Лабораторная работа №4

Модель гармонических колебаний

Шубнякова Дарья НКНбд-01-22

Содержание

1 Цель работы

Ознакомиться с задачей гармонических колебаний. Реализовать даннуб модель на языке Julia, а так же в среде OMEdit.

2 Задание

Постройте фазовый портрет гармонического осциллятора и решение уравнения гармонического осциллятора для следующих случаев. Наш вариант 13.

3 Теоретическое введение

Движение грузика на пружинке, маятника, заряда в электрическом контуре, а также эволюция во времени многих систем в физике, химии, биологии и других науках при определенных предположениях можно описать одним и тем же дифференциальным уравнением, которое в теории колебаний выступает в качестве основной модели. Эта модель называется линейным гармоническим осциллятором.

4 Выполнение лабораторной работы

Прописываем наш код на языке Julia в JupiterNotebook(рис. 1).

[2]:

```
using DifferentialEquations, Plots

# Общие параметры
tspan = (0.0, 75.0)
tsteps = 0.0:0.05:75.0
u0 = [1.0, 0.0]  # Начальные условия: x=1, v=0

# Случай 1: Без затухания, без внешней силы
function oscillator1!(du, u, p, t)
    x, v = u
    w = sqrt(6.5)
    du[1] = v          # dx/dt = v
    du[2] = -w^2 * x   # dv/dt = -ω²x
end

# Случай 2: С затуханием, без внешней силы
function oscillator2!(du, u, p, t)
    x, v = u
    gamma = 4.0
    w = sqrt(5.0)
    du[1] = v                 # dx/dt = v
    du[2] = -gamma*v - w^2*x  # dv/dt = -γv - ω²x
end

# Случай 3: С затуханием и внешней силой
function oscillator3!(du, u, p, t)
    x, v = u
    gamma = 3.0
    w = sqrt(7.0)
    du[1] = v                          # dx/dt = v
    du[2] = -gamma*v - w^2*x + sin(2*t) # dv/dt = -γv - ω²x + sin(2t)
end

# Решение всех трех случаев
prob1 = ODEProblem(oscillator1!, u0, tspan)
sol1 = solve(prob1, Tsit5(), saveat=tsteps)

prob2 = ODEProblem(oscillator2!, u0, tspan)
sol2 = solve(prob2, Tsit5(), saveat=tsteps)
```

Рисунок 1

Продолжение кода(рис. 2).

```
prob3 = ODEProblem(oscillator3!, u0, tspan)
sol3 = solve(prob3, Tsit5(), saveat=tsteps)

# Построение решений во времени
plt_time = plot(layout=(3,1), size=(800, 900), dpi=300)

plot!(plt_time[1], sol1.t, sol1[1,:], label="x(t)", linewidth=2,
      title="Случай 1: Без затухания, без внешней силы", xlabel="Время", ylabel="x(
plot!(plt_time[1], sol1.t, sol1[2,:], label="v(t)", linewidth=2)

plot!(plt_time[2], sol2.t, sol2[1,:], label="x(t)", linewidth=2,
      title="Случай 2: С затуханием, без внешней силы", xlabel="Время", ylabel="x(t
plot!(plt_time[2], sol2.t, sol2[2,:], label="v(t)", linewidth=2)

plot!(plt_time[3], sol3.t, sol3[1,:], label="x(t)", linewidth=2,
      title="Случай 3: С затуханием и внешней силой", xlabel="Время", ylabel="x(t)"
plot!(plt_time[3], sol3.t, sol3[2,:], label="v(t)", linewidth=2)

# Построение фазовых портретов
plt_phase = plot(layout=(1,3), size=(1200, 400))

plot!(plt_phase[1], sol1[1,:], sol1[2,:], label="Фазовая траектория", linewidth=2,
      title="Случай 1: Фазовый портрет", xlabel="x", ylabel="v", legend=false)
scatter!(plt_phase[1], [1], [0], label="Начальная точка", color=:red)

plot!(plt_phase[2], sol2[1,:], sol2[2,:], label="Фазовая траектория", linewidth=2,
      title="Случай 2: Фазовый портрет", xlabel="x", ylabel="v", legend=false)
scatter!(plt_phase[2], [1], [0], label="Начальная точка", color=:red)

plot!(plt_phase[3], sol3[1,:], sol3[2,:], label="Фазовая траектория", linewidth=2,
      title="Случай 3: Фазовый портрет", xlabel="x", ylabel="v", legend=false)
scatter!(plt_phase[3], [1], [0], label="Начальная точка", color=:red)

# Отображение графиков
display(plt_time)
display(plt_phase)
```

Рисунок 2

Получаем первые две модели: Случай без затухания, без внешней силы и Случай с затуханием, но без внешней силы(рис. 3).

```
# Сохранение результатов
savefig(plt_time, "harmonic_oscillators_time.png")
savefig(plt_phase, "harmonic_oscillators_phase.png")

println("Графики сохранены в файлы:")
println("- harmonic_oscillators_time.png - временные зависимости")
println("- harmonic_oscillators_phase.png - фазовые портреты")
```

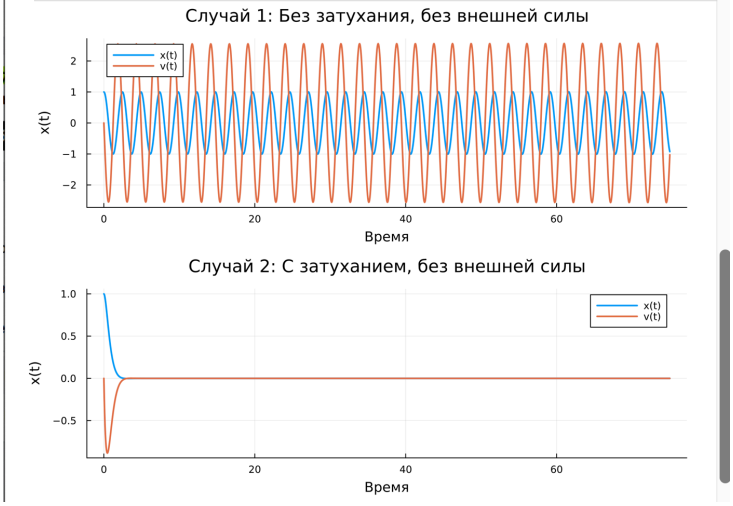

Рисунок 3

Получаем третий случай: с затуханием и внешней силой, а также фазовые портреты для всех трех случаев(рис. 4).

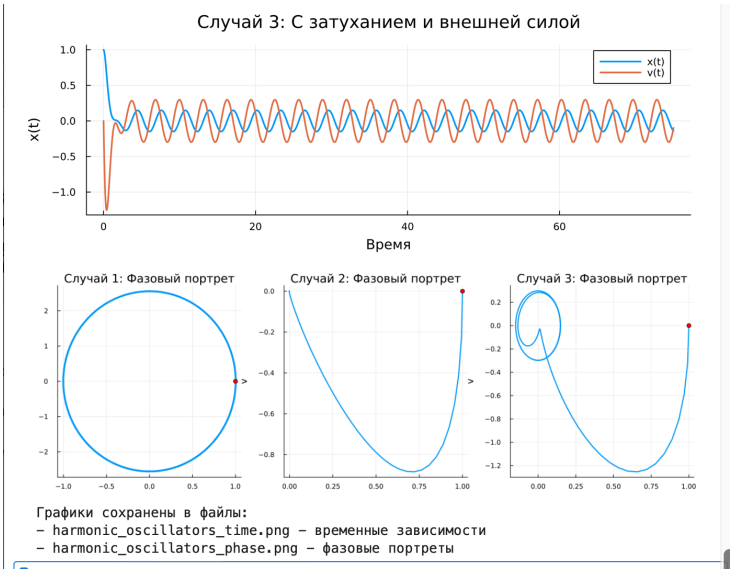

Рисунок 4

Прописываем код в реде OpenModelica для первого случая: без затухания и без внешней силы(рис. 5).

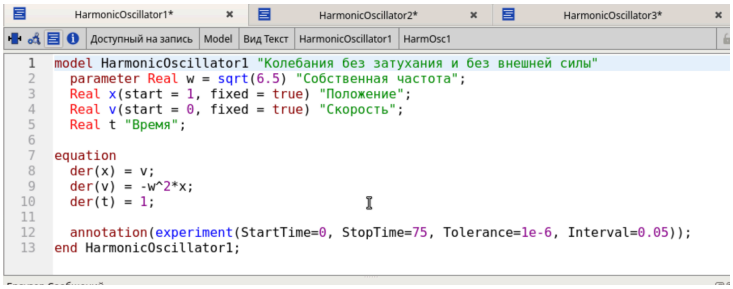

Рисунок 5

Получаем такой график(рис. 6).

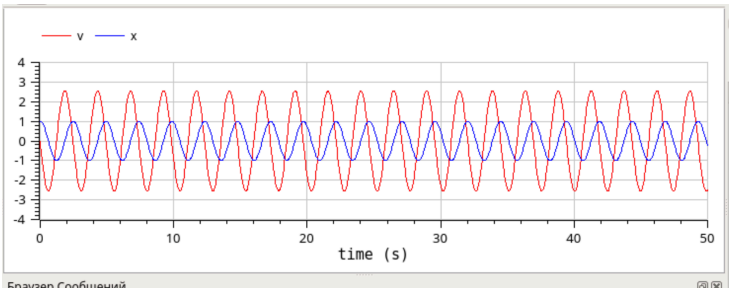

Рисунок 6

Прописываем код в реде OpenModelica для второго случая: без затухания, но с внешней силой(рис. 7).

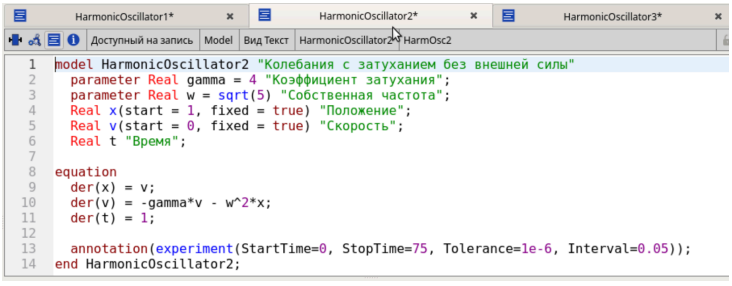

Рисунок 7

Получаем такой график(рис. 8).

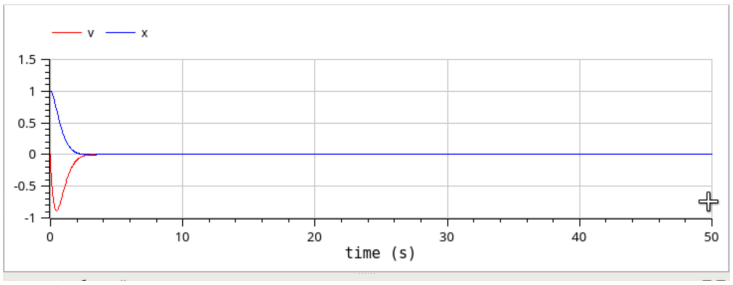

Рисунок 8

Прописываем код в реде OpenModelica для третьего случая: с затуханием и с внешней силой(рис. 9).

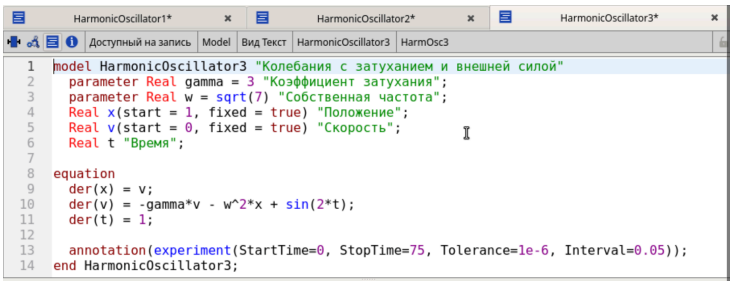

Рисунок 9

Получаем такой график(рис. 10).

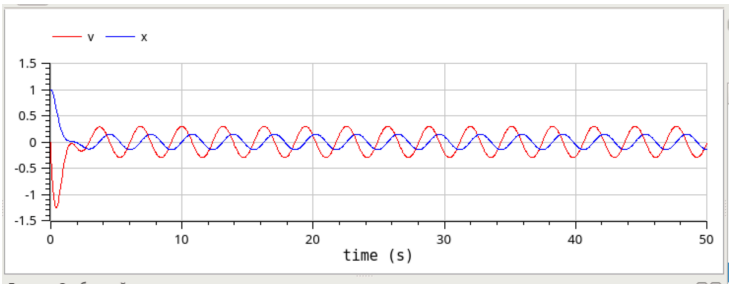

Рисунок 10

5 Выводы

Мы реализовали модель в трех ее видах в OpenModelica и на языке Julia. На выходе получили две картинки: harmonic_oscillators_phase.png и harmonic_oscillators_time.png. Итоговый файл lab4.ibybn с кодом на языке Julia в JupiterNotebook. А также три файла для симуляции в OpenModelica: HarmonicOscillator1.mo, HarmonicOscillator2.mo, HarmonicOscillator3.mo