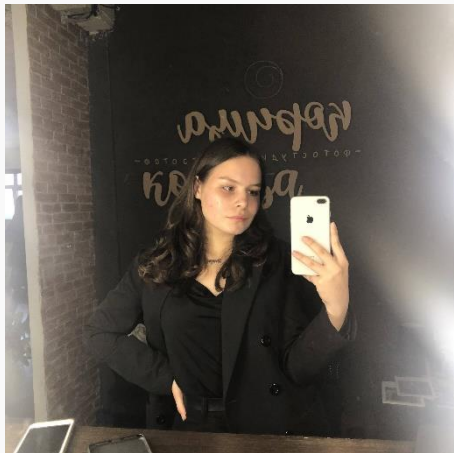
Математическое моделирование

Шатохина В.С.

2024

Российский университет дружбы народов, Москва, Россия

- Шатохина Виктория Сергеевна
- Студентка группы НФИбд-02-21
- Студ. билет 1032217046
- Российский университет дружбы народов



Цель лабораторной работы

- Изучить основы языков программирования Julia и OpenModelica. Освоить библиотеки этих языков, которые используются для построения графиков и решения дифференциальных уравнений. Решить задачу о погоне.

Задание лабораторной работы

Мой вариант 57, исходя из данной нам формулы

Этот же вариант будет использоваться для всех последующих лабораторных работ.

Задача о погоне. Вариант 57:

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 20,1 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 5 раз больше скорости браконьерской лодки.

Задачи:

1. Записать уравнение, описывающее движение катера, с начальными условиями для двух случаев (в зависимости от расположения катера относительно лодки в начальный момент времени).
2. Построить траекторию движения катера и лодки для двух случаев.
3. Найти точку пересечения траектории катера и лодки

Справка о языках программирования

- Julia — высокоуровневый высокопроизводительный 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.

Справка о языках программирования

- OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. Активно развивается Open Source Modelica Consortium, некоммерческой неправительственной организацией. Open Source Modelica Consortium является совместным проектом RISE SICS East AB и Линчёпингского университета. По своим возможностям приближается к таким вычислительным средам как Matlab Simulink, Scilab xCos, имея при этом значительно более удобное представление системы уравнений исследуемого блока.

Математическая справка

- Дифференциальное уравнение — уравнение, которое помимо функции содержит её производные. Порядок входящих в уравнение производных может быть различен (формально он ничем не ограничен). Производные, функции, независимые переменные и параметры могут входить в уравнение в различных комбинациях или отсутствовать вовсе, кроме хотя бы одной производной. Не любое уравнение, содержащее производные неизвестной функции, является дифференциальным.

Физические термины

- Тангенциальная скорость - составляющая вектора скорости, перпендикулярная линии, соединяющей источник и наблюдателя. Измеряется собственному движению - угловому перемещению источника.
- Радиальная скорость — проекция скорости точки на прямую, соединяющую её с выбранным началом координат.
- Полярная система координат — двумерная система координат, в которой каждая точка на плоскости определяется двумя числами — полярным углом и полярным радиусом.

Ход выполнения лабораторной **работы**

Построение математической модели (1)

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 $(20, 1; 0)$. Обозначим скорость лодки u
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.

Построение математической модели (2)

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 $20,1 + x$ (или $20,1 - x$ в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как $\frac{x}{v}$ или $\frac{20,1-x}{5v}$ ($\frac{20,1+x}{5v}$). 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):

$$\left[\begin{array}{l} \frac{x}{v} = \frac{20,1-x}{5v} \\ \frac{x}{v} = \frac{20,1+x}{5v} \end{array} \right.$$

Построение математической модели (3)

Из данных уравнений можно найти расстояние, после которого катер начнёт раскручиваться по спирали. Для данных уравнений решения будут следующими:

$x_1 = \frac{122}{51}$, $x_2 = \frac{122}{31}$. 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 . Для этого скорость катера раскладываем на две составляющие: $v_r = \frac{dr}{dt}$ — радиальная скорость и $v_\theta = r \frac{d\theta}{dt}$ — тангенциальная скорость.

$$v_\theta = \frac{\sqrt{1581}v}{10}$$

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

$$\begin{cases} \dot{x} = v \\ \dot{v} = -\frac{1581v}{10} \end{cases}$$

Построение математической модели (5)

С начальными условиями:

$$\begin{cases} \theta = 0 \\ \gamma = \chi = \frac{122}{51} \end{cases}$$

или

$$\begin{cases} \theta = -\pi \\ \gamma = \chi = \frac{122}{31} \end{cases}$$

Построение математической модели (5)

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

$$\frac{dr}{dt} = \sqrt{\frac{10r}{1581}}$$

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

Решение с помощью программ

К сожалению, OpenModelica не адаптирована к использованию полярных координат, поэтому адекватное отображение результатов данной задачи там невозможно.

Решить дифференциальное уравнение, расписанное в постановке задачи лабораторной работы, поможет библиотека `DifferentialEquations`. Итоговые изображения в полярных координатах будут строиться через библиотеку `Plots`. [1]

- Установим Julia
- Установим нужные библиотеки, проверим их установку

Julia 1.10.0-DEV

Documentation: <https://docs.julialang.org>

```
Type "?" for help, "]]?" for Pkg help.
```

Version 1.10.0-DEV.656 (2023-02-24)

Commit f0eadd076f (0 days old master)

```
julia> using Plots
```

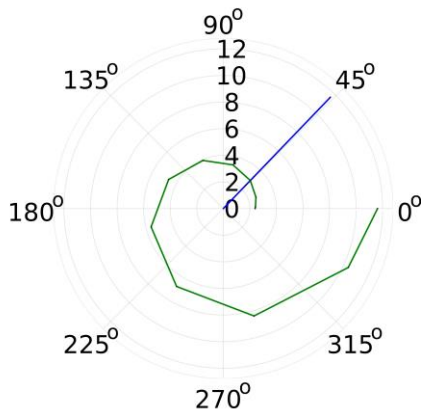
```
julia> using DifferentialEquations
```

julia>

Результаты работы кода на Julia (1)

На рис. @fig:004 и @fig:005 изображены итоговые графики траектории движения катера и лодки для случая обоих случаев.

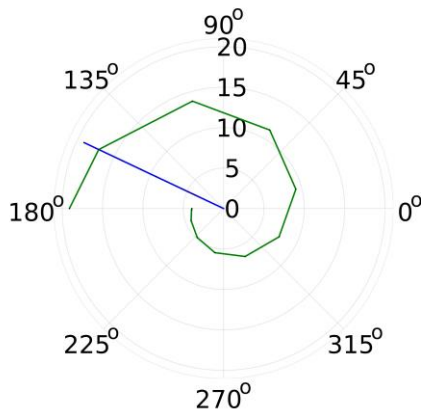
Задача о погоне - случай 1



Результаты работы кода на Julia (1)

На рис. @fig:004 и @fig:005 изображены итоговые графики траектории движения катера и лодки для случая обоих случаев.

Задача о погоне - случай 2



Мною были построены графики для обоих случаев. На них получилось отрисовать траекторию катера, траекторию лодки и получилось наглядно найти их точки пересечения. Мы успешно решили задачу о погоне.

Вывод

Были изучены основы языков программирования Julia и OpenModelica. Освоены библиотеки этих языков, которые используются для построения графиков и решения дифференциальных уравнений. Поскольку OpenModelica не работает с полярными координатами, она пока что не была использована в данной лабораторной работе.

Список литературы. Библиография

- 1 Документация по Julia: <https://docs.julialang.org/en/v1/>
- 2 Документация по OpenModelica: <https://openmodelica.org/>
- 3 Решение дифференциальных уравнений: <https://www.wolframalpha.com/>